Situationsspezifische Methoden zur Serviceidentifikation in serviceorientierten Architekturen

Grundlagen, empirische Befunde und Konstruktion einer Meta-Methode

Dissertation
zur Erlangung des akademischen Grades
Doktor der Wirtschaftswissenschaften (Dr. rer. pol.)
an der Frankfurt School of Finance & Management

vorgelegt von
René Börner
Elise-Meitner-Str. 8
63505 Langenselbold
Matrikelnummer: 4000011

Erstgutachter: Dr. Matthias Goeken
Management Department
Frankfurt School of Finance & Management

Zweitgutachter: Prof. Dr. Peter Roßbach
Management Department
Frankfurt School of Finance & Management

Drittgutachter: Prof. Dr. Carsten Felden
Lehrstuhl für Wirtschaftsinformatik
Technische Universität Bergakademie Freiberg

Frankfurt am Main, 04. Mai 2012

Danksagung

Zuallererst möchte ich meinem Erstgutacher Dr. Matthias Goeken danken. Des Weiteren danke ich Prof. Dr. Peter Roßbach und Prof. Dr. Carsten Felden für viele Hinweise zur Verbesserung meiner Dissertation und die Übernahme des Zweit- bzw. Drittgutachtens. Prof. Dr. Jürgen Moormann möchte ich für die vielen Freiheiten danken, die er mir – trotz der intensiven und umfangreichen Arbeit am ProcessLab – besonders in der Endphase meiner Dissertation zugestanden hat.

Für drei spannende und aufregende Jahre möchte ich meinen Kollegen am ProcessLab danken. Von Dr. Diana Heckl und Dr. Matthias Hilgert habe ich zu Beginn meiner Zeit an der Frankfurt School unglaublich viel lernen können. Dr. Michael Leyer und Anne Schaefer haben mich in der Endphase der Dissertation von nahezu allen Routinetätigkeiten am Lehrstuhl entlastet und somit die Fertigstellung der Dissertation überhaupt erst möglich gemacht. Die konstruktiven und bisweilen nervtötenden Diskussionen mit allen Vieren haben in letzter Konsequenz immer dazu geführt, zahlreiche Artikel und Veröffentlichungen noch ein bisschen besser zu machen. Dafür, dass sich über das Kollegendasein hinaus Freundschaften entwickelt haben, bin ich Euch sehr dankbar.

Bedanken möchte ich mich außerdem bei Dr. Gereon Berghoff und Dr. Andreas Finke, die über die Siemens IT Solutions GmbH meine Stelle als wissenschaftlicher Mitarbeiter an der Frankfurt School gestiftet sowie meine Dissertation begleitet haben.

Viele Artikel dieser kumulativen Dissertation wurden nur durch die konstruktive Zusammenarbeit mit Freunden und Kollegen ermöglicht. An dieser Stelle danke ich meinen Mitautoren Dr. Stefanie Looso, Dr. Falk Kohlmann, Dr. Axel Korthaus und Dr. Thomas Kohlborn für die erfolgreiche Zusammenarbeit.

Der Forschungsaufenthalt in Australien war ein ebenso wichtiger wie auch spannender Teil meines Promotionsvorhabens. Zum einen möchte ich den Verantwortlichen an der Frankfurt School für die Unterstützung dabei danken. Mein besonderer Dank gilt aber meinen Gastgebern Prof. Dr. Fethi Rabhi an der University of New South Wales, Sydney, sowie Prof. Dr. Michael Rosemann an der Queensland University of Technology, Brisbane, für das herzliche Willkommen und die aktive Unterstützung meiner Arbeit.

Allen Arbeitskollegen aus Fakultät und Verwaltung sei hiermit für jede Art der Unterstützung gedankt. Für das Zeichnen unzähliger Folien und Grafiken, das Suchen von Literatur und das Korrekturlesen möchte ich allen unseren studentischen Hilfskräften danken. Ein besonderer Dank für die Unterstützung beim Transkribieren stundenlanger Interviews geht an Peggy Lehnert. Manon Wehlmann möchte ich für das schier endlose (Um-)Formatieren von Artikeln danken.

Natürlich ist das Erstellen einer so umfangreichen Arbeit in guter Qualität nur durch den unerbittlichen Einsatz vieler Korrekturleser möglich. Daher geht mein Dank für das Auffinden aller orthographischen und grammatikalischen Schwächen und die aktive Unterstützung bei (zumeist englischen) Formulierungen an Marie-Noelle Biemer, Friederike Schütz, Heike Reitzammer und meine Frau Karoline. Ihr ward eine unschätzbare Hilfe. Für alle verbliebenen Fehler in dieser Arbeit ist natürlich der Autor allein verantwortlich.

Meiner gesamten Familie möchte ich für jegliche Art der Unterstützung herzlich danken. Ganz besonders danke ich natürlich meinen Eltern und das nicht deswegen, weil ich mich dazu verpflichtet fühle. Dies würde bedeuten, dass sie diesen Dank erwarten, aber gerade das ist nicht der Fall und macht die – aus ihrer Sicht selbstverständliche – Unterstützung umso wertvoller. Sie haben mich bei all meinen privaten und beruflichen Plänen stets gefördert und mir den nötigen Rückhalt geboten. Schließlich haben sie mich wesentlich geprägt und zu dem Menschen gemacht, der ich jetzt bin. Daher möchte ich ihnen diese Arbeit widmen.

Der größte Dank gilt zuguterletzt meiner Frau Karoline. Gut drei Jahre Promotionszeit sind natürlich in erster Linie eine Herausforderung für den Promovierenden. Aber auch sein Umfeld leidet mehr oder minder unter der zeitlichen Inanspruchnahme. Insbesondere für den Partner bedeutet das häufig den Verzicht auf die gemeinsame Freizeitgestaltung. Für das Verständnis und die Geduld möchte ich mich daher ganz herzlich bedanken. Die moralische Unterstützung und Aufmunterung – gerade in schwierigen Zeiten, wenn eine Abgabe der Arbeit in weite Ferne zu rücken schien – waren von unschätzbarem Wert.

RENÉ BÖRNER

Inhaltsverzeichnis

Danksagung ..... **III**

Inhaltsverzeichnis ..... **VI**

Publikationsverzeichnis ..... **VIII**

1. Einleitung **1**

1.1 Grundlagen und Terminologie 1
1.1.1 Serviceorientierte Architekturen 1
1.1.2 Serviceidentifikation 4
1.1.3 Methoden-Engineering in der Serviceidentifikation 6
1.2 Forschungslücke und Zielsetzung der Arbeit 6
1.2.1 Problemstellung 7
1.2.2 Lösungsbeitrag 8
1.3 Aufbau der Arbeit 9
1.4 Wissenschaftliche Vorgehensweise 12
1.4.1 Literaturreview 14
1.4.2 Fallstudien 14
1.4.3 Methoden-Engineering im Design Science-Prozess 15

2. Gebiet I: Vergleich existierender Serviceidentifikationsmethoden **17**

2.1 Einführung 17
2.2 Artikel 1: Identification of Business Services - Literature Review and Lessons Learned 17
2.3 Zwischenfazit 30

3. Gebiet II: Methodologischer Hintergrund **31**

3.1 Einleitung 31
3.2 Artikel 2: Using Grounded Theory for Method Engineering 31
3.3 Artikel 3: Applying Situational Method Engineering to the Development of Service Identification Methods 49
3.4 Zwischenfazit 64

4. Gebiet III: Integration fachlicher Aspekte in die Serviceidentifikation **65**

4.1 Einleitung 65
4.2 Artikel 4: A Framework for the Design of Service Maps 65
4.3 Artikel 5: Operationalisierung der IT-Governance-Kernbereiche für die Identifizierung und Gestaltung von Services 79
4.4 Artikel 6: Towards an Operationalisation of Governance and Strategy for Service Identification and Design 95
4.5 Zwischenfazit 113

5. Gebiet IV: Kontextfaktoren in Serviceidentifikationsprojekten ... 114

5.1 Einleitung ... 114
5.2 Artikel 7: SOA Development and Service Identification – A Case Study on Method Use, Context and Success Factors ... 114
5.3 Artikel 8: Generalization in Qualitative IS Research - Approaches and Their Application to a Case Study on SOA Development ... 171
5.4 Artikel 9: Fragment Selection and Context Factors in Situational Methods for Service Identification ... 207
5.5 Zwischenfazit ... 224

6. Gebiet V: Konstruktion einer Meta-Methode ... 225

6.1 Einleitung ... 225
6.2 Artikel 10: Towards Construction of Situational Methods for Service Identification ... 225
6.3 Artikel 11: A Meta Method for the Construction of Situation-Specific Methods for Service Identification ... 242
6.4 Fazit ... 313

7. Literaturverzeichnis ... 314

Eidesstattliche Versicherung ... 317

Publikationsverzeichnis

1. Börner, R., Goeken, M. (2009): Identification of Business Services - Literature Review and Lessons Learned, in: Proceedings of the 15th Americas Conference on Information Systems (AMCIS 2009), 06.-09.08.2009, San Francisco, Paper 106.

2. Looso, S., Börner, R., Goeken, M. (2011): Using Grounded Theory for Method Engineering, in: Proceedings of the 5th International IEEE Conference on Research Challenges in Information Science (RCIS 2011), 19.-21.05.2011, Guadeloupe, Frankreich, S. 98-106.

3. Börner, R. (2010): Applying Situational Method Engineering to the Development of Service Identification Methods, in: Proceedings of the 16th Americas Conference on Information Systems (AMCIS 2010), 12.-15.08.2010, Lima, Peru, Paper 18.

4. Kohlmann, F., Börner, R., Alt, R. (2010): A Framework for the Design of Service Map, in: Proceedings of the 16th Americas Conference on Information Systems (AMCIS 2010), 12.-15.08.2010, Lima, Peru, Paper 39.

5. Alter, S., Börner, R., Goeken, M. (2009): Operationalisierung der IT-Governance-Kernbereiche für die Identifizierung und Gestaltung von Services, in: Lecture Notes in Informatics (LNI), Lübeck, Deutschland, Springer, S. 3660-3674.

6. Börner, R., Goeken, M., Looso, S. (2009): Towards an Operationalisation of Governance and Strategy for Service Identification and Design, in: Proceedings of the 13th IEEE International EDOC Conference 2009, 01.-04.09.2009, Auckland, Neuseeland, S. 180-188.

7. Börner, R., Goeken, M., Rabhi, F. (2012): SOA Development and Service Identification – A Case Study on Method Use, Context and Success Factors, in: Frankfurt School - Working Paper Series, Nr. 189.

8. Goeken, M., Börner, R. (2012): Generalization in Qualitative IS Research - Approaches and Their Application to a Case Study on SOA Development, in: Australasian Journal of Information Systems, 17. Jg., Nr. 2, S. 79-108.

9. Börner, R., Goeken, M., Kohlborn, T., Korthaus, A. (2011): Fragment Selection and Context Factors in Situational Methods for Service Identification, in: Proceedings of the 5th International IEEE Conference on Research Challenges in Information Science (RCIS 2011), 19.-21.05.2011, Guadeloupe, Frankreich, S. 335-342.

10. Börner, R. (2011): Towards Construction of Situational Methods for Service Identification, in: J. Ralyté, J./Mirbel, I./Deneckère, R. (Hrsg.), ME 2011, IFIP AICT 351, IFIP International Federation for Information Processing, Springer, S. 204-218.

11. Börner, R. (2011): A Meta Method for the Construction of Situation-Specific Methods for Service Identification, Unveröffentlichter Arbeitsbericht.

1. Einleitung

In dieser kumulativen Dissertation werden elf Artikel bzw. Arbeitsberichte und ihre jeweiligen Ergebnisse in ihrem Zusammenhang dargestellt. Vorab werden Grundlagen und Terminologie beschrieben. Nach der Darlegung der Forschungslücke und Zielsetzung der Arbeit wird der Aufbau der gleichen erläutert. Schließlich wird die wissenschaftliche Vorgehensweise erörtert. Da alle Inhalte in den jeweiligen Artikeln ausführlich behandelt werden, dienen die folgenden Abschnitte lediglich der Orientierung und verweisen für eine detaillierte Betrachtung auf die entsprechenden Artikel bzw. Arbeitsberichte.

1.1 Grundlagen und Terminologie

Dieser Abschnitt beschreibt inhaltliche Grundlagen dieser Arbeit und geht dabei insbesondere auf serviceorientierte Architekturen (SOA) und Serviceidentifikation ein und liefert für den weiteren Verlauf der Dissertation wichtige Definitionen. Hinzu kommt ein kurzer Überblick über die Rolle des (situativen) Methoden-Engineering bei der Serviceidentifikation. Für eine ausführliche Beschreibung des Methoden-Engineering (ME) sowie des Situational Method Engineering (SME) sei auf den Artikel „Construction of a Situation-Specific Method for Service Identification“ (Artikel 11) verwiesen.

1.1.1 Serviceorientierte Architekturen

Seit der Einführung von Informationstechnologie (IT) kamen und gingen viele so genannte Architektur-Paradigmen. Auf Großrechner folgten Client-Server-Architekturen; Konzepte wie Enterprise Application Integration schlossen sich an. Ein aktueller Trend zur dynamischen Anpassung der IT an die Geschäftsprozesse von Unternehmen sind serviceorientierte Architekturen. Eine SOA ermöglicht den Aufbau einer modularen und an Prozessen orientierten IT-Landschaft.[1] Durch dieses Architekturkonzept wird die technische Abbildung von Geschäftsprozessen über die Kombination modularer Einzelfunktionen, den so genannten Services, realisiert. Für die weiteren Ausführungen soll folgende SOA-Definition von Bieberstein et al. als Grundlage dienen:

“A service-oriented architecture is a framework for integrating business processes and supporting IT infrastructure as secure, standardized components – services – that can be reused and combined to address changing business priorities.”[2]

Ein Service soll im Folgenden in Anlehnung an Legner und Heutschi wie folgt definiert werden:

„Services represent abstract software elements and/or interfaces which provide other applications with stable, reusable software functionality at an application-oriented, business-related level of granularity using widely applied standards.“[3]

---

[1] Starke, G./Tilkov, S. (2007); Papazoglou, M.P. (2003)

[2] Bieberstein, N. et al. (2006), S. 5

[3] Legner, C./Heutschi, R. (2007), S. 2

Services können flexibel kombiniert werden, wodurch die Geschäftsprozesse eines Unternehmens agil an sich ändernde Kundenwünsche angepasst werden können. Unter Umständen können Services – wie in Abbildung 1 zu sehen – in mehreren Geschäftsprozessen wiederverwendet werden. Grundlage dafür ist die Standardisierung der Services und ihrer Schnittstellen.

Josuttis zeigt am Beispiel einer Kundenbestellung sehr anschaulich, wie sich daraus eine Service-Hierarchie ergibt. Darin werden relativ kleine, in ihrem Funktionsumfang sehr beschränkte Basis-Services, zu Composed- und Prozess-Services zusammengefasst, die eine fachliche Funktionalität umfassend erfüllen (Abbildung 1).[4] Ein Basis-Service holt beispielsweise einen Adressdatensatz aus der Customer Relationship Management (CRM)-Datenbank. Durch die Kombination mit anderen Basis-Services zu einem Composed-Service unterstützt letzterer dann Geschäftsprozesse wie z.B. „Bestellung aufnehmen." Hier stellt sich die Frage nach der richtigen Größe und dem Zuschnitt eines Services, die nicht pauschal beantwortet werden kann. Diese so genannte *Granularität* hängt immer eng mit dem verfolgten Implementierungsziel und der geplanten Verwendung und Einsatzumgebung eines Services zusammen.

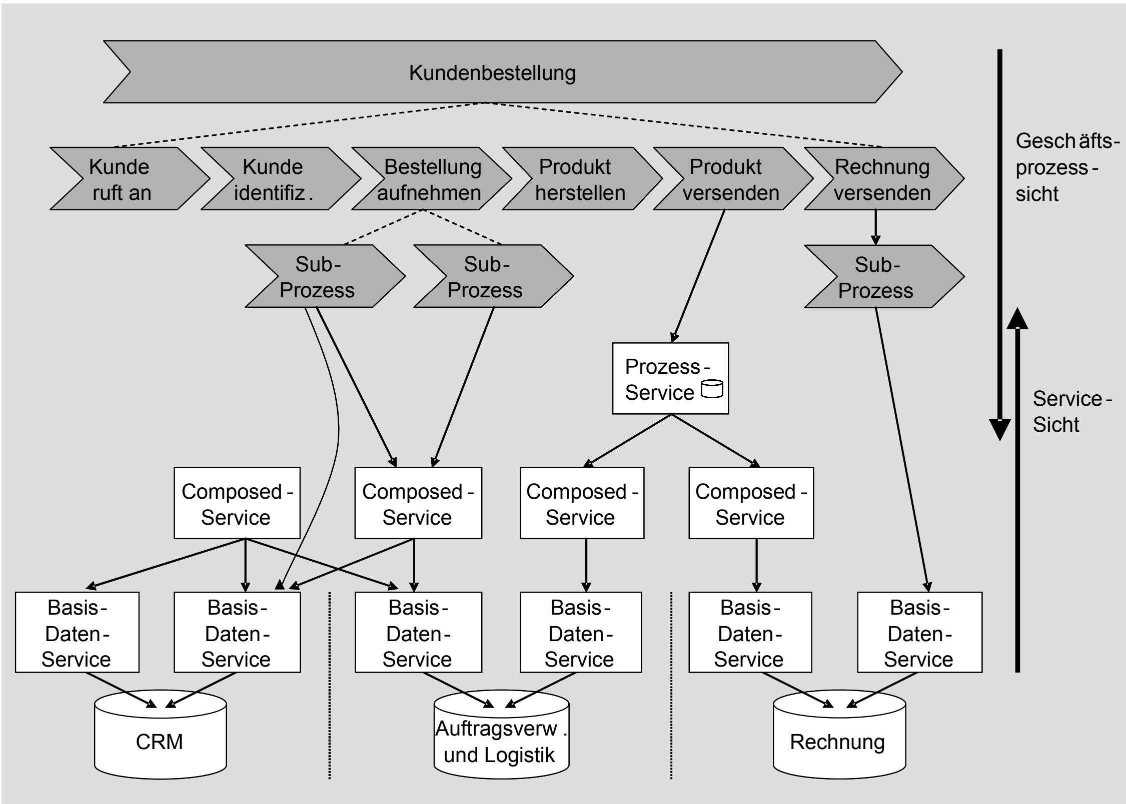

Abbildung 1: Service-Hierarchien (Quelle: Josuttis (2008), S. 111)

Im Folgenden werden die in der Literatur am häufigsten diskutierten Ziele und Merkmale serviceorientierter Architekturen kurz dargestellt. Diese sind keineswegs völlig ge-

[4] Josuttis, N. (2008)

trennt voneinander zu betrachten. Vielmehr gehen sie häufig Hand in Hand und beeinflussen sich gegenseitig.

Die *Agilität* und *Flexibilität* von Geschäftsprozessen ist heutzutage die Grundlage eines erfolgreichen unternehmerischen Handelns und eines der Hauptziele einer SOA. Kunden werden zunehmend anspruchsvoller, ändern ihre Gewohnheiten und Wünsche immer schneller und der Druck globaler Märkte auf Unternehmen wächst stetig. Letztere müssen daher imstande sein, ihre Geschäftsprozesse schnell und flexibel anzupassen. Die Anpassungsfähigkeit der zugrunde liegenden IT-Architektur ist daher überlebenswichtig.[5] Bestehende IT-Systeme (Legacy-Systeme) wurden in der Vergangenheit häufig als so genannte Insellösungen von einzelnen Abteilungen implementiert und nur von diesen genutzt. Sie sind nicht in der Lage, den gesamten Geschäftsprozess durchgehend zu unterstützten. Infolgedessen kommt es zu erheblichen Ineffizienzen an Schnittstellen zu anderen IT-Systemen. Eine Änderung von Geschäftsprozessen in diesen heterogenen IT-Landschaften ist daher langwierig und kostenintensiv. Sowohl die Entwicklungsdauer und -kosten neuer Applikationen als auch der hohe Aufwand für den Test neuer Funktionalitäten ist immens. Serviceorientierte Architekturen versprechen durch ihre Modularität hier Abhilfe.[6]

Die *Standardisierung und Automatisierung* von Services ist ein weiteres Merkmal serviceorientierter Architekturen. Inputs, Outputs und Schnittstellen werden genau definiert. Nur so kann eine flexible Kombination von Services, wie sie im vorherigen Abschnitt beschrieben wurde, sichergestellt werden. Zuvor bestimmte Standards (wie beispielsweise XML, SOAP oder UDDI) stellen die reibungslose Kommunikation zwischen den Services sicher. Damit ist von technischer Seite ein Outsourcing fast aller Services denkbar. Allerdings ist die Entscheidung, ob ein Service selbst erbracht oder von einem externen Anbieter bezogen wird (make or buy), von strategischer Natur. Häufig wird argumentiert, dass Services, die keine besondere strategische Bedeutung für das Unternehmen haben (so genannte Commodity-Services), an externe Anbieter ausgelagert werden können.[7]

Zu den wichtigen Merkmalen einer SOA gehört die *Wiederverwendbarkeit* von Services, die erhebliche ökonomische Potenziale in sich birgt. Vor allem die Vermeidung von Redundanzen gilt als Kosteneinsparmöglichkeit. Die unternehmensweit einheitliche Verwendung eines Services spart Lizenzkosten, Entwicklungskosten und Unterhaltungskosten.[8] An dieser Stelle sei nicht verschwiegen, dass gerade die Wiederverwendung ein großes Problem in der betrieblichen Praxis darstellt. So zeigt beispielsweise Hagen am Beispiel der Credit Suisse, dass die durchschnittliche Wiederverwendungsrate der eingesetzten Services lediglich bei 1,7 liegt.[9]

---

[5] Becker, J./Kugeler, M./Rosemann, M. (2008)

[6] Starke, G./Tilkov, S. (2007); Papazoglou, M.P. (2003)

[7] Allen, P. (2006)

[8] Arsanjani, A. et al. (2008); Böhmann, T./Krcmar, H. (2005)

[9] Hagen, C. (2003)

Die vielfach geforderte *lose Kopplung* von Services bedeutet, dass Interdependenzen zwischen verschiedenen Services so gering wie möglich sein sollten und dadurch eine möglichst flexible Kombination der gleichen erfolgen kann.[10] Dabei kann grundsätzlich zwischen einer Orchestrierung von Services durch eine zentrale Instanz und einer Choreographie unterschieden werden. Letztere entsteht durch den sequentiellen Aufruf anderer Services aus dem zuletzt ausgeführten Service heraus. Hierbei gibt es keine zentrale Steuerung des Ablaufs.[11] Während die Abhängigkeit der Services untereinander dadurch auf ein Mindestmaß beschränkt wird, sollte die innere Kohäsion, d.h. die fachliche Zusammengehörigkeit der in einem Service vereinten Funktionalitäten, möglichst groß sein.[12]

Das Konzept serviceorientierter Architekturen ist dabei nicht auf eine spezielle Technologie beschränkt, d.h. es kann *plattformunabhängig* implementiert werden und ermöglicht so den unternehmensübergreifenden Einsatz von Services.[13] Durch die vorhergehende Standardisierung werden Schnittstellenprobleme auf ein sehr geringes Maß reduziert. Daher kann ein Insourcer durch das Anbieten von Services für viele Unternehmen Skaleneffekte realisieren.

Somit unterstützen serviceorientierte Architekturen nicht nur die agile und flexible Entwicklung von Geschäftsprozessen innerhalb eines Unternehmens, sondern über die gesamte Wertschöpfungskette hinweg. Der strategische Vorteil besteht für Unternehmen primär darin, dass notwendige Weiterentwicklungen an Geschäftsprozessen nicht mehr durch Anwendungen und Technologien beschränkt sind, sondern dass die Prozesse schnell und dynamisch an neue Kundenwünsche und -anforderungen angepasst werden können. Es erfolgt eine Trennung von Information, Anwendungen und Infrastruktur, die eine Dekonstruktion der Wertschöpfungskette unterstützt und vorantreibt.[14]

1.1.2 Serviceidentifikation

Mittlerweile gibt es zahlreiche Erfahrungen mit der Implementierung serviceorientierter Architekturen. In vielen Unternehmen wurde der Einsatz stark durch IT-Abteilungen vorangetrieben. Zunehmend setzt sich allerdings die Erkenntnis durch, dass die fachliche, auf Geschäftsprozessen basierende Identifikation und Gestaltung von Services eine notwendige Voraussetzung für den Erfolg einer SOA ist.[15] Bisherige in Literatur und Praxis verwendete Methoden sind häufig sehr technisch geprägt und teilweise sogar noch der Welt objektorientierter Programmierung zuzuordnen (siehe Artikel 1). Insofern besteht noch erheblicher Handlungsbedarf bei der Übertragung fachlicher Anforderungen im Rahmen der technischen Umsetzung einer SOA.

---

[10] Reinheimer, S. et al. (2007)

[11] Josuttis, N. (2008); Arsanjani, A. et al. (2008)

[12] Bieberstein, N. et al. (2006)

[13] Josuttis, N. (2008)

[14] Bieberstein, N. et al. (2006)

[15] Legner, C./Heutschi, R. (2007); Durst, M./Daum, M. (2007)

Die Serviceidentifikation ist eine der ersten Aktivitäten des Service-Engineering (Abbildung 2). Sie ist besonders wichtig, da sich Fehler, die hier gemacht werden, später in der Design- und Integrationsphase niederschlagen. Die notwendige Nacharbeit führt infolgedessen zu zusätzlichen Kosten.[16]

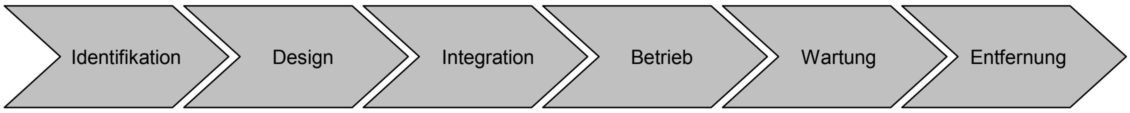

Abbildung 2: Ablauf des Service-Engineering

Bisher entwickelte Methoden zur Identifikation von Services beruhen teils auf verschiedenen SOA-Philosophien. Während die einen SOA als eine Art Middleware verstehen, betrachten andere SOA als ein Konzept zur flexiblen Organisation von Informationssystemen. Die vorliegende Dissertation orientiert sich an letzterem Verständnis.

Eine Folge der uneinheitlichen Perspektiven ist das Fehlen allgemein anerkannter Kategorien und Hierarchien für Services sowie einer einheitlichen Vorgehensweise zur Identifikation von Services. Nadhan beispielsweise schlägt einen Bottom-up-Ansatz vor, der, von der Informationstechnologie ausgehend, bereits vorhandene Services identifiziert.[17] Erl hingegen verfolgt einen Top-down-Ansatz, indem er die Geschäftsprozesse analysiert und daraus Teilprozesse oder Prozessfragmente ableitet, die später durch Services repräsentiert werden sollen.[18] Andere Autoren vereinen beide Herangehensweisen in einer hybriden Strategie, die auch als „meet in the middle approach“ bezeichnet wird.[19] In vielen Fällen wird eine hybride Herangehensweise, die Elemente beider Ansätze beinhaltet, zu besseren Lösungen führen, da die gänzliche Nichtbeachtung eines der Ansätze zu einer lückenhaften Serviceidentifikation führen wird. Die im Rahmen dieser Dissertation entwickelte Meta-Methode wird daher in der Regel in Methoden resultieren, die Fragmente beider Ansätze in einer – auf Basis der gegebenen Rahmenbedingungen – sinnvollen Weise kombinieren.

Die Granularität von Services hängt unter anderem von der Wahl der Identifikationsmethode ab. Umgekehrt kann man je nach Ziel der Serviceidentifikation (bzw. Ziel der SOA-Implementierung im Allgemeinen) auch eine passende Identifikationsmethode wählen. Ein sehr granularer, d.h. im Umfang stark beschränkter Service (wie z.B. eine Adressabfrage), ist möglicherweise so generisch, dass er in sehr vielen unterschiedlichen Prozessen eingesetzt werden kann. Dies ermöglicht die von SOA-Befürwortern häufig hervorgehobene *Wiederverwendbarkeit*. Wenn letztere im Mittelpunkt der Betrachtung steht, muss geprüft werden, ob es sinnvoll ist, bestehende Geschäftsprozesse so zu ändern, dass sie „ähnlicher“ werden und die gleichen Services nutzen können. Dabei stellt sich immer die Frage, ob der Nutzen aus der Wiederverwendung der Services größer als der Aufwand für die Prozessumgestaltung ist, denn nur dann lohnt sich

[16] Inaganti, S./Behara, G.K. (2007)

[17] Nadhan, E.G. (2004)

[18] Erl, T. (2004)

[19] Ivanov, K./Stähler, D. (2005); Zacharias, R. (2005)

die Standardisierung. Die Beantwortung dieser und ähnlich gelagerter Fragen ist nicht immer widerspruchsfrei und erzeugt somit ein Spannungsfeld, in dem die Serviceidentifikation steht.

1.1.3 Methoden-Engineering in der Serviceidentifikation

Methoden beschreiben „eine planmäßige Art und Weise des Handelns mit überprüfbaren Ergebnissen.“[20] Sie bestehen aus Regeln, die dazu dienen, Entwicklungsaktivitäten und -ergebnisse in einer systematischen und strukturierten Weise zu verknüpfen. Seit vielen Jahren gibt es Bemühungen, die Konstruktion von Methoden in der Softwareentwicklung zu unterstützen, um ein hohes Qualitätsniveau zu garantieren. Daraus entstand die Disziplin des Methoden-Engineering. Sie unterstützt die Konstruktion, das Design und die Anpassung von Methoden im Bereich der Entwicklung von Informationssystemen.[21]

Gerade die Anpassungsfähigkeit von Methoden spielt eine immer wichtigere Rolle, da es sehr unwahrscheinlich ist, dass eine universelle Methode in jeder Projektsituation unverändert anwendbar ist.[22] Eine Methode sollte daher konfigurierbar sein. Aus diesem Grund entwickelte sich das Gebiet des Situational Method Engineering.[23] Um die Anpassungsfähigkeit einer Methode zu ermöglichen, werden sogenannte Fragmente konstruiert und anschließend in Abhängigkeit der Situation kombiniert.[24] Die Definition von Zusammenhängen und Interdependenzen von Methodenfragmenten ist eine der schwierigsten Herausforderungen des SME. Input- und Outputbeziehungen von Methodenfragmenten müssen beispielsweise präzise definiert sein.[25] Artikel 11 der vorliegenden Dissertation beschreibt ausführlich die Grundlagen des SME und ihre Übertragbarkeit auf die speziellen Gegebenheiten der Serviceidentifikation.

Die situative Anpassungsfähigkeit von Methoden fehlt bislang weitestgehend in der Literatur zur Serviceidentifikation (vgl. Artikel 9). Daher sollen im Folgenden Konzepte des SME zur Entwicklung von Serviceidentifikationsmethoden genutzt werden. Die Definition von Situationen auf dem Gebiet der Serviceidentifikation, die eine wichtige Voraussetzung für das SME darstellt, wird in Artikel 3 detailliert erläutert.

1.2 Forschungslücke und Zielsetzung der Arbeit

Nach der Erläuterung des Hintergrunds sollen in diesem Abschnitt zunächst die Problemstellung präzisiert und die Forschungsfrage formuliert werden. Anschließend wird aufgezeigt, welchen Problemlösungsbeitrag die in der kumulativen Dissertation zusammengefassten Artikel liefern.

[20] Greiffenberg, S. (2003), S. 10

[21] Brinkkemper, S. (1996)

[22] Aydin, M.N. (2007); Brooks, F.P. (1987); Fitzgerald, B./Russo, N.L./O'Kane, T. (2003)

[23] Harmsen, F./Brinkkemper, S./Oei, H. (1994)

[24] Arni-Bloch, N./Ralyté, J. (2008); Ralyté, J./Rolland, C. (2001)

[25] Harmsen, F./Brinkkemper, S./Oei, H. (1994)

1.2.1 Problemstellung

Nach anfänglicher Zurückhaltung hat sich mittlerweile in vielen Unternehmen die Erkenntnis durchgesetzt, dass serviceorientierte Architekturen sowohl altbekannte Probleme wie die Integration von bestehenden (Legacy-)Systemen zu lösen im Stande sind, als auch die Wettbewerbsfähigkeit des Unternehmens entscheidend verbessern. Dementsprechend fand SOA in den letzten Jahren eine immer größere Verbreitung.[26]

Durst und Daum fanden bei einer Umfrage heraus, dass die „Schaffung der notwendigen Voraussetzungen aus fachlicher Sicht" der am häufigsten genannte Erfolgsfaktor einer serviceorientierten Architektur ist.[27] Auch Thomas et al. stellen fest, dass die Qualität fachlicher Modelle „einen nachhaltigen und signifikanten Einfluss auf die modellbasierte Gestaltung einer SOA"[28] hat. Die fachliche Identifikation von Prozessschritten, die anschließend als Services umgesetzt werden, muss daher deutlicher in den Mittelpunkt treten, als dies in der Forschung aktuell geschieht.

In der Literatur werden bereits verschiedene Ansätze zur Identifikation von Services diskutiert. Einige von ihnen haben ihre Wurzeln im stark ingenieurtechnisch geprägten Bereich der Fertigungsindustrie[29], andere betrachten die Modularisierung im IT-Dienstleistungsbereich.[30] Auch unter den Ansätzen, die sich mit Finanzdienstleistungen beschäftigen, gibt es neben deutlich fachlich orientierten Varianten[31] auch stark technisch getriebene Ansätze.[32] Der erste Artikel „Identification of Business Services - Literature Review and Lessons Learned" vergleicht diese Ansätze zur Serviceidentifikation auf Basis ausgewählter Kriterien. Insbesondere aus der Perspektive des Methoden-Engineering werden einige Schwächen deutlich. Die auffällige Absenz von Rollenzuweisungen in existierenden Ansätzen scheint zunächst kein großes Problem darzustellen. Implizit gehen alle Autoren offenbar davon aus, dass die Geschäftsprozessmodellierung von der Fachabteilung oder verantwortlichen „Prozess-Eignern" durchgeführt wird, während die Prüfung der technischen Umsetzung der IT-Abteilung obliegt. Die Praxis zeigt jedoch, dass diese scheinbar triviale Erkenntnis nicht selbstverständlich ist und es dadurch immer wieder zu Problemen und Fehleinschätzungen bei der Serviceidentifikation kommen kann.

Ein darüber hinaus gehender Vergleich in Artikel 11 identifiziert die mangelnde Konfigurierbarkeit als eine weitere Schwäche vorhandener Serviceidentifikationsmethoden. Der in ihnen verwendete „One-size-fits-all"-Ansatz kann dabei verschiedene Folgen haben. Einerseits besteht die Gefahr, dass durch unnötigerweise durchgeführte Aktivitäten Ressourcen verschwendet werden. Andererseits kann durch die falsche oder im Kontext unangemessene Ausführung bestimmter Methodenteile (bspw. durch den Ein-

---

[26] Eckert, J. et al. (2009); Rabhi, F.A. et al. (2006); Hagen, C. (2003)

[27] Durst, M./Daum, M. (2007)

[28] Thomas, O./Leyking, K./Dreifus, F. (2007), S. 46

[29] Klose, K./Knackstedt, R./Beverungen, D. (2007)

[30] Böhmann, T./Krcmar, H. (2005)

[31] Kohlmann, F./Alt, R. (2007)

[32] Winkler, V. (2007)

satz falsch qualifizierter Mitarbeiter) ein unbefriedigendes Ergebnis zustande kommen. Zudem können Umstände und Restriktionen eines Projekts (so genannte Kontextfaktoren) dazu führen, dass bestimmte Teile einer Methode schlicht nicht anwendbar sind, was den Projektleiter inmitten eines Projekts vor erhebliche Probleme stellen kann.

Vor diesem Hintergrund lässt sich die Forschungsfrage der Dissertation wie folgt formulieren:

Wie kann die Konstruktion von konfigurierbaren Serviceidentifikationsmethoden unterstützt werden?

1.2.2 Lösungsbeitrag

Neben den fehlenden Rollenzuweisungen in existierenden Methoden wurde im vorherigen Abschnitt insbesondere die fehlende Möglichkeit zur Konfiguration von Methoden als Problem identifiziert. Daher soll im Rahmen der Dissertation eine Meta-Methode entwickelt werden, die unter Berücksichtigung von Kontextfaktoren und den Implementierungszielen einer SOA die Konstruktion von geeigneten, konfigurierbaren Methoden zur Serviceidentifikation unterstützt. Dabei werden für die Serviceidentifikation spezifische Methodenfragmente verwendet, um Methoden an konkrete Situationen anzupassen. Die verschiedenen Artikel bzw. Arbeitsberichte tragen auf sehr unterschiedliche Weise zur Unterstützung der Methodenkonstruktion bei.

Artikel 2 zeigt, wie die im Methoden-Engineering verwendeten Elemente mit Hilfe von Grounded Theory-Techniken besser empirisch fundiert werden können. Artikel 3 beschreibt die Übertragung von Ideen des SME auf das Forschungsgebiet der Serviceidentifikation.

In den Artikeln 4, 5 und 6 werden fachliche Aspekte, wie Strategie und IT-Governance in die Serviceidentifikation integriert. Dazu werden spezielle Methodenelemente wie Service Maps oder eine Service Design Unit eingehend behandelt. Diese werden später wichtige Bestandteile der Methodenfragmente sein.

Situationen spielen im SME eine entscheidende Rolle. Die erste Säule der präsentierten Meta-Methode ist daher die Identifikation von Situationen auf dem Gebiet der Serviceidentifikation (Abbildung 3). Dazu werden erstens relevante Kontextfaktoren und mögliche Ausprägungen der gleichen identifiziert. Zweitens müssen SOA-Implementierungsziele beachtet werden (Artikel 7, 8 und 9). Gemeinsam determinieren diese beiden Bestandteile eine Situation (vgl. Artikel 3).

Die Verwendung von Methodenfragmenten, folgend den Ideen des Situational Method Engineering, unterstützt dabei die Konfigurierbarkeit und bildet die zweite große Säule der Meta-Methode. Die Beschreibung von Methodenelementen wie Aktivitäten, Techniken, Ergebnissen und insbesondere Rollen ist daher ein wichtiger Beitrag zur Problemlösung. Artikel 5 und 6 führen beispielsweise neue Rollen ein, die für die Durchführung bestimmter Aktivitäten in den entsprechenden Methodenfragmenten verantwortlich sind. In Artikel 11 werden 16 solcher Methodenfragmente vorgestellt. Die Entwicklung dieser Fragmente, die analog zu den Situationen speziell auf das

Gebiet der Serviceidentifikation zugeschnitten sind, basiert auf den Prinzipien des „Methoden-Re-engineering“ und der „Ad-hoc-Konstruktion“ (vgl. Artikel 10 und 11).[33]

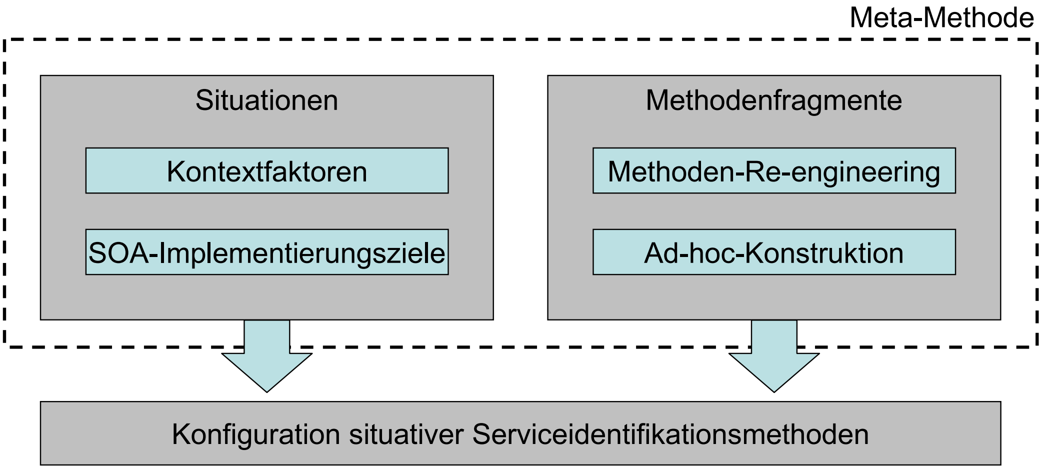

Abbildung 3: Umfang und Grenzen der Meta-Methode

Abbildung 3 zeigt, dass die Meta-Methode sowohl die Identifikation von Situationen als auch das Design von Methodenfragmenten umfasst. Beide sind notwendig für die Konfiguration von situativen Methoden.

Anhand eines Beispiels wird in Artikel 11 die Konstruktion einer Methode zur Serviceidentifikation demonstriert. Allgemein gültige Regeln für die Konstruktion solcher Methoden bedürfen weiterer Forschungsarbeit bspw. auf Basis zusätzlicher Fallstudien und befinden sich außerhalb des Rahmens dieser Dissertation.

1.3 Aufbau der Arbeit

Diese kumulative Dissertation setzt sich aus fünf Gebieten zusammen, die aus jeweils ein bis drei Artikeln bzw. Arbeitsberichten bestehen. Die Nummerierung der Artikel erfolgte aufgrund ihrer Zuordnung zu den jeweiligen Themenfeldern und spiegelt nicht die chronologische Ordnung ihrer Erscheinung wieder. Abbildung 4 zeigt alle Bestandteile der kumulativen Dissertation und die Zuordnung der Artikel zu den jeweiligen (grau hinterlegten) Gebieten.

[33] Ralyté, J. (2004)

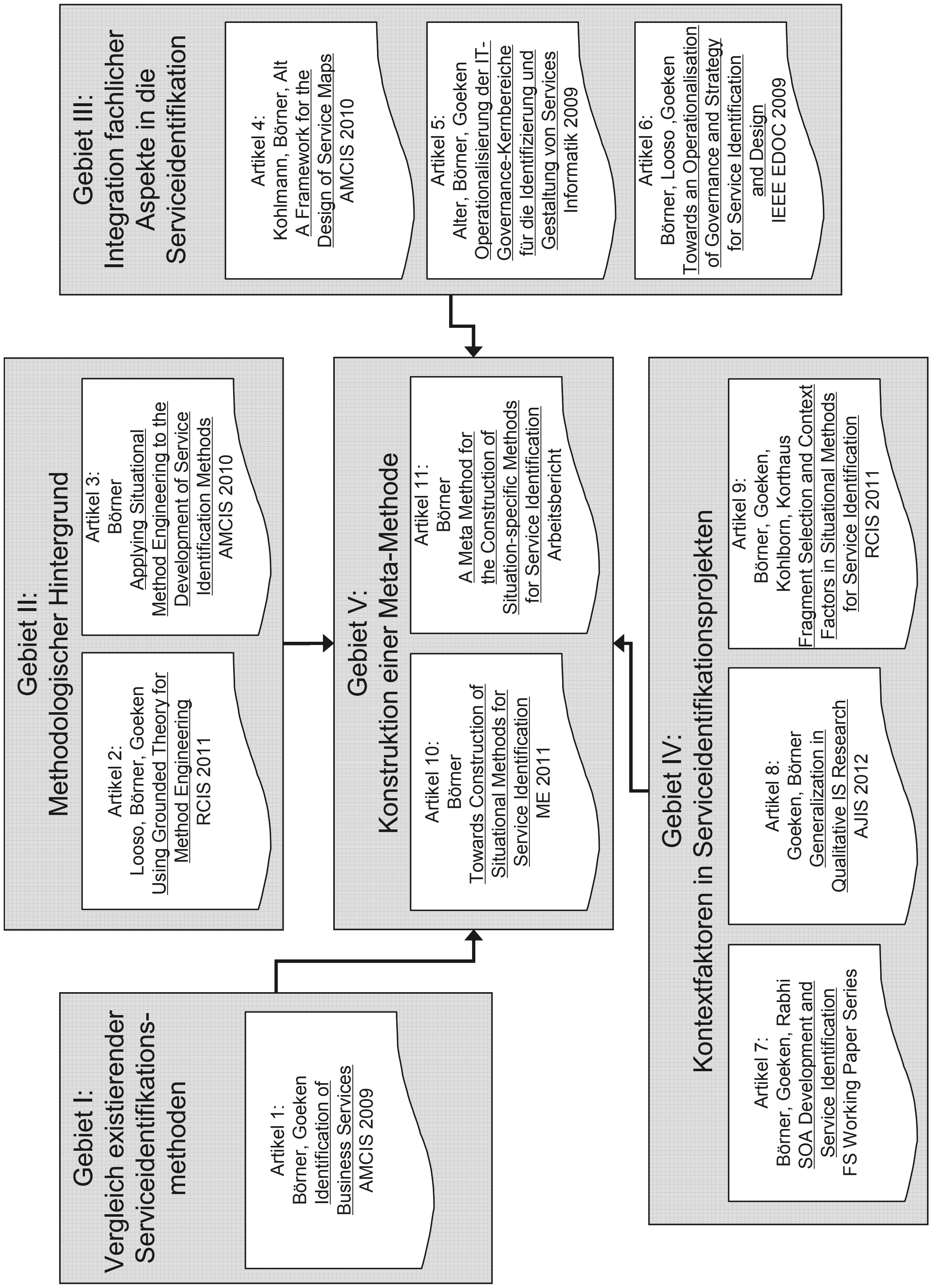

Abbildung 4: Bestandteile der kumulativen Dissertation

Gebiet I besteht aus der Literaturanalyse und umfasst den Artikel „Identification of Business Services“ (Börner/Goeken), der in den Proceedings der 15. Americas Conference on Information Systems 2009 veröffentlicht wurde. Diese Literaturanalyse beschreibt den Status quo existierender Serviceidentifikationsmethoden, offenbart Schwächen der gleichen und dient somit als Grundlage für die Entwicklung einer Meta-Methode zur Konstruktion von Serviceidentifikationsmethoden.

Das zweite Gebiet befasst sich mit den methodologischen Grundlagen der Arbeit und diskutiert insbesondere die Disziplin des Methoden-Engineering und ihre Bedeutung für die Dissertation. Dazu gehört einerseits die empirisch gestützte Entwicklung von Methoden, die in Artikel 2 „Using Grounded Theory for Method Engineering“ (Looso/Börner/Goeken) beschrieben wird. Dieser erschien in den Proceedings der fünften IEEE International Conference on Research Challenges in Information Science 2011. Andererseits ist die Anwendung des Konzepts des Methoden-Engineering auf den Bereich der Serviceidentifikation, die im Artikel „Applying Situational Method Engineering to the Development of Service Identification Methods“ (Börner) beschrieben wird, Bestandteil dieses Gebiets. Dieser dritte Artikel wurde in den Proceedings der 16. Americas Conference on Information Systems 2010 veröffentlicht.

Gebiet III beschäftigt sich mit der Einbeziehung fachlicher Aspekte in die Serviceidentifikation. Artikel 4 „A Framework for the Design of Service Maps“ (Kohlmann/Börner/Alt) wurde in den Proceedings der 16. Americas Conference on Information Systems 2010 veröffentlicht. Er beschreibt den Entwurf von Service Maps, die in bestimmten Situationen wichtiger Bestandteil einer Serviceidentifikationsmethode sein können. Die beiden Artikel „Operationalisierung der IT-Governance-Kernbereiche für die Identifizierung und Gestaltung von Services“ (Alter/Börner/Goeken) und „Towards an Operationalisation of Governance and Strategy for Service Identification and Design“ (Börner/Looso/Goeken) bauen aufeinander auf. Ersterer zeigt, wie IT-Governance-Aspekte in die Identifikation von Services einbezogen werden können. Letzterer erweitert diesen Ansatz unter Beachtung von strategischen Unternehmenszielen. Sie wurden in den Springer Lecture Notes in Informatics „Informatik 2009 – Im Focus das Leben“ bzw. in den Proceedings der 13. IEEE International EDOC Conference 2009 veröffentlicht.

Ein bedeutendes Element der in dieser Arbeit entwickelten Meta-Methode sind Kontextfaktoren. Das vierte Gebiet enthält daher drei Artikel, die auf unterschiedliche Weise solche Faktoren identifizieren und ihre Auswirkung auf Serviceidentifikationsmethoden analysieren. Der in der Frankfurt School Working Paper Series veröffentlichte Arbeitsbericht „SOA Development and Service Identification - A Case Study on Method Use, Context and Success Factors“ (Börner/Goeken/Rabhi) befasst sich mit der Verwendung von Serviceidentifikations- und Softwareentwicklungs-Methoden. Dazu betrachtet er die Rolle, die Kontextfaktoren in einer Fallstudie – dem ADAGE-Projekt – spielten. Aufbauend darauf diskutiert der Artikel „Generalization in Qualitative IS Research - Approaches and Their Application to a Case Study on SOA Development“ (Goeken/Börner) die methodische Unterstützung von Generalisierungen in der qualitativen Forschung anhand dieser Fallstudie. Er wurde im Australasian Journal of Information Systems veröffentlicht. Artikel 9 „Fragment Selection and Context Factors in Situational Methods for Service Identification“ (Börner/Goeken/Kohlborn/Korthaus) wurde in den Proceedings der fünften IEEE International Conference on Research Challenges in Information Science 2011 veröffentlicht. Unter Verwendung einer weiteren Fallstudie wird darin die Auswirkung von Kontextfaktoren auf die Selektion von Methodenfragmenten untersucht.

Das fünfte Gebiet bildet schließlich den Kern der vorliegenden Dissertation. Artikel 10 „Towards Construction of Situational Methods for Service Identification“ (Börner) ist in dem Herausgeberwerk „ME 2011“ erschienen. Er beleuchtet insbesondere die Bedeutung von Kontextfaktoren und Implementierungszielen serviceorientierter Architekturen für die entwickelte Meta-Methode, baut dabei insbesondere auf Artikel 3 auf und schlägt schließlich die Brücke zu Artikel 11 „A Meta Method for the Construction of Situation-specific Methods for Service Identification“ (Börner), der als Arbeitsbericht zunächst unveröffentlicht bleibt. In diesem elften Artikel werden Erkenntnisse und Ergebnisse aller zuvor veröffentlichten Artikel konsolidiert und die Meta-Methode zur Entwicklung situationsspezifischer Methoden zur Serviceidentifikation beschrieben. Des Weiteren umfasst dieser Artikel ein Fazit sowie eine ausführliche Diskussion über Grenzen und Beschränkungen des Forschungsprozesses und der Meta-Methode.

Jedes Gebiet beginnt jeweils mit einer kurzen Einleitung, in der die zu dem Gebiet gehörenden Artikel thematisch eingeordnet werden. Es folgen die entsprechenden Artikel, bevor am Ende eines jeden Kapitels ein Fazit die Bedeutung der gleichen für die gesamte Dissertation zusammenfasst.

1.4 Wissenschaftliche Vorgehensweise

Die hierin beschriebene Entwicklung einer Meta-Methode für Serviceidentifikationsmethoden basiert auf einem hybriden Ansatz, der mehrere Forschungsmethoden kombiniert, um ein tieferes Verständnis des Forschungsgebiets zu erlangen.[34] Die Wirtschaftsinformatik setzt eine Reihe von verschiedenen Forschungsmethoden zum Erkenntnisgewinn ein, die im Folgenden auch Bestandteil dieser Arbeit sein werden. Dazu zählen Fallstudien, Referenzmodellierung[35] sowie der Literaturreview.[36] Nach Ansicht des Verfassers lässt sich das Methoden-Engineering in dieses Spektrum zweckmäßig einordnen, da Methoden, die in dieser Forschungsrichtung entwickelt werden, wie Referenzmodelle normativen (Referenz-)Charakter haben.

Im Vordergrund steht die Schaffung eines Artefakts (nämlich der Meta-Methode), das der Praxis als Handlungsempfehlung dienen soll. Insofern kann die Wirtschaftsinformatik im Rahmen dieser Arbeit als eine angewandte Disziplin, ähnlich einer Ingenieurswissenschaft betrachtet werden. Das Anliegen, mittels dieser Meta-Methode eine Problemstellung besser lösen zu können als bisher, impliziert bereits den präskriptiv-wertenden Charakter der Arbeit.

[34] Mingers, J. (2001)

[35] Wilde, T./Hess, T. (2007)

[36] Fettke, P. (2006)

In der gestaltungsorientierten Forschung (Design Science) werden die Forschungsergebnisse wie beispielsweise Modelle, Methoden oder Prototypen als *Artefakte* bezeichnet.[37] Letztere können nicht gegen die Realität getestet werden, da es sich um „Soll-Modelle“ handelt. Darum fordern Hevner et al. eine besondere Sorgfalt bei der Entwicklung dieser Artefakte und präsentieren sieben Richtlinien gestaltungsorientierter Forschung, die auch Möglichkeiten der Evaluation aufzeigen.[38] Da eine ausführliche Behandlung der gleichen in Artikel 11 zu finden ist, wird an dieser Stelle auf nähere Erläuterungen verzichtet.

Eine Meta-Methode, wie sie im Rahmen dieser Dissertation entwickelt wird, stellt ein solches Artefakt dar. Daher dienen Hevner et al.’s Richtlinien als Basis für die Entwicklung dieser Meta-Methode und spielen in beinahe allen Artikeln, die Bestandteil dieser kumulativen Dissertation sind, eine Rolle. Dabei ist jedoch zu beachten, dass es sich bei den Richtlinien nicht um strikte Vorgaben, sondern vielmehr um Orientierungshilfen handelt. Die Autoren selbst schreiben dazu: „We advise against mandatory or rote use of the guidelines.“[39] Die Bedeutung dieser Richtlinien für diese Arbeit wird in Artikel 11 detailliert dargestellt.

Abbildung 4 zeigt die Bestandteile der kumulativen Dissertation. Den verschiedenen Gebieten können jeweils entsprechende Forschungsmethoden zugeordnet werden. Zu Beginn des Dissertationsprojekts stand die Themenfindung und Strukturierung des Problems. Aufbauend auf einem *Literaturreview*, dessen Ergebnisse in Gebiet I dargestellt sind, konnte eine Forschungslücke identifiziert und das zu lösende Problem definiert werden. Zwei *Fallstudien* in Verbindung mit Techniken der Grounded Theory wurden genutzt, um Kontextfaktoren zu identifizieren, die ein wesentlicher Bestandteil der Meta-Methode sind (Gebiete II und IV). Der nächste Schritt war die durch Konzepte des *Situational Method Engineering* unterstützte Entwicklung (*Modellierung*) einer Meta-Methode, die die dargestellten Probleme zu lösen vermag (Gebiete II, III und V). Schließlich wird in einem Anwendungsbeispiel in Artikel 11 demonstriert, dass die entwickelte Meta-Methode in der Praxis einsetzbar ist.

Zur Konfiguration des Design Science-Forschungsprozesses schlagen Peffers et al. den in Abbildung 5 dargestellten Ablauf vor.[40]

---

[37] Simon, H.A. (1996)

[38] Hevner, A.R. et al. (2004), S. 83

[39] Hevner, A.R. et al. (2004), S. 82

[40] Peffers, K. et al. (2008)

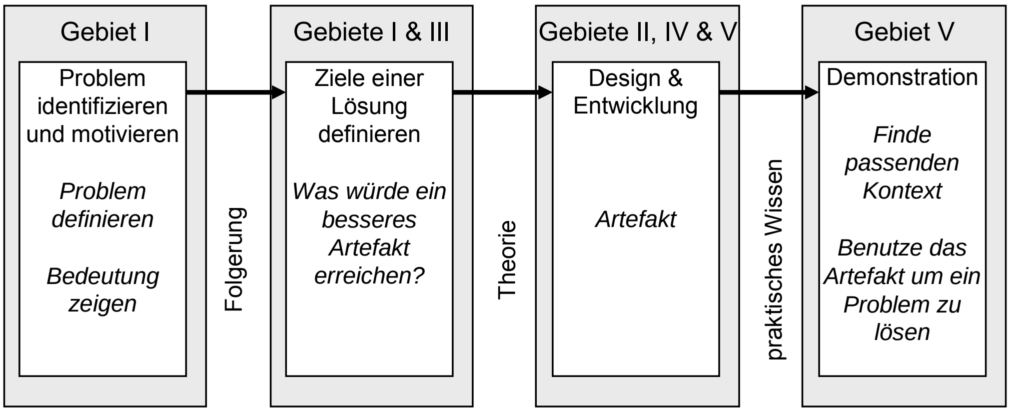

Abbildung 5: Design Science-Forschungsprozess (in Anlehnung an Peffers et al. (2008), S. 54)

Auch hierin lassen sich die verschiedenen Artikel und Gebiete dieser kumulativen Dissertation einordnen. Die Identifizierung und Motivation des Problems geschieht im ersten Gebiet. Der Definition von Zielen widmen sich die Gebiete I und III. Grundlagen für das Design und die Entwicklung der Meta-Methode werden in den Gebieten II, IV und V gelegt. Schließlich folgt eine Demonstration an einem Beispiel in Gebiet V.

1.4.1 Literaturreview

Ein intensiver Literaturreview, der in der gestaltungsorientierten Forschung notwendige Grundlage für den Prozess des Problemlösens ist, war der erste Schritt der Arbeit. Sie diente zunächst dazu, einen Überblick über die in Theorie und Praxis relevanten Aspekte auf dem Gebiet serviceorientierter Architekturen zu gewinnen. Dies ermöglichte die Definition des Problems und der Ziele, die mit der Dissertation verfolgt werden sollen. Gleichzeitig dienen die Ergebnisse, die in Artikel 1 vorgestellt werden, nun als Basis für die Entwicklung der Meta-Methode. Der Review umfasste ein breites Spektrum von deutsch- und englischsprachiger Literatur in den Bereichen SOA, Serviceidentifikation, Methoden-Engineering und gestaltungsorientierte Forschung. Zu den Quellen gehören Lehrbücher, Artikel in Fachzeitschriften und Konferenzbeiträge.

Insbesondere bereits existierende Methoden zur Identifikation von Services wurden intensiv untersucht und auf Basis eines Kriterienkatalogs in Artikel 1 verglichen. Die Kriterien wurden aus den analysierten Quellen extrahiert und um eigene – für die Entwicklung der Meta-Methode wichtige – ergänzt. Im Laufe des Dissertationsprojekts wurden weitere Serviceidentifikationsmethoden berücksichtigt und bei der Entwicklung der Meta-Methode mit einbezogen (vgl. Artikel 11).

1.4.2 Fallstudien

Ein entscheidender Beitrag der Meta-Methode besteht darin, relevante Kontextfaktoren für das Gebiet der Serviceidentifikation bereitzustellen. Da die Identifizierung dieser Faktoren ein exploratives Ziel darstellt, wurden gezielt Fallstudien als Forschungsmethode gewählt, um den Ergebnissen eine empirische Grundlage zu geben. Fallstudien sind deshalb angemessen, weil sie Beschreibungen von Phänomenen liefern und aktuelle Probleme in ihrer natürlichen Umgebung erforschen.[41] Zudem sind Fallstudien besonders dann von Bedeutung, wenn sich die Forschung in einem frühen, formativen Stadium befindet.[42] Dies trifft auf das Gebiet serviceorientierter Architekturen zu, denn trotz der Popularität von SOA in den vergangenen Jahren ist über methodische Ansätze bei der Entwicklung serviceorientierter Architekturen und der Identifikation von Services wenig bekannt.[43] Dieses Phänomen korrespondiert zu den wenigen empirischen Erkenntnissen im Bereich der System- und Software-Entwicklung im Allgemeinen.[44]

Die beiden Fallstudien, die für diese Dissertation benutzt wurden, werden in den Artikeln 7, 8 und 9 ausführlicher beschrieben. Die in der Fallstudie des ADAGE-Projekts geführten Interviews wurden transkribiert und gemeinsam mit allen vorhandenen schriftlichen Dokumentationen und Präsentationen analysiert. Auf dieser Basis wurden Kontextfaktoren ermittelt und mit verwandter Literatur verglichen. Die Identifizierung der Faktoren erfolgte iterativ unter Verwendung von Techniken aus der Grounded Theory, wie bspw. Open Coding und Axial Coding.[45]

Viele Autoren kritisieren Einzelfallstudien aus verschiedenen Gründen. Lee bspw. kritisiert, dass sie den Forscher nicht in die Lage versetze, kontrollierte Beobachtungen zu machen, kontrollierte Ableitungen zu begründen, eine Wiederholbarkeit sicherzustellen oder eine Generalisierbarkeit zu ermöglichen.[46] Darke et al. hingegen argumentieren, dass eine Einzelfallstudie als Basis für Erklärungen dienen kann, warum ein Phänomen auftritt.[47] Yin betrachtet Einzelfallstudien dann als angemessen, wenn sie einen kritischen Fall, einen extremen oder einzigartigen Fall, einen repräsentativen Fall bzw. einen enthüllenden Fall beschreiben.[48] Die im Rahmen dieser Dissertation durchgeführte Einzelfallstudie des ADAGE-Projekts stellt zweifelsohne einen extremen Fall dar. Die Einzelheiten sind in den Artikeln 7 und 8 geschildert und werden daher an dieser Stelle nicht näher erläutert.

1.4.3 Methoden-Engineering im Design Science-Prozess

Ähnlich der von Wilde und Hess in ihrem Katalog von Forschungsmethoden enthaltenen Referenzmodellierung[49], wird das Methoden-Engineering in dieser Dissertation dazu verwendet, eine Meta-Methode zu entwickeln. Ausgehend von Beobachtungen und bereits existierenden Methoden wird eine Gestaltungsvorlage erstellt, um zukünftig bessere Serviceidentifikationsmethoden zu konstruieren.

---

[41] Darke, P./Shanks, G./Broadbent, M. (1998); Yin, R.K. (2003)

[42] Benbasat, I./Goldstein, D.K./Mead, M. (1987)

[43] Luthria, H./Rabhi, F.A. (2009); Stebbins, R.A. (2001)

[44] Jarke, M. (2009)

[45] Strauss, A./Corbin, J.M. (1990)

[46] Lee, A.S. (1989)

[47] Darke, P./Shanks, G./Broadbent, M. (1998)

[48] Yin, R.K. (2003)

[49] Wilde, T./Hess, T. (2007)

Die situative Anpassungsfähigkeit von Methoden fehlt bislang weitestgehend in der Literatur zur Serviceidentifikation (vgl. Artikel 9). Die Modellierung mit Hilfe von Konzepten des SME ist daher zentraler Bestandteil bei der Entwicklung einer Meta-Methode zur Konstruktion von Serviceidentifikationsmethoden. Dementsprechend finden sich diese Konzepte in vielen Artikeln dieser kumulativen Dissertation wieder. Die Identifikation von Situationen, die eine wichtige Vorbedingung für das SME darstellt, wird in Artikel 3 detailliert erläutert. Artikel 7 und 9 analysieren Kontextfaktoren und Implementierungsziele und liefern somit die Grundlage für diese Situationen. Schließlich basiert die Entwicklung der Meta-Methode in den Artikeln 10 und 11 auf diesen Konzepten.

2. Gebiet I: Vergleich existierender Serviceidentifikationsmethoden

2.1 Einführung

Die Serviceidentifikation ist ein wichtiger Bestandteil des Service-Engineering. Daher hat sich bereits eine Reihe von Autoren mit Methoden zur Serviceidentifikation befasst. Grundlage einer Meta-Methode zur Entwicklung von Serviceidentifikationsmethoden ist die Analyse bereits existierender Ansätze. Der folgende Artikel „Identification of Business Services – Literature Review and Lessons Learned“ untersucht fünf in der Literatur vorhandene Ansätze aus der Fertigungsindustrie, dem IT-Dienstleistungssektor und der Finanzbranche. Dabei werden fachliche, ökonomische, technische sowie methodologische Aspekte berücksichtigt und Stärken und Schwächen der jeweiligen Methoden diskutiert.

2.2 Artikel 1: Identification of Business Services - Literature Review and Lessons Learned

Veröffentlicht: Börner, R., Goeken, M. (2009): Identification of Business Services - Literature Review and Lessons Learned, in: Proceedings of the 15th Americas Conference on Information Systems (AMCIS 2009), 06.-09.08.2009, San Francisco, Paper 106.

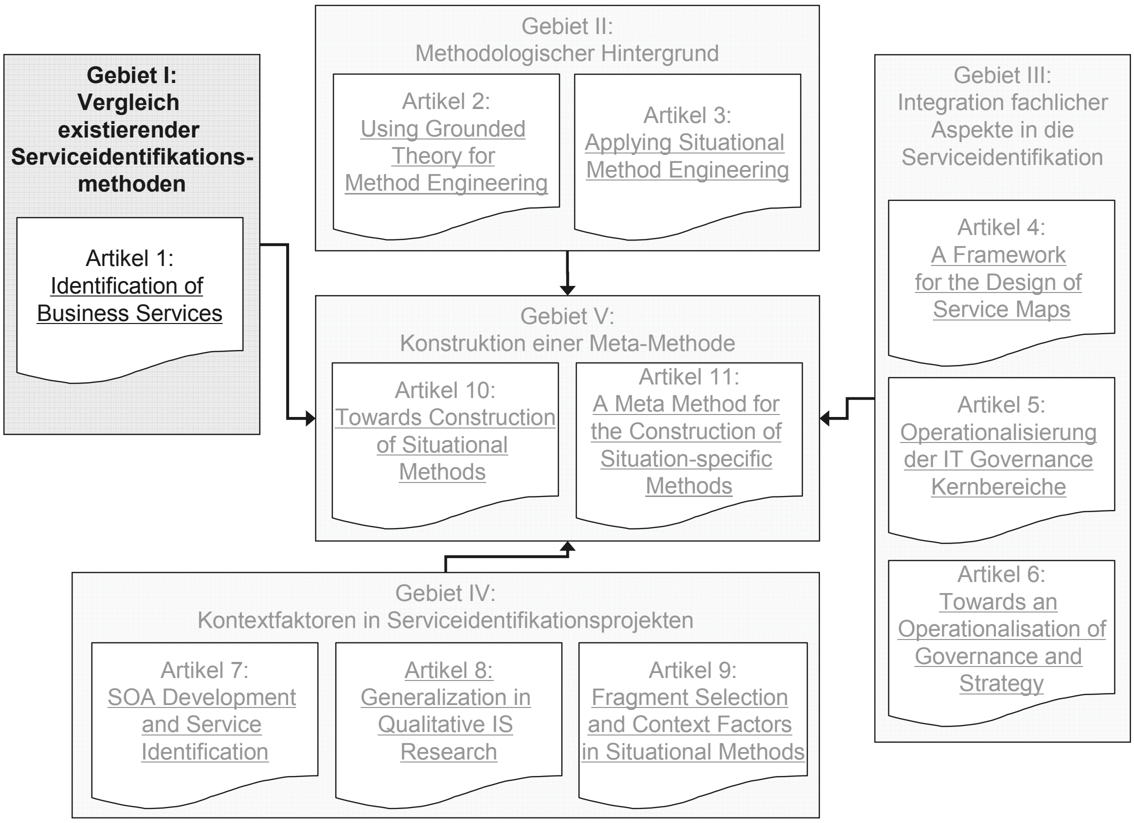

Abbildung 6: Artikel 1 im Kontext der kumulativen Dissertation

Identification of Business Services Literature Review and Lessons Learned

René Boerner
Frankfurt School of Finance & Management
ProcessLab
Sonnemannstrasse 9-11,
60314 Frankfurt, Germany
r.boerner@frankfurt-school.de

Matthias Goeken
Frankfurt School of Finance & Management
IT-Governance-Practice-Network
Sonnemannstrasse 9-11,
60314 Frankfurt, Germany
m.goeken@frankfurt-school.de

Abstract

Business-driven identification of services is a precondition for a successful implementation of service-oriented architectures (SOA). This article compares existing identification methods retrieved from related work and discusses the shortcomings. In particular, a lack of economic aspects constitutes a problem and leaves space for improvements. Finally, the paper proposes a process-oriented method of service identification. This approach incorporates the business point of view, strategic and economic aspects as well as technical feasibility.

Keywords

Service-Oriented Architecture, Service Identification, Business Process Modeling, Business Services.

SERVICE IDENTIFICATION AND DESIGN

Service-orientation is a new and highly recognized paradigm in enterprise architecture. There are lots of expected benefits related to SOA in a technical and in a business-oriented sense. Although the business-oriented benefits, like flexibility and reusability/standardization, are of high importance, up to now, development of SOAs is mainly technically driven and most approaches consider technical aspects in the first place. However, experience and empirical evidence show that technically-driven implementations often fail in realizing the mentioned business-oriented benefits (Hagen, 2003; Legner and Heutschi, 2007).

Hence, approaches which focus on business issues are of primary importance. For the last couple of years some authors have been looking at the identification of services, which is one of the first most important steps in an overall approach. As these services are strongly related to business processes, they are often termed “business services”. However, there is still a lack of common understanding of what services (and business services) are and which goals are to be achieved. Due to this, existing approaches for service identification differ significantly from one another. In chapter 2 we present a framework of several criteria in order to compare approaches found in literature. Their strengths and shortcomings are discussed in chapter 3. Based on these findings, requirements of a new method to identify services from a business point of view are presented in chapter 4. Particularly, method engineering aspects as a foundation for this new process-oriented service identification (POSI) will be discussed.

COMPARISON CRITERIA

In order to compare the approaches for service identification, a catalogue consisting of selected criteria is applied to give an overview of approaches currently discussed in related literature. Some criteria have already been used by other researchers (Allen, 2006; Erl, 2004; Josuttis, 2008); others have been added to complement the existing ones. Table 1 shows the selected criteria summarized to six groups for a better understanding. All criteria will be discussed in detail in this chapter.

Basic aspects	**Economic aspects**
Industry sector	Value creation
Understanding of services	Maintenance and operation costs
Service hierarchy	Testing effort for new functionality
Granularity	Vendor dependency
SOA paradigm	Demand-oriented QoS levels
Starting point and direction of derivation	Customer satisfaction
"Tools"	Individualization of products and services
Types of categorization	Specialization in core competences
Business aspects	Increase of product range
Consideration of strategic aspects	Internal services offered to external customers
Legal compliance	Scalability
Internal policies / IT governance	Time-to-Market
Service level agreements	SOA controlling
Goal	**Components of method engineering**
Supported object	Activities
SOA lifecycle	(SOA-)Roles
Functional similarity	Results
Technical aspects	Techniques
Orchestration vs. choreography	Sequence of activities
Customer interaction	**Guidelines of design science research**
Employee interaction	Documentation
Criteria of information technology	Research rigor
Call frequency	Method evaluation

Table 1. Criteria by Groups

Basic Aspects

The *industry sector* is important to understand the background of the approaches discussed. Identified similarities and differences might be grounded in the industry sector in which they are applied. Possibly, some elements can be transferred successfully from one industry sector to another. The *understanding of services* differs tremendously among the approaches. Some consider a service comprehensively, i.e. it represents a complete business process. On the other extreme authors tend to a workflow-oriented view in which a (fully automated) service represents a single task. However, all authors use a *service hierarchy*. This classification usually consists of two or three levels (Erl, 2004; Josuttis, 2008). The differentiation of basic services and composed services is a common feature although there are varieties in detail. The right choice of *granularity* within an SOA is critical and extremely difficult. In the following, granularity shall describe the functional scope of a service. Obviously, there is no silver bullet for the right granularity. Fine-grained services can easily be reused in different contexts (i.e. for many processes) but this can lead to higher complexity when orchestrating the huge number of services. Coarse-grained services are able to fulfill more complex tasks but they are less flexible and harder to reuse.

The underlying *SOA paradigm* affects the identification and specification of services. It represents the idea of what an SOA actually is. The *direction of the analysis* (i.e. bottom-up or top-down) has an important effect on the specification of services and is therefore another criterion. The authors use a range of *tools* that can be subsumed into business process modeling (BPM), process decomposition, domain decomposition, asset analysis and portfolio management (Josuttis, 2008). Depending on the focus of the respective approach *types of categorization* vary. Whereas technically-driven approaches categorize e.g. by implementation strategy, business-driven approaches might differentiate by service consumer type (i.e. internal or external).

Business Aspects

The *business aspects* are the second group of criteria. C*onsideration of strategic aspects* is very important because an SOA is not implemented for its own sake but seeks tangible benefits for the company. Although the strategic relevance might be less critical for the identification of services itself, it is crucial for their design and subsequent sourcing strategies. A categorization by Allen therefore differentiates between three types of services (Allen, 2006). Commodity services are stable, sufficiently established services that every market player must have. They are suitable for outsourcing and standardization. Territory services are fairly wide-spread but less stable and usually represent business rules. Value-added services constitute the special value of a company's product or service in the market, i.e. a company's core competence. It is this highly innovative service that gives distinction to the company (Allen, 2006).

Laws and regulations limit a service's suitability for outsourcing. Thus, services handling sensitive customer data must be checked for *legal compliance*. Furthermore, *internal policies* must be taken into account when services are identified and specified afterwards. Ever growing requirements concerning flexibility and agility for existing and heterogeneous IT landscapes necessitate an *IT governance*. Thus, frameworks such as COBIT must not be ignored. *Service level agreements* (SLA) are important for the composition of services. Whether the figures in service levels can be summed up or have to be recalculated in one or another way must be checked for every single case.

SOAs are frequently mentioned as means for standardization and flexibilization without noticing the ambivalence of these *goals*. Most approaches implicitly hint at which goal should be achieved. The *object supported* by a service may be a complete value creating process or just one single task, i.e. a step in a workflow. Consideration of the *SOA life-cycle* shall ensure the sustained maintenance of identified and implemented services as well as the intake of new services. In order to identify redundant services, existing ones must be checked for *functional similarity*.

Technical Aspects

The way services are controlled belongs to the *technical aspects* of services. Basically, there is a differentiation between *orchestration and choreography* of services (Arsanjani, Ghosh, Allam, Abdollah, Ganapathy and Holley, 2008). Orchestration implies a central instance that coordinates all activities of a process and results in a composed service. Choreography means that services are called by other services and there is no unique steering unit. The sequence of services involved in a process is not stored as metadata (Josuttis, 2008).

Customer interaction is considered in different degrees by the approaches presented. As far as services (and not tangible products) are concerned, the inclusion of the external factor “customer” is essential. The same holds true for *employee interaction* because it sets certain limits for standardization, automation and outsourcing. Several *IT criteria* are especially important for the specification of previously identified services. Thus, they are part of most of the presented approaches. The *call frequency* of a service hints at its application. On the one hand a high frequency can point to a service with a small scope of functionality that can therefore be used flexibly in many business processes. On the other hand a sufficient standardization of coarser grained services could be a reason for a high call frequency as well.

Economic Aspects

Value creation is the added value created through deployment of a service. The customer has to be willing to pay for the result of a process, i.e. services should always increase the value of a product. The degree of value creation depends on an effective and efficient combination and coordination of resources (Roth, 2007). *Maintenance and operation costs* correlate strongly with the complexity of an IT infrastructure. An SOA can lead to a significant reduction of complexity. Moreover, well-defined functions and interfaces contribute to the robustness of IT systems which in turn lowers operational costs.

An intake of new services into the IT landscape of a company causes only little *testing effort for new functionality*. Only interfaces of the newly implemented services must be tested because interactions of other services are untouched. Implementation of an SOA decreases *vendor dependency* because such an architecture is platform independent. Firstly, this leads to immediate savings because the purchase of licenses may be unnecessary when open source products can be used. Secondly, a lock-in effect is avoided so the company is not bound to a vendor because of prohibitively high swapping costs. Thirdly, web services can flexibly be used and increase the agility of business processes. These web services can be purchased ad hoc from the cheapest provider respectively.

Flexible orchestration of services enables a *demand-oriented quality of service (QoS) level* for products. Customers receive exactly the quality they request. Thus, *customer satisfaction* is increased at the same time. This kind of orchestration allows for an *individualization of products* that leads to competitive advantages and thus is another economic aspect. *Specialization on core competencies* plays an ever bigger role in today's competitive environment (Prahalad and Hamel, 1990). Identified service candidates can be classified on the basis of their strategic importance which has implications for sourcing decisions.

The *product range* can be *widened* by recombining services on the basis of existing core competencies. Originally internal services can be *offered in the marketplace* after being identified as services with such potential. The acquisition of service users generates even more economies of scale and leads to decreasing costs per unit. This, in turn, can boost the market share through decreasing prices for the service. The necessary *scalability* is another strength of SOAs. Deployment of certain services can significantly reduce the *time-to-market* of new products. This advantage can be crucial to position new products in the marketplace. Due to its agility and flexibility SOAs can react quickly to changing customer requirements. An *SOA controlling* could be implemented through a balanced scorecard. Qualitative goals have to be translated into quantitatively measurable key performance indicators (Mueller, Viering, Ahlemann, and Riempp, 2007).

Method Engineering

The approaches compared in this paper are so called methods in design science research (Hevner, March, Park and Ram, 2004). For several years there have been efforts to guide the development of such methods in order to guarantee a high quality. The task of method engineering is to give this guidance. The most popular approaches all identify activities, roles, results, techniques and the sequence of activities as important components of methods (Gutzwiller, 1994; Heym, 1993; Karlsson, 2002; Goeken, 2006). Thus, a further set of criteria looks at how far *components of method engineering* are incorporated into existing approaches of service identification.

- Activity: Unit of execution that produces a result by facilitating techniques and notations.
- Role: Definition of who carries out which activities.
- Result: Artifact that is produced through an activity.
- Technique: Instruction that describes the course of action within an activity.
- Sequence of activities: Succession of activities.

For the evaluation of these components a 5-level Likert scale that ranges from "--" (not fulfilled) via "-", "o" and "+" to "++" (completely fulfilled) is applied.

Guidelines of Design Science Research

The same scale is used to evaluate the application of Hevner et al.'s *guidelines of design science research* (Hevner et al., 2004). *Documentation, research rigor* and *evaluation* are the three guidelines discussed here. The documentation has to ensure that results are communicated both technology-oriented as well as management-oriented. Research rigor corresponds to the applied research methodologies (e.g. a sound literature study). Evaluation is ought to guarantee quality and usability of the newly created method.

STRENGTHS AND SHORTCOMINGS OF EXISTING APPROACHES

The methods found in related literature differ considerably in their methodological approach. Advantages and disadvantages as well as a possible usability for adequate and process-oriented service identification are subject to discussion in the following.

Table 2 compares five approaches and facilitates the criteria explained in chapter two. The most comprehensive *understanding of services* can be found in Böhmann & Krcmar's approach (Böhmann and Krcmar, 2005). Their services (modules) represent complete packages of service products offered to customers. Klose et al. and Arsanjani et al. look at process chunks with a smaller scope of functionality (Klose, Knackstedt and Beverungen, 2007; Arsanjani et al., 2008). Still, these chunks implement a complete and self-contained business functionality. The change from an object-oriented view to a service-oriented view that is postulated by many authors is not to be found in Winkler's approach (Winkler, 2007; Zacharias, 2005). Kohlmann & Alt's services support business processes, too (Kohlmann and Alt, 2007). However, the scope of their services differs significantly as far as functionality is concerned.

	Klose et al. (2007)	**Böhmann & Krcmar (2005)**	**Winkler (2007)**	**Arsanjani et al. (2008)**	**Kohlmann & Alt (2007)**
Basic aspects					
Industry sector	Production	IT services	Financial services	Financial services	Financial services
Understanding of services	Business process oriented	As module, very comprehensive	Object-oriented	Business process oriented	Business process oriented
Service hierarchy	2 levels: elemental and composed service	Process service	2 levels: basic and composed service	2 levels: elemental and composed service	3 levels: process, rule and entity service
Granularity	Middle	Coarse	Fine	Coarse	From coarse to fine
SOA paradigm	Architectural concept	Architectural concept	Architectural concept	Architectural concept	Architectural concept
Direction of analysis	Hybrid	Hybrid with bottom up tendency	Top down	Hybrid with focus on top down	Hybrid
"Tools"	Decomposition of business processes and SOA principles	Asset analysis	Decomposition of business processes	Goal service modeling, domain decomposition, asset analysis	BPM, asset analysis
Types of categorization	Consumer type	Consumer type, implementation strategy	Implementation strategy	Role in business model, consumer type, implementation strategy	Role in business model, implementation strategy

Business aspects					
Consideration of strategic aspects	Lines of interaction & line of visibility	Threats and opportunities of modular service architectures, external sourcing	-	Reference models and best practices from own industry, sourcing strategies	Sourcing strategies, inter-organizational cuts
Legal compliance	Legal requirements concerning customer data	-	-	-	Customer data remains in own company
Internal policies / IT governance	Only implicit	Only implicit	-	"Rules and policy analysis" within BPM	Naming of services
Service level agreements	-	Defined individually with performance indicators	-	-	-
Goal	Flexibilization	Flexibilization	Standardization	Flexibilization	Unclear
Supported object	Task	Business process	Task	Task	Business process
SOA lifecycle	-	-	-	Fractal model for service-oriented software development	-
Functional similarity	Industry standards	-	-	Self similar fractals, industry standards	Functional and semantic similarity in clustering phase
Technical aspects					
Orchestration vs. choreography	Orchestration	Orchestration	Unclear	Unclear	Orchestration
Customer interaction	Line of visibility, line of interaction	Customer specific configuration, customer integration, line of visibility	-	-	-
Employee interaction	Automatic, dialogue, manual	-	-	-	-
Criteria of information technology	Design principles of SOA	Reusability, standardization, independence	Reusability, redundancy, frequency	Reusability, flexibility	Reusability
Call frequency	-	-	Calls per time	-	-
Economic aspects					
Value creation	-	-	-	-	-
Maintenance and operation costs	-	Utilization of common resources	-	Elimination of redundancies	-
Testing effort for new functionality	-	-	-	-	-
Vendor dependency	-	-	-	-	-
Demand-oriented QoS levels	-	Within performance and design analysis	-	-	-
Customer satisfaction	-	-	-	-	-
Individualization of products and services	-	Included in goal definition	-	Inflexible architecture replaced by reusable components	-

Specialization in core competences	-	External sourcing options	-	-	Sourcing models
Increase of the product range	-	Included in goal definition	-	-	-
Internal services offered to external customers	-	-	-	-	-
Scalability	-	-	-	-	-
Time-to-Market	-	Included in goal definition	-	-	-
SOA controlling	-	-	-	-	-
Components of method engineering					
Activities	++	++	++	++	+
(SOA-)Roles	--	--	--	o	--
Results	+	++	+	++	+
Techniques	+	o	+	++	-
Sequence of activities	Sequential	Sequential	Sequential	Iterative, fractal	Sequential, iterative where applicable
Guidelines of design science research					
Documentation	++	++	+	++	o
Research rigor	++	++	o	-	+
Method evaluation	++	++	-	+	+

Table 2. Comparison of service identification methods

Granularity of services differs immensely among the compared methods. Klose et al. mainly describe composed services. Böhmann & Krcmar and Arsanjani et al. rather look at more encompassing process services. On the contrary, Winkler uses very fine grained, elemental services and thus is fairly close to an object-oriented approach. Kohlmann & Alt vary the granularity of services depending on the situation. The *SOA paradigm* of all five methods is an architectural concept. The *direction of the analysis* is usually hybrid, i.e. a top-down approach (which is the focus) is complemented by a bottom-up analysis of existing infrastructure. Only Winkler solely uses a top-down approach.

The most common *tools* that are used are BPM and domain decomposition. Different *types of categorization* are facilitated to classify services in various dimensions. Particularly Arsanjani et al. look at services from different points of view. The role within a business model distinguishes basic services from process services. Consumer type categorizes services in internally used ones and those (also) used by partners and customers. The implementation strategy marks composed services or externally sourced ones. The consumer type is a focus in Klose et al.'s and Böhmann & Krcmar's approaches because customer integration and interaction are crucial. Only Klose et al. fail to discuss implementation strategies.

A *consideration of strategic aspects* is omitted from Winkler's method. Klose et al. rarely mention these aspects. Still, their thoughts on line of visibility and line of interaction somehow hint at a link to strategic aspects. Arsanjani et al. advocate the use of reference models and best practices obtained from peer groups. They consider the sourcing potential of identified service candidates. Böhmann & Krcmar examine strategic implications of an SOA in much more detail. Threats and opportunities as well as sourcing strategies are part of their identification method. Similarly, Kohlmann & Alt discuss these strategies as well as cuts in processes in inter-organizational networks. *Internal policies* are only incorporated by Arsanjani et al. and Kohlmann & Alt. The latter for instance make the point of consistent naming of service candidates. This consistency is necessary for a high rate of reusability among services and for adequate SOA governance.

Goal of the implementation of an SOA in Winkler's method is standardization. In Kohlmann & Alt's approach there is no clear goal to be identified. The other three methods clearly aim at a flexibilization. Apart from Arsanjani et al., who present a fractal model for service-oriented software development, the *SOA lifecycle* is ignored by other authors. *Functional similarities* are not discussed by Böhmann & Krcmar and Winkler. In contrast, Kohlmann & Alt discuss not only functional but also semantic similarities. Arsanjani et al. use the self-similarity of fractals for service-oriented software development. Apart from Winkler and Arsanjani et al.'s methods that cannot be classified unambiguously, all authors imply an *orchestration* of services by a central instance. *Customer interaction* is a focus in Klose et al.'s and Böhmann & Krcmar's approach although the former originates from a production company. The huge importance of the "customer factor" in service industries is not reflected at all in the three other approaches. *Employee interaction* is only discussed by Klose et al. They differentiate between automated, semi-automated and manually conducted services. Looking at *IT criteria* the nomination of reusability stands out in all approaches. This is not surprising considering the prominence of it in recent SOA literature. Klose et al. use a comprehensive catalogue of design principles of an SOA.

The *economic aspects* of services are completely out of scope in Klose et al.'s and Winkler's approaches. With the notable exception of *specialization on core competencies* there is no discussion of economic aspects in Kohlmann & Alt's method. *Maintenance and operation costs* are addressed by Böhmann & Krcmar and Arsanjani et al. The utilization of common resources through reduction of redundancies and multiple calls by the implementation of services is brought forward in both approaches. The only authors considering a *demand-oriented QoS level* are Böhmann & Krcmar with their performance and design analysis. This is plausible because their stakeholder-based approach demands an integration of customers. *Individualization of products and services* is supported by Böhmann & Krcmar's modularization and by the usage of reusable components (Arsanjani et al.). *Specialization on core competencies* is also a postulation in Böhmann & Krcmar's method. Within their goal definition they consider an *increase of the product range* and the *time-to-market* of new products. All other economic aspects, namely *value creation, testing effort for new functionality, vendor dependency, customer satisfaction, internal services offered to external customers, scalability* and *SOA controlling* are not considered in any of the approaches.

As far as *components of method engineering* are concerned all compared approaches do fairly well regarding the described *activities*. *Results* and *techniques* are usually explained in a satisfactory way. Solely *roles* are not explained in any of the approaches. Arsanjani et al. – explicitly mentioning components of method engineering – hint at the existence of roles in their method but do without further detailing. The *sequence of activities* is usually sequential. Kohlmann & Alt allow iteration at certain points. Exceptionally, Arsanjani et al. present an iterative, fractal procedure. Based on three selected *guidelines of design science research* (Hevner et al., 2004) especially Klose et al. and Böhmann & Krcmar excel with their methods. Both approaches comply entirely with the guidelines concerning *documentation*, *research rigor* and *evaluation*. Winkler particularly misses an evaluation of her method whereas a lack of research rigor is the weakest point in Arsanjani et al.'s approach. Kohlmann & Alt show shortcomings in both documentation and research rigor but have a clear advantage in evaluation though.

LESSONS LEARNED & FURTHER RESEARCH: A METHOD FOR PROCESS-OREINTED SERVICE IDENTIFICATION

As shown in previous chapters approaches for service identification differ in many ways. However, a comparison on the basis of selected criteria also identifies commonalities both in the existence and the absence of certain aspects. A new method for process-oriented service identification (POSI) has to resolve relevant flaws. Thus, aspects that are vital for the design of POSI are to be discussed in the following.

Business processes represented in a formal or semi-formal notation such as Business Process Modeling Notation (BPMN) should be the foundation of POSI. However, no SOA project will create an IT infrastructure from scratch. Therefore, given factors, e.g. existing hardware and software, must be considered. A top-down approach based on BPM or domain decomposition has to be complemented by a bottom-up analysis to guarantee a successful technical implementation.

A new method should be configurable to be applicable in different contexts, to realize different goals and to reflect company specific characteristics, e.g. it should be possible to configure POSI with respect to the users' preferences and goals of the identification process (either standardization or flexibilization). Especially the level of composed services is important in this context. Basic services, e.g. retrieval or alteration of data, are regularly subject to standardization. In contrast, process services should be flexible in most cases (Papazoglou and Georgakopoulos, 2003). However, looking at composed services the goal may differ case by case because the complexity of such services varies depending on the situation and other characteristics.

Composed services will most likely be subject to sourcing decisions because neither whole process services (which constitute the existence of an enterprise) nor basic services (that are too small) are suitable for outsourcing. Table 2 shows that economic aspects in particular find little or none adherence in existing methods. For this reason, POSI has to combine the identification of services with the consideration of these aspects. Especially functional similarities shall serve as a basis for identifying standardization potentials. Subsequently, sourcing strategies can be evaluated. Summing up, a new method for process-oriented service identification has to focus on the following aspects:

- Service identification based on BPM complemented by a bottom-up analysis
- Discovery of functional similarities to evaluate standardization potential
- Configuration regarding standardization and flexibilization
- Consideration of economic aspects

Future research will aim at consistency and soundness of the method which we plan to realize by complying with the formal requirements of method engineering. This means that activities, techniques, results and roles have to be designed, explained and documented. Furthermore, their relationships must be described and defined.

Based on Gutzwiller (1994) and Goeken (2006), activities will be defined canonically, i.e. parts of activities or sub activities are again seen as activities. Thus, they are structured hierarchically and constitute a process model which describes the activities, their relationships as well as their sequence. This process model will also define inputs and outputs for the designated activities (results of or for other activities respectively) as well as roles that perform them. The model must also provide for XOR choices to support an inherent configurability e.g. concerning standardization and flexibilization.

Additionally, techniques, which give a detailed guidance how to perform activities in order to produce results, are part of a method. The activity “reusability analysis” for example will be supported by a technique to measure functional similarities in business processes. Thus, a certain set of techniques is used to create results by supporting the activities. Finally, a documentation of these results (intermediary or final) should be provided by the method. The former should contain all relevant information either to initiate the next activity or to complete the business process.

The striking absence of roles in all presented methods is a major flaw due to the fact, that SOA is often seen as a means to accomplish alignment of business and IT. Therefore, roles must be described which support this alignment in a structural manner by guaranteeing the existence of both, the business and the technical perspective within a development endeavor. Furthermore, process orientation can only be ensured if a process owner who knows the business context is involved in the identification and design of services. Consequently, a new method has to manage roles explicitly. The implementation of an organizational unit called “Service Design Unit” (consisting of a process owner from the business side and a service owner from the IT department) is one example for a measure suggested in POSI. In future work, the activities and results of this unit within the method will be discussed in more detail.

REFERENCES

1. Allen, P. (2006) Service Orientation: Winning Strategies and Best Practices, Cambridge University Press, Cambridge.
2. Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S. and Holley, K. (2008) SOMA: A method for developing service-oriented solutions, *IBM Systems Journal*, 47, 3, 377-396.

3. Böhmann, T. and Krcmar, H. (2005) Modularisierung: Grundlagen und Anwendung bei IT-Dienstleistungen, in Thomas Herrmann, Uwe Kleinbeck and Helmut Krcmar (Eds.) *Konzepte für das Service Engineering. Modularisierung, Prozessgestaltung und Produktivitätsmanagement*, Heidelberg.
4. Erl, T. (2004) Service-Oriented Architecture - A Field Guide to Integrating XML and Web Services, Prentice Hall, Upper Saddle River, NJ.
5. Goeken, M. (2006) Entwicklung von Data-Warehouse-Systemen, Deutscher Universitäts-Verlag, Wiesbaden.
6. Gutzwiller, T. A. (1994) Das CC-RIM-Referenzmodell für den Entwurf von betrieblichen, transak-tionsorientierten Informationssystemen, Heidelberg.
7. Hagen, C. (2003) Integrationsarchitektur der Credit Suisse, in Stephan Aier and Marten Schoenherr (Eds.) *Enterprise Application Integration - Flexibilisierung komplexer Unternehmensarchitekturen*, Berlin.
8. Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004) Design Science In Information Systems Research, *MISQ*, 28, 1, 75-105.
9. Heym, M. (1993) Methoden-Engineering - Spezifikation und Integration von Entwicklungsmethoden für Informationssysteme, Universität St. Gallen.
10. Josuttis, N. (2008) SOA in der Praxis - System-Design für verteilte Geschäftsprozesse, Heidelberg.
11. Karlsson, F. (2002) Meta-Method for Method Configuration - A Rational Unified Process Case, Linköping University.
12. Klose, K., Knackstedt, R. and Beverungen, D. (2007) Identification of Services - A Stakeholder-Based Approach to SOA Development and Its Application in the Area of Production Planning, *15th European Conference on Information Systems (ECIS 2007),* St. Gallen, 1802-1814.
13. Kohlmann, F. and Alt, R. (2007) Business-Driven Service Modeling - A Methodological Approach from the Finance Industry, in Witold Abramowicz and Leszek Maciaszek (Eds.) *Business Process & Service Computing: First International Working Conference on Business Process and Services Com-puting,* GI, Leipzig, 180-193.
14. Legner, C. and Heutschi, R. (2007) SOA Adoption in Practice - Findings From Early SOA Imple-mentations, *15th European Conference on Information Systems (ECIS 2007),* St. Gallen, 1643-1654.
15. Mueller, B., Viering, G., Ahlemann, F. and Riempp, G. (2007) Towards Understanding The Sources Of The Economic Potential Of Service-Oriented Architecture: Findings From The Automotive And Banking Industry, *15th European Conference on Information Systems (ECIS 2007),* St. Gallen, 1608-1619.
16. Papazoglou, M. P. and Georgakopoulos, D. (2003) Service-Oriented Computing, *CACM*, 46, 10, 25-28.
17. Prahalad, C. K. and Hamel, G. (1990) The Core Competence of the Corporation, *HBR*, 68, 3, 79-91.
18. Roth, R. (2007) Wertschöpfungsorientierter Serviceentwurf, in Gernot Starke and Stefan Tilkov (Eds.) *SOA - Expertenwissen*, Heidelberg.
19. Winkler, V. (2007) Identifikation und Gestaltung von Services. Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich, *Wirtschaftsinformatik*, 49, 4, 257-266.
20. Zacharias, R. (2005) Serviceorientierung: Der OO-König ist tot, es lebe der SOA-König, *OBJEKTspektrum*, 12, 2, 43-52.

2.3 Zwischenfazit

Der Vergleich existierender Serviceidentifikationsmethoden auf Basis verschiedener Kriterien zeigte, dass sich die Ansätze zum Teil sehr deutlich unterscheiden. Während einige – ausgehend von Geschäftsprozessen – Services top-down identifizieren, wählen andere eine sehr stark technisch getriebene Bottom-up-Vorgehensweise. Eine Gemeinsamkeit aller Methoden hingegen ist das beinahe vollkommene Fehlen von Rollen im Sinne des Methoden-Engineering. Keine der verglichenen Methoden geht ausführlich darauf ein, *wer* bestimmte Aktivitäten bei der Serviceidentifikation ausführen soll. Gerade im Spannungsfeld zwischen Fachbereichen und IT-Abteilungen scheint aber eine solche Zuordnung von großer Bedeutung zu sein.

Eine Meta-Methode, die die effektive Konstruktion neuer Methoden anleiten soll, muss daher insbesondere Rollen berücksichtigen. Je nach Projektsituation stehen einige Rollen (wie z.B. IT-Administrator) nicht zur Verfügung. Eine adäquate Methode für das entsprechende, konkrete Projekt muss dies berücksichtigen.

Darüber hinaus sind die existierenden Methoden nicht konfigurierbar, d.h. ihre einzelnen Bestandteile sind vorgegeben und bis auf eine einzige Ausnahme auch in ihrer Abfolge festgelegt. Weder die gesamte Methode noch einzelne in ihr enthaltene Aktivitäten werden auf die projektspezifische Ausgangslage und die Ziele der Serviceidentifikation abgestimmt. Insbesondere die Anwendung von Bottom-up- oder Top-down-Techniken sollte aber von den mit der Serviceidentifikation verfolgten Zielen abhängig gemacht werden. Erstere sind beispielsweise besser geeignet, um Legacy-Systeme zu integrieren, letztere bieten Möglichkeiten für eine strategische Positionierung des Unternehmens in einem Wertschöpfungsnetzwerk. Die Konfigurierbarkeit neuer Methoden ist demnach eine Voraussetzung, um projektspezifische Charakteristika in geeigneter Form zu berücksichtigen. Die Analyse dieser Charakteristika und die Verwendung von konfigurierbaren Methodenfragmenten wird daher im weiteren Verlauf eine bedeutende Rolle spielen.

3. Gebiet II: Methodologischer Hintergrund

3.1 Einleitung

In den folgenden beiden Artikeln werden die methodologischen Grundlagen der neuen Meta-Methode beschrieben. Dabei spielen das Methoden-Engineering und das daraus hervorgegangene Situational Method Engineering eine entscheidende Rolle.

Artikel 2 befasst sich zunächst mit dem Problem, dass zwar die Konfiguration und Anwendung von Methoden und ihren Fragmenten in der Literatur relativ ausführlich untersucht wurde, die Fragmente selbst jedoch ohne jede empirische Grundlage scheinbar willkürlich kreiert werden. Um hier Abhilfe zu schaffen, kombinieren die Autoren Bestandteile des Methoden-Engineering und der Grounded Theory.

Im dritten Artikel wird gezeigt, wie Konzepte des SME auf den Bereich der Serviceidentifikation übertragen werden können. Dazu werden Kontextfaktoren und Implementierungsziele diskutiert. Die Kombination dieser konstituierenden Elemente ergibt Situationen, denen später entsprechende Methodenfragmente zugeordnet werden können.

3.2 Artikel 2: Using Grounded Theory for Method Engineering

Veröffentlicht: Looso, S., Börner, R., Goeken, M. (2011): Using Grounded Theory for Method Engineering, in: Proceedings of the 5th IEEE International Conference on Research Challenges in Information Science 2011 (RCIS 2011), 19.-21.05.2011, Guadeloupe, Frankreich, S. 98-106.

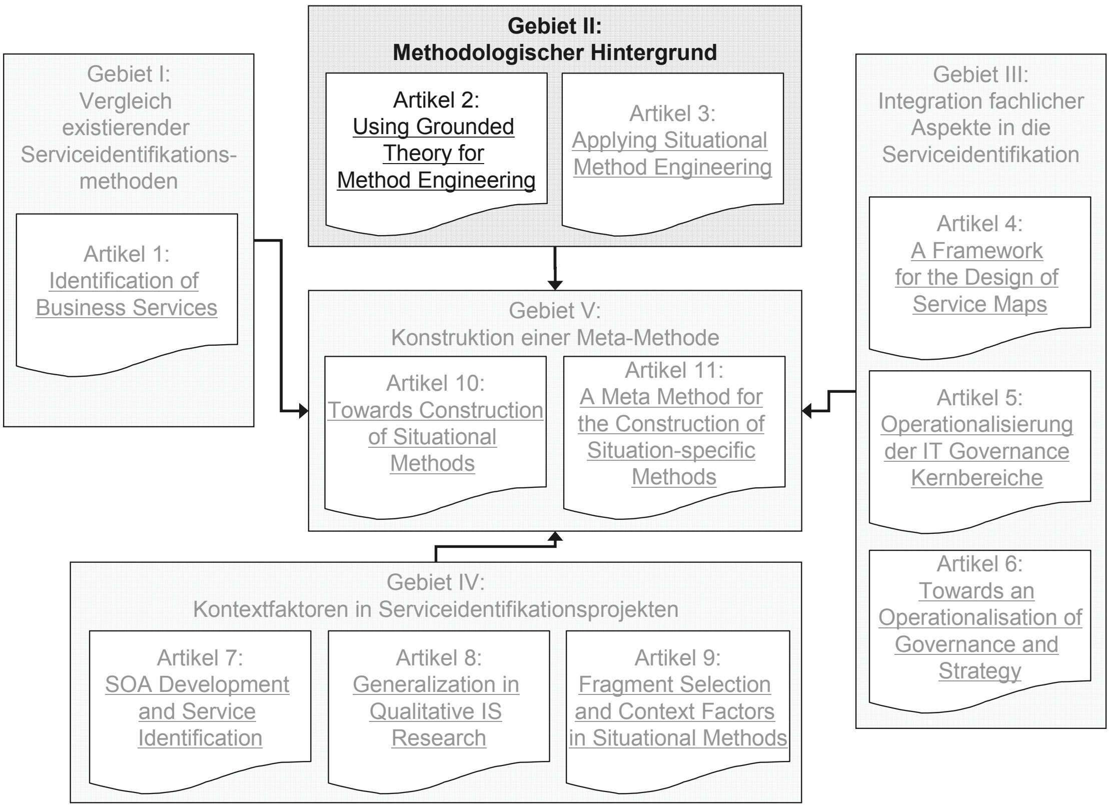

Abbildung 7: Artikel 2 im Kontext der kumulativen Dissertation

Using Grounded Theory for Method Engineering

Stefanie Looso, René Börner, Matthias Goeken
Frankfurt School of Finance & Management
Frankfurt/Main, Germany
{s.looso;r.boerner;m.goeken}@fs.de

Abstract

Grounded theory is a qualitative research approach that can be used to build theory from empirical data in an explorative fashion. Although the field of method engineering is a relatively mature research area, a lack of empirical grounding can be attested since method engineering is commonly based on literature reviews and follows a deductive, construction-oriented engineering process. Particularly, the constituent elements of methods (such as activities and techniques) are seldom grounded in empirical data. In order to enrich method engineering research and to base methods and their elements on empirical data, this paper combines features of method engineering and grounded theory. The result is a methodological support for the method engineering process that follows techniques acquired from grounded theory. This new approach to method engineering shall be called grounded method engineering. It is meant to improve a method's quality before an application or configuration takes place by anchoring its constituent elements in empirical data. Hence, grounded method engineering will underpin methods to be developed by extracting domain knowledge using techniques found in grounded theory.

Keywords

Grounded Theory, Method Engineering, Empirical Research, Method Elements, Design Science Research

I. INTRODUCTION

As for any applied science, the debate between scientific rigour and practical relevance also characterises information systems research (ISR). Especially research areas with strong relationships to the day to day practice of IS professionals are often confronted with the rigour vs. relevance discussion [1; 2]. Methods are dedicated to solving existing problems. Therefore, method engineering focuses the construction of relevant artefacts that can be used in practice, but researchers have started to discuss how the method engineering process can be made more rigorous [3]. But so far method engineering research largely follows a research process which focuses on relevance instead of rigour and the transfer of domain knowledge follows an unstructured and intuitive process [4; 5].

Goal of the approach presented herein is to enrich method engineering by grounding resulting methods in empirical data. Therefore, we support method engineering with grounded theory [6], a research method designed as a behaviouristic method which gains sociological theory from empirical data. Although [6] clearly argue that only sociologists can use grounded theory to generate sociological theory, the concept of grounded theory has been applied to several other research areas in recent years. Reference [7] already notices that grounded theory could be used in various disciplines. They dedicated their book to researchers that are interested in inductively building theory through qualitative analysis of data.

We argue that parts of the grounded theory methodology can improve the construction of methods and thus add rigour to method engineering approaches. Therefore, we present grounded method engineering (GME) which aims to produce relevant methods following a more rigorous process by using empirically grounded method elements. The basic idea is to enrich the usual construction-oriented and literature-based derivation of method elements with grounded information derived from empirical data.

The paper is organised as follows: Section 2 presents our research process, while section 3 discusses method engineering and the concept of grounded theory. The fourth section theoretically presents grounded method engineering and exemplarily demonstrates the application in the field of IT governance. Finally, in section 5 we will discuss our findings, show limitations of the approach and potential for further research.

II. RESEARCH DESIGN

On our way to designing grounded method engineering, we follow the research design proposed by [8] that is depicted in figure 1. The intriguing question for our research is what method engineering can learn from grounded theory and its application in IS research. Therefore, our research process commences with an analysis of literature from the fields of method engineering and grounded theory in order to identify challenges of method engineering and motivate our research. These analyses yielded two results: First, the engineering of methods and their constituting elements is rarely based on empirical evidence but on literature or intuition instead. Second, procedures from grounded theory such as open coding and axial coding can be used to complement method engineering approaches. The definition of a solution’s objectives is important to evaluate a newly created artefact. Along with the identification of the problem and the motivation, this definition of objectives is part of section 3 of this paper.

In section 4, design and development of our artefact, i.e. GME, is the next step. We therefore show how procedures from grounded theory can be used for an empirical grounding of method engineering. A central aspect is the transformation of the coding paradigm used for axial coding in grounded theory to adapt it to method engineering. The second part of section 4 demonstrates how grounded theory procedures can be applied in a method engineering context. For this purpose, an example from the field of IT governance is described.

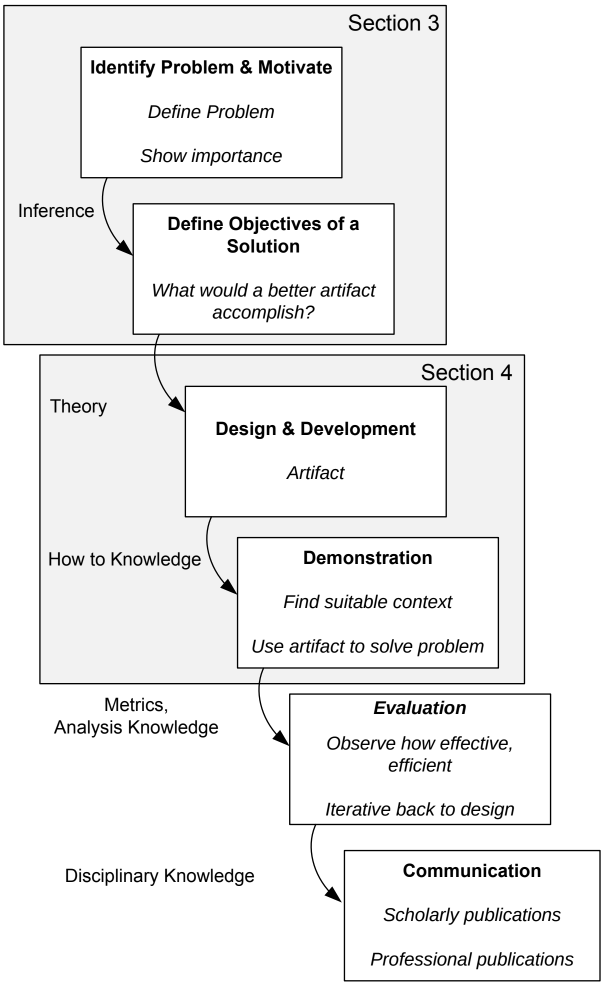

Figure 1. Research Process (following [8], p. 54)

The evaluation that is demanded not only by [8] but also by [9] is out of the scope of this paper. Observations concerning the effectiveness and efficiency of the created artefact can be fed back into the design process and are left to further research.

III. METHOD ENGINEERING AND GROUNDED THEORY

Since the early 1990s there have been efforts to guide the development of methods for information systems development in order to guarantee a high quality of artefacts generated by them. The task of method engineering is to give this guidance [10]. Several approaches to method engineering [11; 10; 12] identify activities, roles, results and techniques as important elements of methods.

Figure 2 illustrates our understanding of a method and its engineering process. Method engineering activities that are conducted by ME roles and supported by ME techniques lead to ME results, namely methods. The dashed lines show that all parts of a method (elements and relationships) are a result of the method engineering process. One example for an ME result could be the activity "analyze the company's current situation" that is supported by the technique "SWOT analysis". The derivation of elements as well as their configuration rules so far mostly follow a literature-based and construction-oriented engineering process [13].

Considering the engineering process of modelling methods, Siau argues that “despite the pivotal role of modelling methods in successful information systems development, most modelling methods are designed based on common sense and intuition of the method designers with little or no theoretical foundation or empirical evidence“ [4, p.193]. As far as theoretical foundations are concerned, we disagree with [4] view because most of the method engineering literature uses theories to ground the development of methods. Reference [14], for instance, combines activity theory and method engineering as theoretical grounding of the method component concept, whereas [15] develop a design theory nexus that connects numerous design theories.

Reference [4]’s second point, empirical evidence in method engineering, has to be investigated in more detail since it can be used for two purposes. It “includes both empirically based generation of the method, as well as its justification by using it in real life projects” [16, p.620]. The latter is often included in method engineering research processes by case studies or action research that evaluate previously engineered methods ([17; 18]). In contrast, we are focusing the former and argue that methods can gain empirical evidence ex ante. Therefore, we need a research procedure which first and foremost supports the empirical grounding of method elements during their construction process.

The quality of empirical data used to ground method elements is crucial for a method’s quality. For this reason, relying on one source only is not enough. As a second requirement, construction of method elements should be based on multiple sources in order to avoid any biases that might be introduced by a single source.

Reference [19] differentiates mode 1 knowledge as “generated in a context of established institutions and disciplines” and mode 2 knowledge as “generated in a context of application” (p. 91). So far, method engineering is partly based on mode 1 knowledge. We assume that including practical experiences by analysing interviews is strongly relevant for method engineering research. Thus, we argue that a method for practical application should thirdly consider the course of action and the experience of its dedicated context. Hence, an empirical grounded construction of method elements should be context-oriented.

Summarising the previously explained requirements, we were looking for research methods that enable an empirical grounded, multiple sources-based and context-oriented derivation of method elements. In the following, the appropriateness of grounded theory [6] will be elaborated in order to show how it fulfils the requirements stated above.

The concept of grounded theory appeared as „a counter to the then prevailing practice in social science research of focusing on the testing and verification of existing theories rather than the generation of new ones“ [20, p.5]. According to [21], grounded theory investigates empirically “how practice behaves instead of (…) if practice behaves in a specific way” (p. 1). Theories built with grounded theory are empirically grounded and using it in the field of method engineering will enrich method construction by increasing empirical evidence of methods elements ex ante.

A general strength of grounded theory is that it includes multiple sources of evidence. This is an enrichment for method engineering because it is often based mainly on literature. Grounded theory approaches accept a broad variety of data sources including annotation, internal reports, requirements engineering specifications, various kinds of manuals, protocols, videos etc. [21; 22]. For method engineering researchers it could be helpful to use several sources of the dedicated context, i.e. process descriptions [23; 24] or observations achieved through action research techniques etc. [25].

Grounded theory develops context-based theories instead of universally valid theories [6] and enables method engineering researchers to investigate the subject of interest in its context [26]. Reference [27] states that grounded theorising is well suited for the interpretation of experiences of managers and developing theoretical propositions from them. Therefore, she chooses grounded theory procedures to analyse case study interview transcripts.

Although method engineering is not creating descriptive theories of human behaviour but normative methods to guide it instead, we argue that the concept of grounded theory is well adaptable and useful to method engineering. We assume that descriptive research findings resulting from grounded theory can be used to support the construction of normative methods in addition to desk and literature research. Especially if a certain company is very successful, the elicitation of their as-is-procedure (descriptive) is a solid base for the construction of the target state of other companies (normative). The construction of these “best practice methods” should be based on experiences and knowledge of successful companies.

By using grounded theory, our approach called grounded method engineering (GME) supports the ME activity “identifying elements” by complementing other ME techniques such as literature reviews. Grounded theory procedures will be adapted to method engineering and subsequently support “ME activities” as depicted in figure 2. With GME, we propose a symbiosis of method engineering and grounded theory that enhances the empirical grounding of method elements and guides the process of engineering a method. GME shall be established as an addition to conventional method engineering techniques. Literature reviews and desk research are important nonetheless.

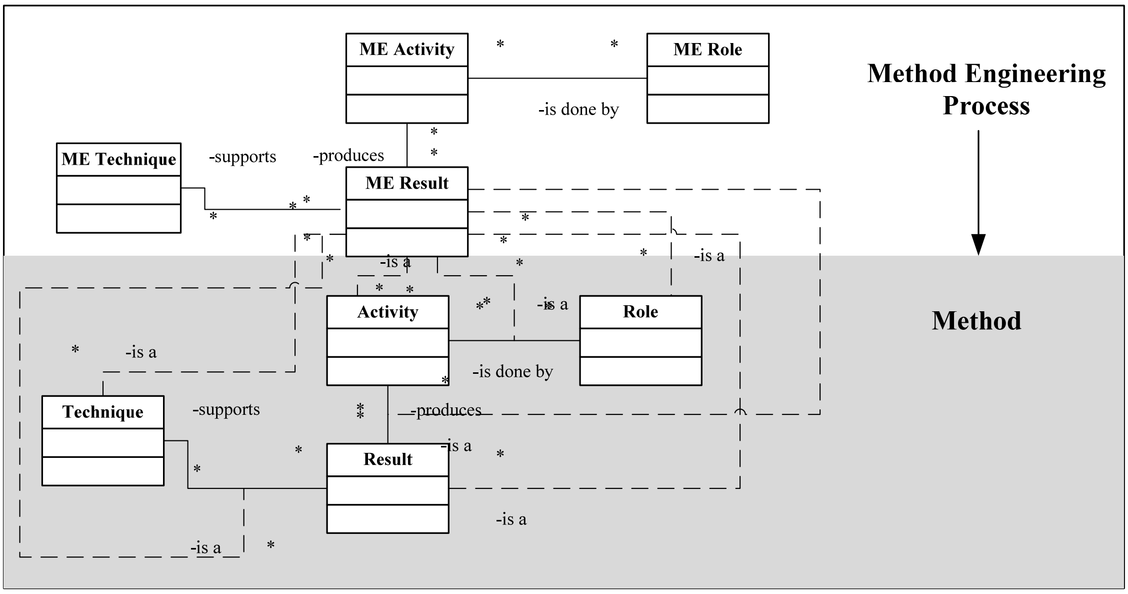

Figure 2. Method Engineering Process and Method

IV. HOW GROUNDED THEORY CAN SUPPORT METHOD ENGINEERING

A. Design and Development of Grounded Method Engineering

In the following, we investigate concrete grounded theory techniques in order to utilise them for method engineering. According to figure 1, this section describes design and development of grounded method engineering.

Grounded theory techniques are mainly based on the research of [7]. They extend grounded theory to a more formal research method including rules, typical activities, techniques, etc. Earlier research by [6] describes grounded theory as an unsystematic creative approach. Their research procedure requires a steady iteration between concept and data. The different steps of research are mainly supported by three techniques. Table I sketches these techniques of grounded theory by [7]. In the following we will discuss how these techniques can be applied to the field of method engineering.

Grounded Theory Technique	**Description**
Open Coding	Qualitative data is collected, examined, and put into concepts. This iterative process is done by constantly referring to the substantive area to ‚compare notes' and further refine or categorize the data.
Axial Coding	The categories which emerge from open coding are systematically related in a causal model to describe the dynamic relationship between them using a coding paradigm referred to as ‚Paradigmatic Model'. This PM makes use of the conditions, contexts, actions, strategy and consequences to relate identified categories to each other.
Selective Coding	A core category is selected that is deemed central. All other categories are integrated and systematised around the former. This is the stage where a grounded theory starts to emerge as a set of categories that are related to each other, thus forming theory.

TABLE I. TYPICAL GT TECHNIQUES (SEE [20] AND [7])

(1) The first step of grounded theory is to open the data for a free coding [21] and discover concepts from the transcripts. The corresponding technique named open coding means breaking down, examining, comparing, conceptualising, and categorising data. Concepts represent the basic analytic blocks. They label sections of text (also called "units of code") that describe or identify certain important aspects of the phenomena under consideration [7]. Particularly, questions like how, what, and who are used to break down the text into smaller units. Because grounded theory does not work with a predefined code system, first concepts are inspired by the interview texts. In the open coding process, we continuously compare the original text and concepts to improve the latter. Finally, categories that subsume similar concepts emerge, i.e. that, for instance, synonyms are harmonised and similar concepts are aggregated to categories. Thus, a certain degree of abstraction is reached during the open coding already.

In order to identify elements of a method, method engineering researchers need to discover concepts concerning their planned method. Concrete examples will be presented in the following section. The grounded theory technique open coding can be used for grounded method engineering without major adjustments.

(2) The second technique is axial coding which formalises a set of procedures that allows the reformulation of the data obtained from open coding. Goal of this step is to look for different meanings in categories and identify connections. Reference [7] states that categories should be sorted according to a certain inherent hierarchy. They recommend axial categories according to the coding paradigm depicted in figure 3 for the classification of categories. Therefore, the identified categories are grouped to these axial categories. Categories either describe a phenomenon, its context, its consequences, its reasons or the strategies to cope with it.

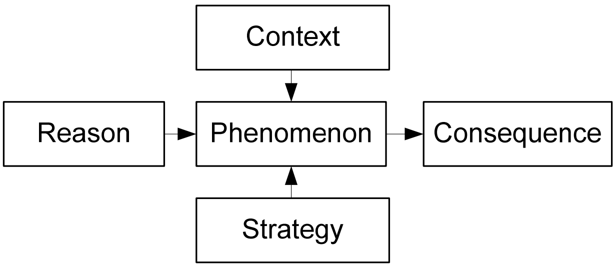

Figure 3. Coding Paradigm by Strauss and Corbin

Turning our back at sociological theories and looking at method engineering, the coding paradigm has to be adapted to the particularities of method engineering. In order to apply grounded theory to the latter, we need an equivalent to the above paradigm in the field of method engineering. A sociological theory is represented by several phenomena, their reasons, specific context etc. having a variety of different relationships. Since the result of GME is not a theory but a method, this coding paradigm has to be adapted accordingly. A method's elements (as shown in the lower part of figure 2) thus can be used as axial categories for method engineering. The phenomena, reasons, context, consequences and strategies are replaced by activities, roles, techniques and results. Hence, figure 4 illustrates the coding paradigm of grounded method engineering.

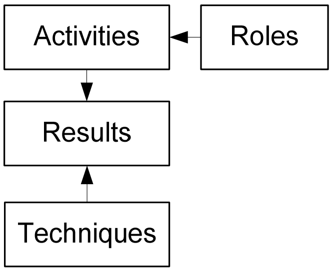

Figure 4. Coding Paradigm of Grounded Method Engineering

This a priori definition of certain method elements as a coding paradigm may be criticised because grounded theory is generally understood to be an open research method. Thus, open coding should not be restrained by categories that were defined in advance. However, axial coding aims to structure the categories retrieved from open coding.

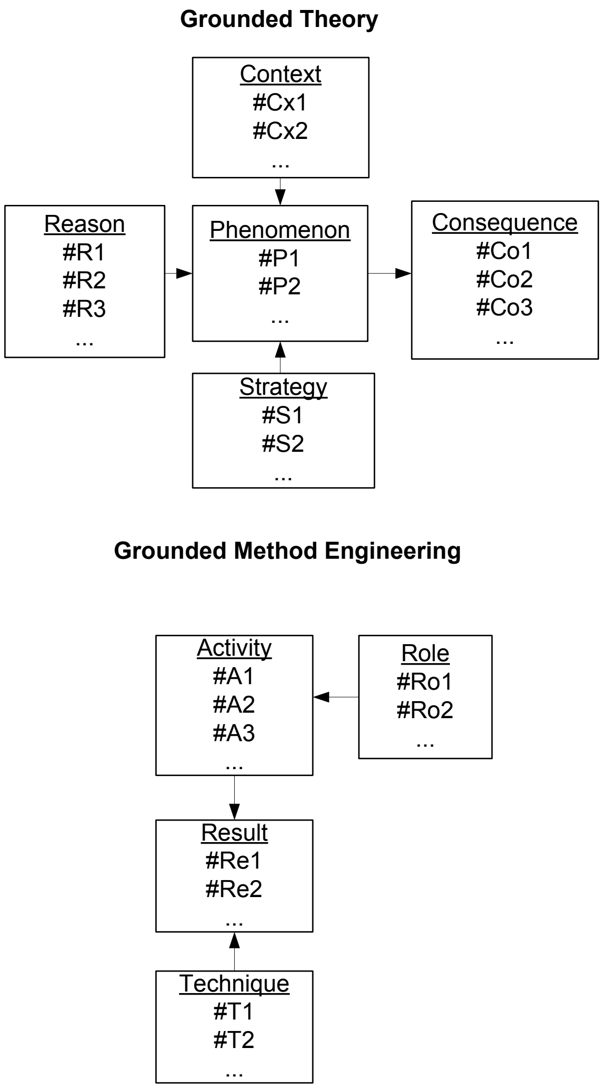

Figure 5. Axial Coding in GT and GME

Figure 5 shows a schematic result of axial coding. Generally, the categories identified during the open coding process are assigned to the axial categories shown in figure 3. In the case of GME, categories are classified as activities, results, techniques and roles (axial categories of GME). It is important to acknowledge that this is only one plausible coding paradigm for method engineering. Depending on the underlying method engineering approach, further axial categories, i.e. elements, could be added to this paradigm. Reference [12], for instance, includes “reason”; [10] uses “principle” in his approach. Hence, the four axial categories depicted in figure 4 would have to be complemented by one or more additional axial categories. Before using GME, the underlying method engineering approach has to be outlined to adjust the coding paradigm, respectively. The elements shown in figure 4 are the coding paradigm of grounded method engineering based on the method engineering approach by [28].

(3) The third technique is selective coding. Selective coding models relationships among categories that were classified in the axial coding phase, i.e., it relates categories to other categories. Furthermore, these relationships are validated, e.g. by comparison

with existing literature. The relationships between elements of the coding paradigm, i.e., axial categories, can support the description of concrete relationships between identified categories.

In case of grounded method engineering, the identified and classified categories (concrete method elements) have to be arranged and related to each other. A concrete activity, for instance, is assigned to one or more actual results. The same way, one or more techniques that can be used to generate results are determined and related to the latter. Roles are linked to activities, respectively. As shown above, axial coding was used to group categories to axial categories. Now, insights from the open coding phase are used to generate relationships. In the selective coding phase, usually a core category that is deemed central is selected. All other categories are integrated and systematised around this central category. We assume that for method engineering all categories are of equal importance. Hence, we do not specify a core category.

Instead, our approach includes a core axial category. We argue that methods are generally process-oriented and the sequence of activities plays a significant role. Therefore, the axial category "activity" will be deemed a core axial category. "Activity" should be used as core axial category because all method elements are bound to activities in one way or another. Subsequently, all identified concrete activities have to be arranged in a sequence using, for instance, an event-driven chain. Only after this is done, other categories (such as concrete roles and techniques) are assigned and related to these activities. A schematic result of the selective coding phase is illustrated in figure 6. In contrast to general GT approaches, the categories #A1, #A2 and #A3 are arranged in a sequence in the case of GME.

All activities identified during the axial coding phase are considered to be central and all other categories should be arranged around them. Modelling activities in a certain sequence is also useful for the illustration of methods as an event-driven chain. The relationships between activities are not the only ones though. Relationships between other (axial) categories have to be considered as well. Roles, techniques and results are commonly linked in more or less complex ways to one another.

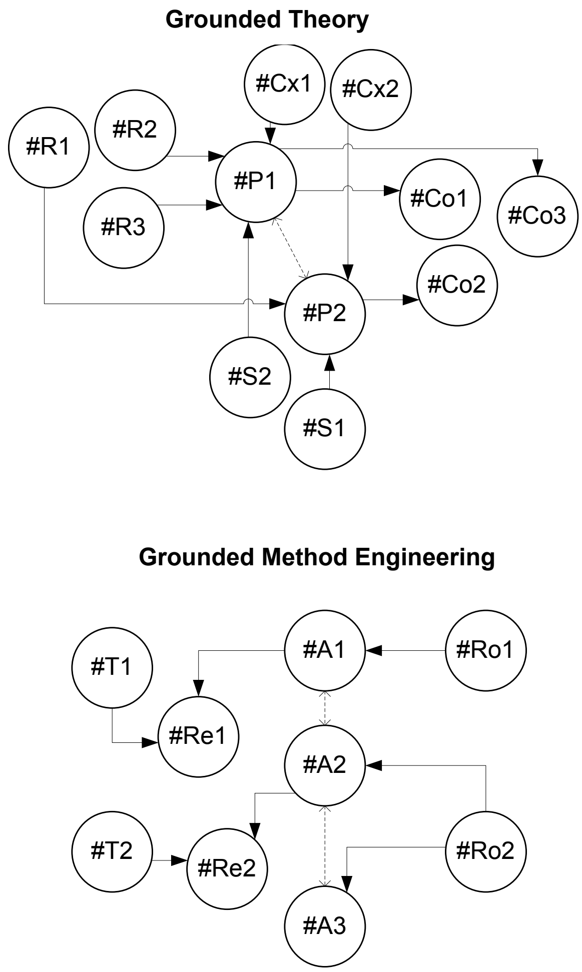

Figure 6. Selective Coding in GT and GME

B. Demonstration and Exemplary Application in the Field of IT Governance

The exemplary application – or so-called demonstration in figure 1 – presented in the following is taken from the field of IT governance. The importance of this topic has increased during the last decade since IT infrastructures are becoming ever more complex. Best practice reference models (BPRM) promise support for manifold challenges IT departments are confronted with [29; 30]. Therefore, the interest in these best practice reference models is growing, and more and more companies apply BPRM to support their IT governance. However, there is limited knowledge about how BPRM are applied in practice and there is no structured method to support the application and lift the full potential of BPRM. The challenge of creating such a method by using GME is described in this section.

Following our GME approach, we base our method engineering process on interviews out of the dedicated context. These interviews are taken from a qualitative study on the "use of multiple reference models of IT governance" which focuses on how enterprises deal with a variety of BPRM (such as COBIT or ITIL) applied simultaneously in an organisation. The interviewees are skilled experts whose long-time experience accounts for their knowledge on the research topic. Twelve interviews lasting several hours have been carried out. The interview transcripts are the starting point for GME. Table II shows examples of interview excerpts referencing the research topic.

Interview excerpts
„|The management| is a frequent trigger| for the use of best practice models|. Being an |internal client|, it puts pressure on the |IT department."
"The first step of our consulting process is |position fixing|. The question is which problem the client has and which projects fail more often than others. Then we look in the frameworks and standards and look for gaps and solutions."
"By using a |structured SWOT assessment| we analyse the strengths and weaknesses of the clients' processes|. An outcome of this step is a catalogue of recommenda-tions|."
"Management look for possibilities to improve their |IT departments| During this search they often find several frameworks and standards. And they want only the best parts of all them."
"A team discusses what is the best for our purpose,| they derive requirements| and collect policies and so on. |Outcome of this process is a big excel sheet.|"
"If there is a problem with a process| the models are used to improve it.| The models are a supporting tool for Process Improvement|."
"Process owners use the frameworks as a blueprint for our |IT processes."
…

TABLE II. INTERVIEW EXCERPTS

Step 1: Discovering Concepts with Open Coding

By using the grounded theory technique open coding, we analyse all interview parts that deal with the application of BPRM. The bars in table II separate the text into different units of code that have to be conceptualised.

Coding without limitations is an important principle of open coding. This could be a problem if the investigator already has method elements in mind while coding the concepts. Such a preoccupation can be avoided by leaving the coding to researchers that are not involved in the actual method engineering process. Moreover, a stringent use of questions like "how", "what" and "who" allows for an unrestricted open coding because the focus is not yet on method elements.

Concepts that emerged from the units of code taken from the interview transcripts can be categorised, e.g. the concepts "position fixing" and "structured assessment of strengths and weaknesses" can be aggregated to the category "analyse current situation". Accordingly, all other concepts are categorised. While analysing the interview excerpts in this open coding phase, we came to 14 categories. Two of these categories and respective concepts are exemplarily illustrated in table III.

Category	Concepts
Analyse current situation	Position Fixing; Structured Assessment of Strengths and Weaknesses
Design Process	Process Improvement, Process Redesign, Process Design
…	…

TABLE III. RESULT OF OPEN CODING

Step 2: Structuring Categories along Method Elements with Axial Coding

This technique aims to structure the extracted categories with regard to the coding paradigm illustrated in figure 4 including the method elements activity, result, role, and technique.

In order to classify categories according to the coding paradigm illustrated in figure 3, [7] recommend a detailed semantic analysis. This includes, for instance, an a priori definition of signal words that hint at *reasons* (like “because”, “therefore”, etc.). If one of these words appears in the text, the unit of code is likely to be a concept belonging to the axial category “reason”. In our case, whenever a verb occurred in the text, we investigated if there was an *activity* related to it. Hence, the category “analyse current situation”, for instance, emerged from units of code such as “we analyse the strengths and weaknesses of the clients’ process” (see table II and III). Phrases like “by using” or “applying” strongly hint at *techniques*. Consequently, the interview quote “by using a structured SWOT analysis” that can be found in table II led to the category “SWOT analysis” that is associated with the axial category *technique*. Table IV shows a structured list of axial categories and respective categories as a result of axial coding.

Method Element (Axial Category)	Category
Activity	Analyse Current Situation, Select Subset, Design Process
Result	BPRM, IT Process Model; BPRM Subset
Role	Management; Process Owner; IT Department, Consultant
Technique	Spreadsheets, GAP Analysis, SWOT Analysis, Group Discussion

TABLE IV. CATEGORIES SORTED BY METHOD ELEMENTS USED AS AXIAL CATEGORIES

Step 3: Relating Method Elements by Selective Coding

In the selective coding phase, the identified categories (concrete method elements such as SWOT analysis) are related to one another in order to support method engineering. Therefore, transcripts of interviews are analysed once again to identify relations between categories. The relations between axial categories, i.e. between the method elements, used in GME (figure 4) can support the arrangement of categories conducted in the selective coding phase. Concrete results are assigned to activities, techniques to actual results, etc.

In our case depicted in figure 7, the management triggers the activity “analyse current situation”. Results such as “IT process model” and “BPRM” are necessary inputs for this activity. Utilized techniques are a GAP analysis, a SWOT analysis as well as various spreadsheets. The selection of BPRM subsets is supported by a group discussion. For the design of processes by the process owner no technique could be identified in the interview transcripts.

The next step is to find a reasonable sequence for the identified activities. Since the interview transcripts comprise only little evidence as far as a sequence is concerned, arranging the activities in a sensible way poses a major problem. Some parts of the text indicate that “analyse current situation” should be the initial activity which seems to be obvious at first sight. However, some interviews suggest that this activity could be carried out (again) after “design process”. This clearly shows that – due to insufficient or contradictory evidence gained from empirical data – some relationships (like the sequence of activities) have to be modelled with a degree of uncertainty. It is thus important to keep these flaws in mind and collect additional data to underpin these relationships. Although the arrangement is preliminary, other relationships (e.g. between roles and techniques) can be modelled independently so that a later re-arrangement of activities will not affect other parts of the model.

Following steps one to three, a method as partly depicted in figure 7 can be configured. After conducting these steps once, the resulting method is likely to be preliminary. Grounded method engineering is an iterative approach and advises to move back and forth between collecting and analysing data. This improves methods in two ways. On the one hand, concepts, categories and their relationships can be confirmed by collecting more empirical evidence. On the other hand, new concepts and categories might occur and enable a refinement. Moreover, assignments to axial categories might be redefined or relationships might change.

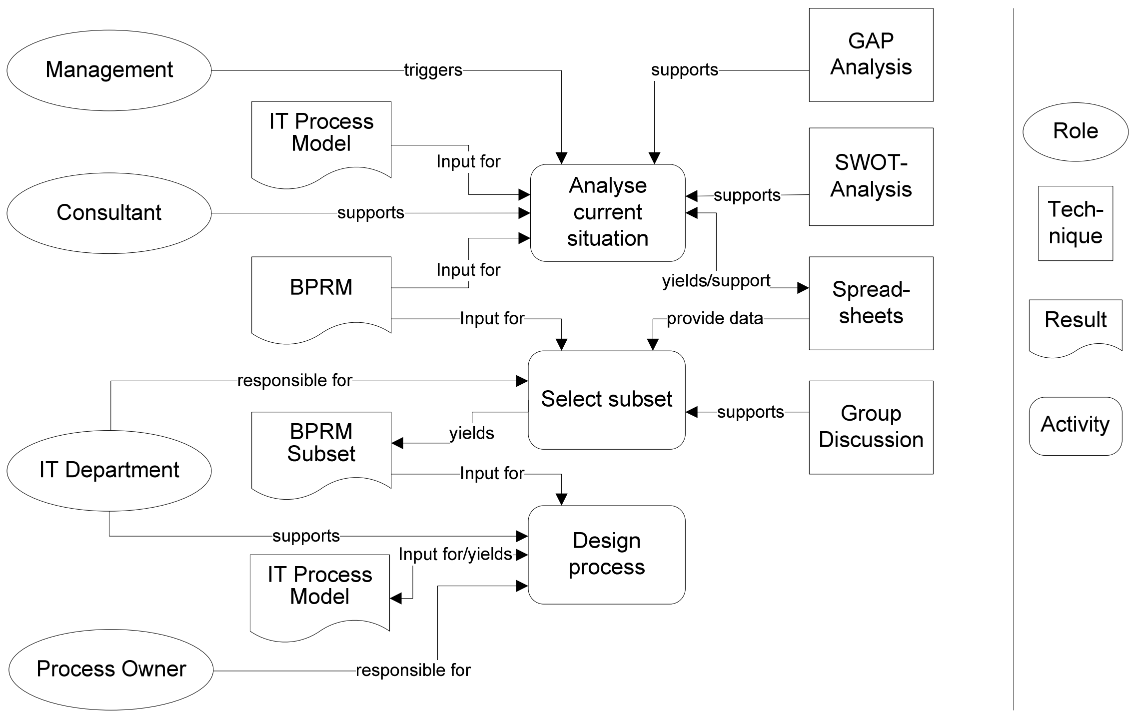

Figure 7. Result of Selective Coding

It is important to adhere to the principle of openness at that point and not to stick to once made definitions. In our concrete example, we observed the following: After an initial round of interviews, we assumed that techniques support results (see figure 4) which is quite plausible considering related literature on method engineering. In later interviews, an increasing number of interviewees talked about using techniques for certain activities. Therefore, relationships of (axial) categories might have to be changed if further data confirms this. Furthermore, new categories (such as the technique “workshop”) that have not been mentioned earlier might appear and complement the pool of identified categories. Thus, every additional iteration contributes to the completeness and reliability of methods and further research will certainly improve the method exemplarily presented herein. Particularly, relationships will be uncovered enabling the method engineer to use illustrations such as event-driven chains.

V. CONCLUSION, LIMITATIONS AND FUTURE RESEARCH

In order to proof rigour in design science in general and in method engineering in particular, many scientists demand an ex post evaluation of research results. That means that methods are tested after their development. Our approach to add rigour to method engineering already begins in the construction process of methods by grounding method elements in empirical data. As shown in this paper, GME enriches current research practices in the field of method engineering. We argue that grounding method elements in empirical data is a useful addition to literature-based research and construction-oriented method engineering. Hence, it should complement research methods like literature reviews and case studies. GME is an opportunity to transfer domain knowledge to methods and bridge the gap between methods and their dedicated context.

Furthermore, our research provides concrete steps that can be performed in a method engineering process. These steps have been inspired by grounded theory and are adapted for method engineering purposes. By following these steps, constituting parts of methods, i.e. activities, roles, techniques and results can be extracted from multiple sources of evidence such as interviews and project documentation. The methods (and their elements) created by the subsequent engineering process are properly grounded in domain knowledge. Thus, they are able to specifically support method engineering in the respective context.

A limitation to our results is that grounded theory is usually used to reach “insights in human behaviour”. GME aims to guide human and organisational behaviour instead. However, we argue that these insights gained by grounded theory techniques can be used to extract domain knowledge and thus support method engineering. The example showed that after acquiring knowledge about the application of BPRM from experts, this knowledge could be used to engineer a method guiding the application of BPRM in other companies.

Another limitation is that our results are solely based on interviews. The quality thus depends heavily on the expertise of the interviewees. Therefore, additional sources like internal documents (e.g. company policies) should be analysed as well.

REFERENCES

[1] C.H. Lee, J.H. Lee, J.S. Park, K.Y. Jeong, "A study of the causal relationship between IT governance inhibitors and its success in Korea enterprises" HICSS 2008 - Proceedings of the 41st Annual Hawaii International Conference on System Sciences. Washington, DC, USA: IEEE Computer Society.

[2] M. Rosemann, I. Vessey "Toward Improving the Relevance of IS Research to Practice: The Role of Applicability Checks" MIS Quarterly 32(1), 2008, pp.1-22.

[3] S. Hoppenbrouwers, H. Proper, T. van der Weide "A Fundamental View on the Process of Conceptual Modeling" Proceedings of the Conceptual Modeling - ER 2005. 24th International Conference on Conceptual Modeling, Klagenfurt, Austria, October 24-28. Berlin Heidelberg: Springer-Verlag GmbH, 2005, pp.128-143.

[4] K. Siau, "Information Modeling and Method Engineering" Valenti S, (eds). Successful Software Reengineering: IRM Press, 2002, pp.193-209.

[5] M. Markus, "The Qualitative Difference in Information Systems Research and Practice". Lee AS, Liebenau J, Decross JI, (eds). Proceedings of the IFIP TC8 WG 8.2. London: Chapman & Hall, 1997, pp.11-27.

[6] A. Glaser, A. Strauss "The Discovery of Grounded Theory: Aldine de Gruyter, New York, USA, 1967.

[7] A. Strauss, J. Corbin "Basics of qualitative research: Grounded theory procedures and techniques", Sage Publications Inc., Thousand Oaks, CA, US 1990.

[8] K. Peffers, M. Rothenberger, S. Chatterjee "A Design Science Research Methodology for Information Systems Research" Journal of Management Information Systems 24(3), 2007/2008, pp.45-77.

[9] A. Hevner, S. March, J. Park, R. Sudha "Design science in information systems research," MIS Quarterly 28(1), 2004, pp.75-106.

[10] S. Brinkkemper "Method engineering: engineering of information systems development methods and tools," Information & Software Technology 38(4), 1996, pp.275-80.

[11] T. Gutzwiller "Das CC-RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen," Physica, Heidelberg, 1994.

[12] F. Karlsson "Meta-Method for Method Configuration - A Rational Unified Process Case" [Dissertation]: Linköping University, Faculty of Arts and Sciences, Linköping 2002.

[13] R. Hurrion "A Sequential Method for the Development of Visual Interactive Meta-Simulation Models Using Neural Networks," The Journal of the Operational Research Society 51(6), 2000, pp.712-719.

[14] F. Karlsson, K. Wistrand "Combining method engineering with activity theory: theoretical grounding of the method component concept.," European Journal of Information Systems 15(1), 2006, pp.82—90.

[15] J. Pries-Heje, R. Baskerville "The Design Theory Nexus," MIS Quarterly 32(4), 2008, pp.731-55.

[16] F. Karlsson, P. Agerfalk "Method Configuration - Adapting to Situational Characteristics while Creating Reusable Assets,". Information & Software Technology 46(9), 2004, pp.619-33.

[17] M. Rossi, B. Ramesh, K. Lyytinen, J.-P. Tolvanen "Managing Evolutionary Method Engineering by Method Rationale," Journal of the Association for Information Systems 5(9), 2004, pp.356-391.

[18] M. Leppänen "An Ontological Framework and a Methodical Skeleton for Method Engineering: A Contextual Approach," Jyväskylä 2005.

[19] D. Partington "Building Grounded Theories of Management Action,". British Journal of Management 11, 2000, pp.91-102..

[20] G. Galal "From contexts to constructs: the use of grounded theory in operationalising contingent process models,” European Journal of Information Systems 10(1), 2001, pp.2-14.

[21] B. Hansen Hansen, K. Kautz "Grounded Theory Applied - Studying Information Systems Development Methodologies in Practice,”: Proceedings HICSS 2005., pp.264f.

[22] B. Lings, B. Lundell "On the adaptationof Grounded Theory procedures: insights from the evolution of the 2G method,” Information Technology and People 18(3), 2005, pp.196—211.

[23] J.C. Hood, “The Sage Handbook of Grounded Theory,” Thousand Oaks, CA, US: Sage 2007.

[24] K. Charmaz "Constructing grounded theory: A practical guide through qualitative analysis,” Sage, London 2006.

[25] D. Avison, F. Lau, M. Myers, P. Nielsen "Action Research”. Communications of the ACM 42(1), 1999, pp.94-107.

[26] J. Hughes, S. Jones "Reflections on the use of Grounded Theory in Interpretive Information Systems Research,” Electronic Journal of Information Systems Evaluation 6(1), 2003, pp.61-71.

[27] K. Eisenhardt "Building Theories from Case Study Research,”. Academy of Management Review 14(4), 1989, pp.532-550.

[28] R. Debreceny, G. Gray "IT Governance and Process Maturity: A Field Study” HICSS 2009 - Proceedings of the 42th Hawaii International Conference on System Sciences, Washington, DC, USA: IEEE Computer Society 2009.

[29] M. Larsen, M. Pedersen, K. Andersen "IT governance: reviewing 17 IT governance tools and analyzing the case of Novozymes A/S” HICSS 2006: Proceedings of the 39th Annual Hawaii International Conference on System Sciences. Washington, DC, USA: IEEE Computer Society, 2006.

3.3 Artikel 3: Applying Situational Method Engineering to the Development of Service Identification Methods

Veröffentlicht: Börner, R. (2010): Applying Situational Method Engineering to the Development of Service Identification Methods, in: Proceedings of the 16th Americas Conference on Information Systems (AMCIS 2010), 12.-15.08.2010, Lima, Peru, Paper 18.

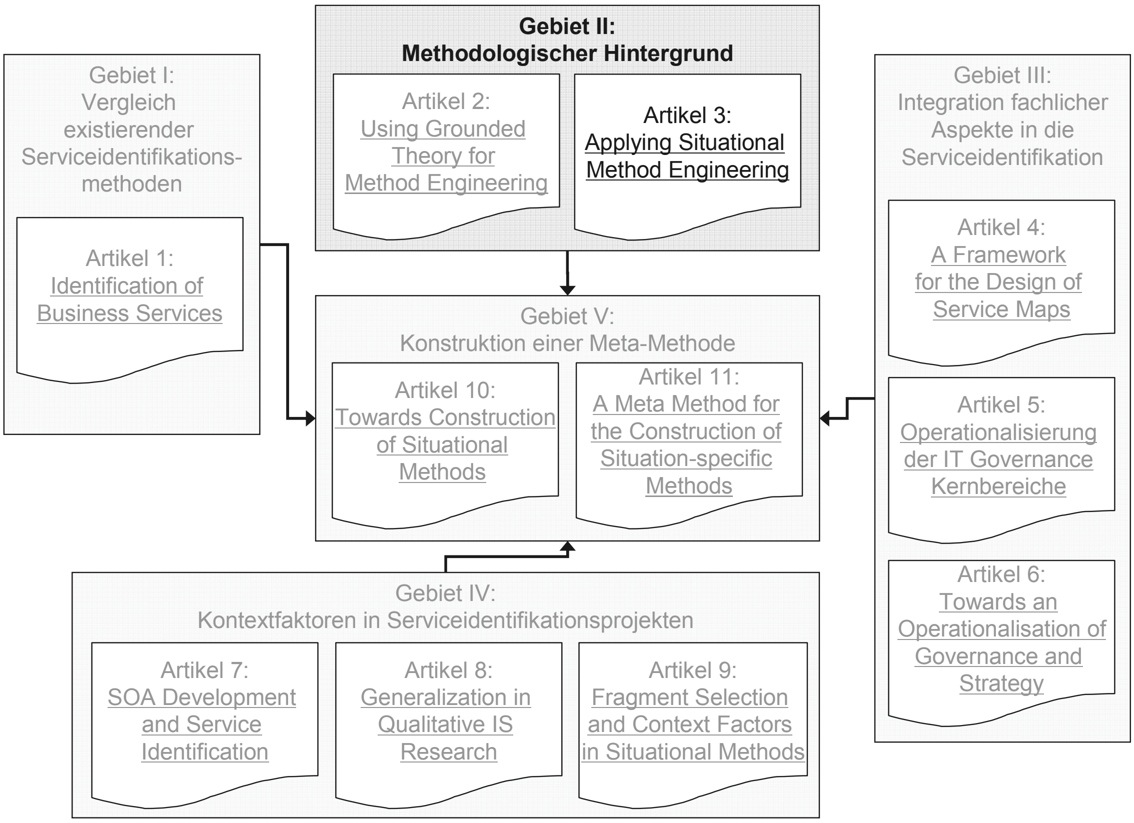

Abbildung 8: Artikel 3 im Kontext der kumulativen Dissertation

Applying Situational Method Engineering to the Development of Service Identification Methods

Rene Börner
ProcessLab, Frankfurt School of Finance & Management,
Sonnemannstr. 9-11, 60314 Frankfurt/Main, Germany

r.boerner@frankfurt-school.de

Abstract

Recently, situational method engineering (SME) has become more and more popular. Due to the fact that different situations require different methods, SME offers a flexible adaptation of methods. Furthermore, various approaches for the identification of services within service-oriented architectures have been proposed in the last few years. However, none of them provide a significant degree of situation-specific adaptability. Goal of this paper is to develop a meta model that can be used to construct service identification methods specific to certain situations. Therefore, ideas of SME are transferred into the realm of service identification. The meta model is further applied to a fictitious case.

Keywords

Service-oriented Architectures, Service Identification, Situational Method Engineering, Method Composition.

INTRODUCTION

Service-orientation is a highly recognized paradigm in enterprise architecture. There are lots of expected benefits related to service-oriented architectures (SOA) in a technical and in a business-oriented sense. Although the business-oriented benefits, like flexibility, reusability and standardization are of high importance, up to now, development of SOAs is mainly technically driven and most approaches consider technical aspects in the first place (Nadhan, 2004).

For the last couple of years many authors have been looking at the identification of services, which is one of the first and most important steps for an SOA implementation (for an overview see (Börner and Goeken, 2009b)). Interestingly, most existing methods to identify services are based on a one-fits-all approach. Only few consider a configuration of methods depending on different circumstances such as the goal of an SOA implementation. Even if the latter are considered, the scope of configurations is usually very limited.

The field of method engineering (ME) that has emerged since the early 1990s has been advanced by ideas of situational method engineering (SME) in the last decade. The central aspect behind it is that a fixed method is not suitable for all situations that occur in reality. Thus, methods have to be adaptable to different kinds of situations. The objective of this paper is to apply SME to the configuration of methods for service identification. Therefore, a meta model that transfers ideas of SME to the challenges of developing service identification methods is presented.

The paper is structured as follows: Section 2 discusses related work of both the field of SME and service identification. In section 3 a meta model for the development of service identification methods based on SME is suggested. One instantiation of the meta model at a concrete example in described in section 4. Finally, section 5 gives a summary, reflects on limitations and proposes future research on the topic.

RELATED WORK

Method engineering is a discipline in information systems research meant to "design, construct and adapt methods (…) for systems development." (Ralyté, Brinkkemper and Henderson-Sellers, 2007) Based on the fact that one method constructed at the time (t1) cannot fit to all conceivable conditions and circumstances when it is used at the time (t2), the concept of situational method engineering was created. Indeed, it is quite improbable that a rigid method developed from theory is applicable in a concrete setting without modification (Aydin, 2007). A method should thus be configurable. To provide for the adaptability of a method so called fragments are constructed and afterwards composed depending on the situation (Ralyté and Rolland, 2001).

ME literature offers many method elements that constitute method fragments as discussed in this section. The four most commonly used elements shown in Figure 1 are activities, techniques, results and roles (Gutzwiller, 1994; Heym, 1993; Brinkkemper, 1996; Brinkkemper, Saeki and Harmsen, 1999; Karlsson, 2002). Subsequently, according to (Cossentino, Gaglio, Henderson-Sellers and Seidita, 2006) a method fragment consists of precisely these elements.

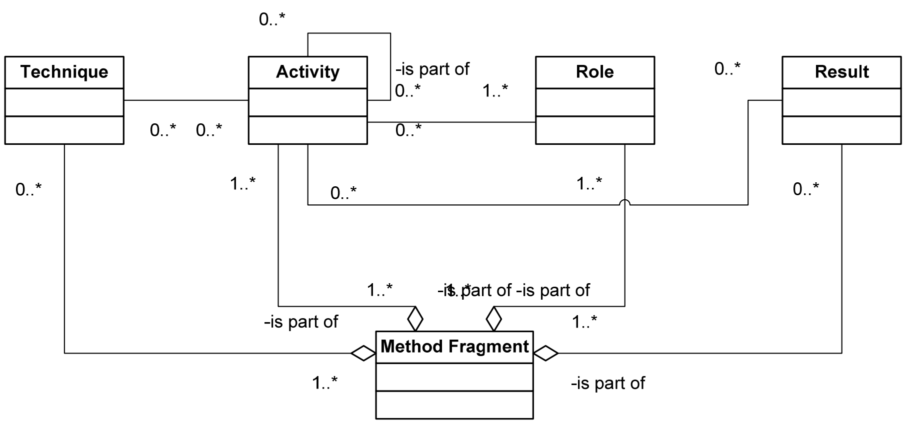

Figure 1. Concept of a method fragment

The term method fragment is used inconsistently in literature (Sunyaev, Hansen and Krcmar, 2008). (Agerfalk, Brinkkemper, Gonzalez-Perez, Henderson-Sellers, Karlsson, Kelly and Ralyté, 2007) define method fragments as "standard building blocks based on a coherent part of a method. A situational method can be constructed by combining a number of method fragments.” Some authors use the term synonymously with method chunk (Ralyté and Rolland, 2001). For the purpose of this paper, the notion of method fragments will be defined in the next section and used throughout the paper.

A number of approaches for service identification can be found in related literature. Herein, the analysis is limited to criteria such as configurability and facilitated method elements (i.e. activities, roles, techniques and results) that will be subject to further considerations in the following sections (see Table 1). A detailed analysis of existing methods can be found in (Börner and Goeken, 2009a).

	Klose et al. (2007)	**Böhmann & Krcmar (2005)**	**Winkler (2007)**	**Arsanjani et al. (2008)**	**Kohlmann & Alt (2007)**	**Kohlborn et al. (2009)**
Activities	++	++	++	++	+	++
Roles	--	--	--	o	--	o
Techniques	+	o	+	++	-	+
Results	+	++	+	++	+	++
Configurability	-	--	--	o	-	--
-- not existent - only implicitly o mentioned + defined/used ++ special focus						

Table 1. Method engineering elements and configurability in existing approaches

Usually, activities and results are described fairly well in all compared methods for service identification. The stakeholder-based approach by (Klose, Knackstedt and Beverungen, 2007) does not deal with configuration issues explicitly. Activities and techniques are described in detail whereas roles are not mentioned. Although a strict sequence of activities leaves no room for flexible adaptation, they acknowledge that certain preconditions can influence the process of service identification. The method presented by (Böhmann and Krcmar, 2005) neither discusses different roles nor describes techniques in depth. Although one of their first steps is the documentation of goals and scope (which should lead to different application contexts), this has no apparent influence on the method design. In Winkler’s approach (2007) any hints to roles are missing, too. She presents a strictly sequential proceeding. (Arsanjani, Ghosh, Allam, Abdollah, Ganapathy and Holley, 2008) note the importance of roles but omit a detailed description from their approach. They describe a fractal model and allow for an iterative sequence of activities. Their method can be adapted to the respective circumstances of a project but the authors fail to give guidelines how to configure it. Again, roles are not discussed in (Kohlmann and Alt, 2007) and even techniques are rarely mentioned. Their activities are in sequential order allowing iteration at only one point. Finally, (Kohlborn, Korthaus, Chan and Rosemann, 2009) consider roles in so far that their clear distinction between business services and software services necessitates that employees from functional departments as well as from the IT department are involved in the service identification process. They provide techniques but their method is not configurable for different application contexts. Table 1 shows the result of the conducted analysis.

A META MODEL FOR SITUATIONAL METHOD ENGINEERING OF SERVICE IDENTIFICATION METHODS

Configuration vs. Composition of Method Fragments

For the purpose of this paper any reasonable combination of method elements representing a coherent part of a method shall be referred to as method fragment. Figure 1 illustrates the concept of a method fragment as used herein. A method fragment does not have to include either all elements or one special element. It is meant to support the composition of a method to identify services. It may include even multiple instances of elements (e.g. more than one activity or technique).

According to (Bucher, Klesse, Kurpjuweit and Winter, 2007), there are two ways to build a configurable method, namely situational method configuration and situational method composition.

Situational method configuration follows the so called adaptive principle. This means that at design time (t1) of the method changes depending on the situation are explicitly allowed. There are precise instructions how an existing method has to be configured in certain contexts at time (t2). If situational method configuration is used, situational changes to a base method have to be foreseen and planned when a situational method is developed at time (t1). *Situational method composition* provides for a spontaneous combination of method fragments (orchestration) that does not have to be foreseen at (t1). There is no pre-defined base method that is adapted. Instead, method fragments are combined and aggregated as required at (t2).

Since all method fragments will be coherent parts of a method for service identification, situational method composition will be used in the following. After a situation is identified, a method for service identification will be composed at (t2). Method fragments are the building blocks of this method. Composition will be based on well-defined composition rules. These rules will be attached to the fragments (see Fig. 2).

What makes a situation?

Many authors agree that characteristics of a project have to be defined in order to describe a situation (Brinkkemper, 1996; Karlsson and Agerfalk, 2004). Still, according to (Bucher et al., 2007) they do not define sufficiently what constitutes a situation. In order to identify different situations, (Bucher et al., 2007) define a situation as combination of context and project types. The context is represented by environmental contingency factors. Contingency factors remain constant while the method is applied to solve a certain problem. A project type is inherent to a project and is transformed from its initial state to a target state by the method's activities.

In practice there are numerous context variables and each of them can have multiple parameter values. Examples can be found in the next section. In order to identify a situation, all reasonable context parameter value combinations (CPVC) have to be considered. All of these combinations make up one dimension necessary to identify a situation. For the purpose of this paper the above model is altered to better support the needs inherent to service identification in SOA. Thus, the second dimension is not a project type as proposed by (Bucher et al., 2007) but an SOA implementation goal (SOAIG) (see Fig. 2).

A matrix used for identifying relevant situations will therefore consist of the dimensions CPVC and SOAIG. Every intersection of this Situation Identification Matrix combining a CPVC with an SOAIG results in one specific situation (see Figure 3). Thus, the number of situations is determined by the number of CPVCs times the number of SOAIGs.

(# of situations) = (# of CPVCs) x (# of SOAIGs)

Keeping in mind that already the number of context parameter value combinations can be quite large the number of situations can be significant. An early limitation of context variables, parameter values, CPVCs and implementation goals will help to save time and reduce complexity. The earlier dispensable variables and parameters are eliminated the more effective an identification of situations will proceed. Figure 4 shows five steps (a) to (e) that are necessary to define situations within the Situation Identification Matrix:

(a) Context variables: Identification of influential environmental contingency factors is the first important step. There are a number of potential factors that could probably influence a situation and it is important to find the decisive ones. These factors can be derived from literature or expert interviews.

(b) Parameter values: All context variables identified in (a) have certain parameter values. The latter have to be defined in this second step. Although the parameter values small, middle and large are commonly used to describe a company's size (Brooksbank 1991), a differentiation between small and large might be sufficient for certain purposes. A finer-grained differentiation increases complexity and should only be used if the choice of the method is influenced by that.

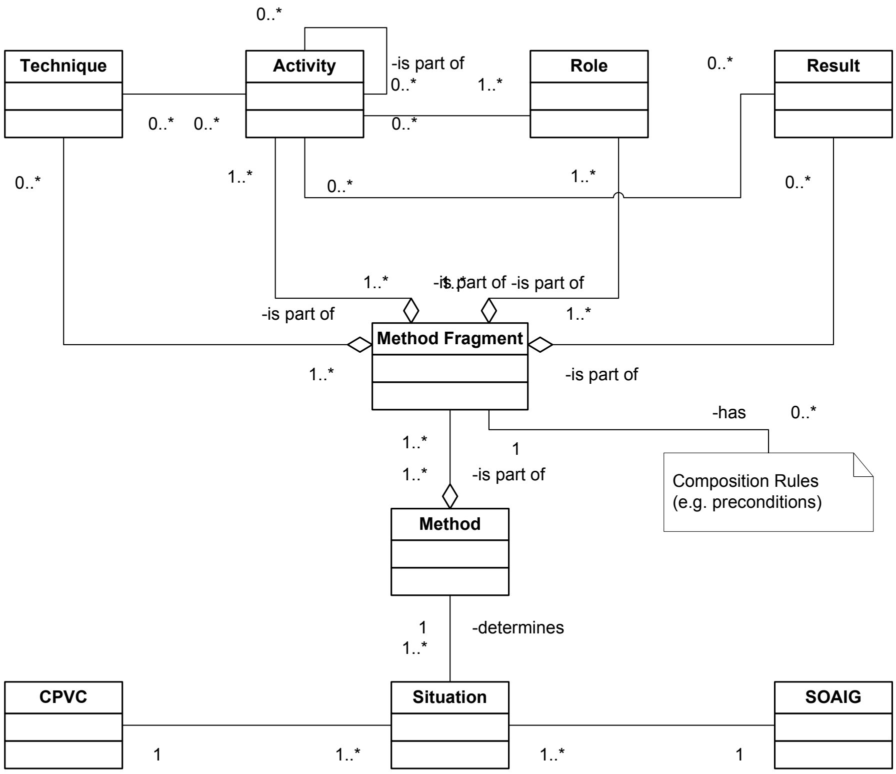

Figure 2. Concept of a situational method

(c) Context parameter value combinations: All parameter values are combined to constitute one dimension of the Situation Identification Matrix. At this point some combinations can be discarded. E.g. a small company combined with a project budget bigger than five million euros is not reasonable. Therefore, this parameter value combination can be removed from the matrix. From case to case combinations that can be excluded have to be determined. Certainly, steps (a) and (b) have a much stronger influence on the total number of combinations than the exclusion of single combinations described here.

(d) Similar to the choice of context variables described in (a), relevant SOA implementation goals (such as legacy system integration) have to be chosen. These goals form the second dimension of the matrix.

(e) Finally, the Situation Identification Matrix is used to illustrate all possible situations. Generally, all combinations of CPVCs and SOAIGs should lead to a situation. Still, there are some combinations that can be discarded. Combining a small start-up company with the goal to integrate legacy systems is not a reasonable presumption for instance.

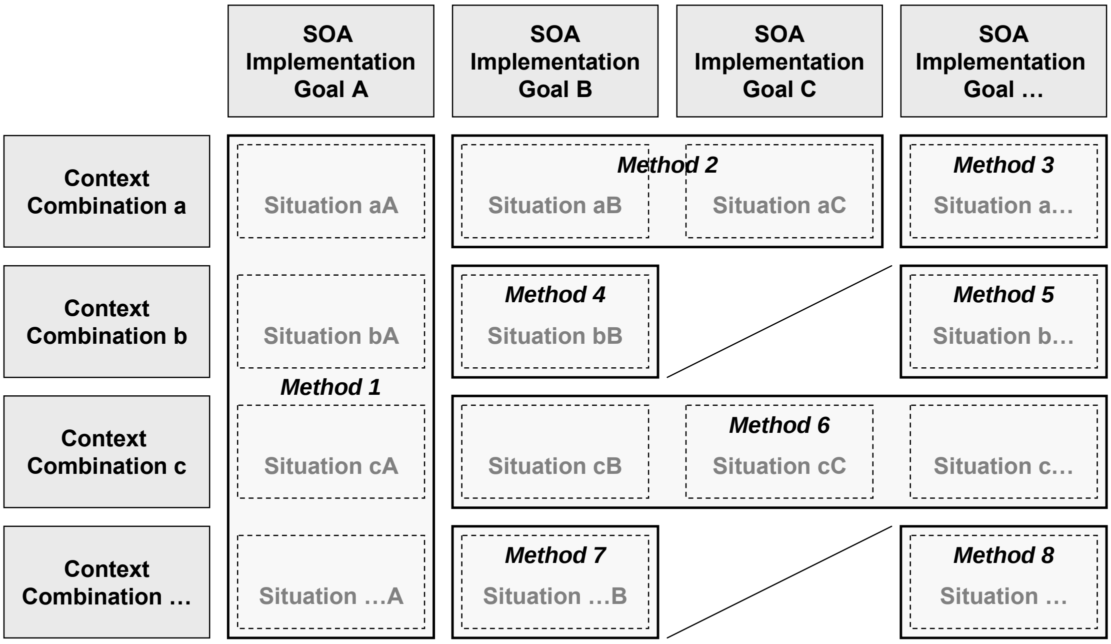

Figure 3. Situation Identification Matrix (following (Bucher et al. 2007))

Assigning Methods to Situations

Subsequently, every situation is mapped to a method that is tailored to the needs of the respective situation. The mapping of situations to methods is an n:1 relation, i.e. that every situation is assigned to exactly one method. However, methods may be used for several situations in case the preconditions of the latter are similar. As depicted in Figure 3 there are basically three types of methods. First, methods like number 3 or 4 apply to exactly one situation, i.e. they support one given set of context variables (in the form of a CPVC) and an implementation goal. Second, a so called goal-specific method like method 1 covers all situations with SOA implementation goal A irrespective of CPVCs. In this case the implementation goal is so dominant that any given CPVC does not influence the choice of a method. Third, methods like 2 or 6 are context combination-specific but span multiple implementation goals. In these cases only environmental factors determine the suitability of a method. The goal of an SOA implementation does not have a considerable effect on this selection.

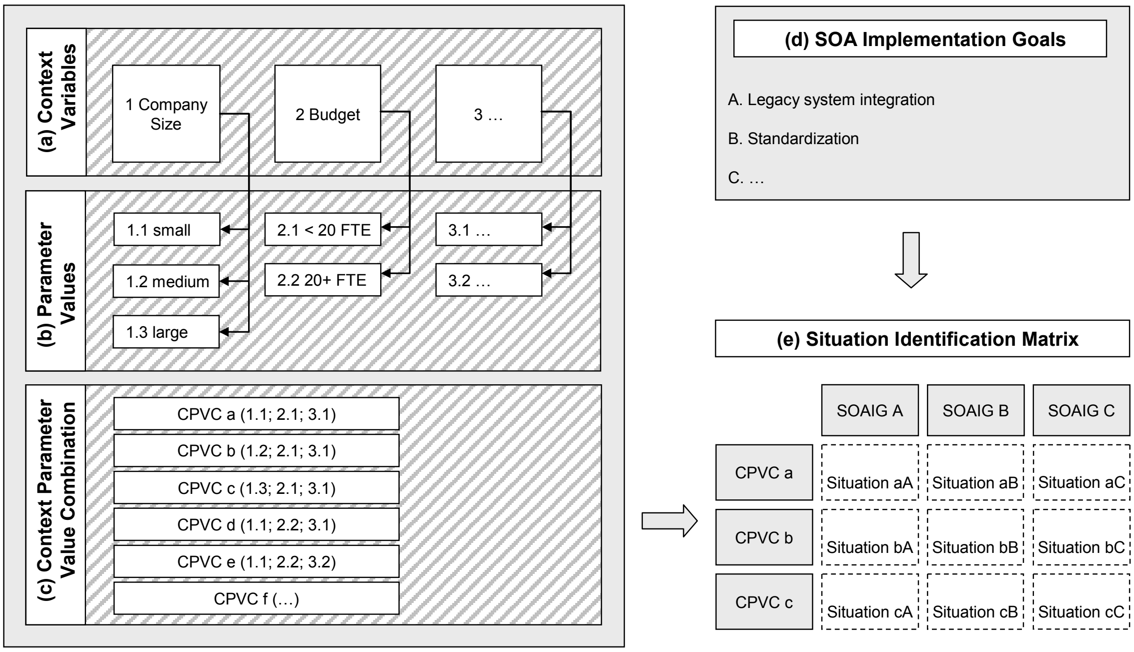

Figure 4. Five steps to define situations

INSTANTIATION OF A SERVICE IDENTIFICATION METHOD

The following section shows how a concrete method for service identification can be derived from the meta model described in the previous section. Firstly, some instances of method elements are shown and exemplarily combined to method fragments. Secondly, situations for service identification based on relevant contingency factors and SOA implementation goals are identified. Thirdly, a concrete example of a company is used to show how a method is composed out of method fragments. Hence, this method is tailored for the previously identified situation.

Building Method Fragments

In order to identify services, many activities, techniques, roles and results exist. Examples are shown in Table 2. Method fragments consist of these elements (see above). Although the latter are independent from one another in first place, they can be closely linked. For instance, the activity “create activity diagram” usually comes with the result “activity diagram”.

Method fragments should have meaningful names to improve composition. The broader process of service identification encompasses many steps that are represented by method fragments and finally part of an overall method. Examples for such fragments in this context are “Business Process Model”, “Breakdown of Business Processes”, “Strategic Alignment”, and “Positioning in the Value Chain”. It is important to design fragments as coherent and autonomous as possible to provide for an easier composition. However, there will be dependencies in using certain fragments that have to be reflected in composition rules.

Method Element	Instances
Activity	Identify sub processes Derive elementary activities Create activity diagrams Identify service candidates Normalize activity diagrams Analyze request frequency of services Explore technical feasibility Assess strategic potential
Technique	Business process modeling Asset analysis Goal service modeling Strategy questionnaire Governance questionnaire Technical feasibility checklist
Role	Business process owner (business department) Service owner (IT department) Service Design Unit (SDU) Service Excellence Center
Result	Activity diagram Business process documentation/landscape Service map

Table 2. Examples for method elements in service identification

The fragment “Breakdown of Business Processes” e.g. encompasses the activities “Identify Sub Processes”, “Derive Elementary Activities” and “Create Activity Diagrams”. The business process owner (role) is responsible for the activities and finally delivers activity diagrams (results). We assume that there is no distinct technique involved. However, existing business process models are a precondition for using this fragment (composition rule). Method fragments such as “Strategic Alignment” can require new organizational roles such as a service design unit (SDU) as proposed by (Börner, Looso and Goeken, 2009). Based on a questionnaire this SDU provides for an alignment of (IT) strategy and services that support business (Alter, Börner and Goeken, 2009).

Identifying Contingency Factors and Implementation Goals

While a huge number of context variables could be used to describe situations in which a method is ought to be applied, only relevant factors for the identification of services shall be described in the following. On the one hand the list should be comprehensive in order not to miss a crucial variable. On the other hand – due to the complexity problem described previously – the number should be as small as possible. Table 3 shows the herein discussed context variables and SOA implementation goals.

According to (Sedera, 2008) the *company size* is important in this context. The *geographic scope* of a company’s operation has to be considered in the process of service identification. It influences the cooperation of employees as far as activities are concerned and can lead to different roles and artifacts in the process of service identification. The *skills and qualification of employees* can vary significantly among projects and has to be taken into account (Becker, Knackstedt, Pfeiffer and Janiesch, 2007). There might be a necessity for external support if required know-how is not sufficiently available. The existence of a designated *IT department* is often bound to a company's size but should be considered explicitly as the existence of certain roles will depend thereon (Anderson, Howell-Barber, Hill, Javed, Lawler and Li, 2005). The available budget determines the number of full time equivalents (FTE) available in the project and is another important factor (Becker et al., 2007). The influence of other contingency factors, e.g. the legal form of the company, has been considered negligible in the context of service identification.

Context variables (parameter values)	**SOA implementation goals**
Company size (small, medium, large)	Integration of legacy systems
Geographic scope of operations (national, international)	Identification of outsourcing candidates
Employee qualification / skills (BPM skills, SOA skills, both, none)	Agility and flexibility of business processes
IT department (yes, no)	Standardization
Available budget (<20 FTE, 20+ FTE)	Provision of services for third parties

Table 3. Context variables, parameter values and SOA implementation goals

The goal of an SOA implementation and hence the identification of services is crucial for configuring a method. *Integration of legacy systems* is frequently a goal of SOA implementations (Erl, 2004; Heutschi, 2007; Arsanjani et al., 2008) especially in medium and large enterprises (Offermann, 2009). The identification of *outsourcing candidates* is another goal for SOA implementations (Beverungen, Knackstedt and Müller, 2008). In this case, costs, performance and strategic relevance of services must be analyzed. The *agility and flexibility of business processes* is a competitive advantage and strongly tied to the concept of SOA (Papazoglou, 2003; Heutschi, 2007). An alignment of business and IT is a necessary precondition to achieve this flexibility (Becker, Buxmann and Widjaja, 2009). In contrast to an enhanced flexibility on process services level, the *standardization* of basic services is meant to reduce redundancies in development and maintenance of IT and thus reduce costs significantly (Bieberstein, Bose, Walker and Lynch, 2005; Legner and Heutschi, 2007). A completely different perspective is taken by companies that aim at the *provision of services* for third parties. The former specialize on a small part of a value chain concentrating on their core competencies. They are able to generate economies of scale by providing services for many other companies typically – but not necessarily – belonging to the same industry sector. These cross company value networks become increasingly important e.g. in the banking industry (Kohlmann, 2007).

Composing a Method for a Concrete Situation

Exemplarily, one concrete situation, i.e. a combination of one SOA implementation goal and one context parameter value combination, shall be elaborated in the following. We assume that a small company (< 100 employees) with operations in only one country wants to provide services for third parties. This young company plans to use its core competencies in the production of industrial printing machines to establish itself as a layer player in a value network (Heuskel, 1999). Due to its size, there is no designated IT department. Instead, every business division takes care of its own IT infrastructure. Thus, no employee possesses any SOA know-how but some of the company's employees have considerable BPM skills. The budget is rather small so there is little scope to include external help.

Since the situation at hand is comprehensively described, a suitable method can now be selected. This method includes e.g. the fragment "Business Process Modeling" and uses the role "business process owner" to achieve a complete overview over the company's business processes. Fragments including roles such as "service owner" or "SDU" are not applicable due to the non-existence of an IT department. Thus, the method fragment "Strategic Alignment" as described above cannot be used in its current form. Since strategy plays an important role in this case, a similar fragment like "Service Provider Strategy" using other roles but pursuing the same goals has to be identified to complement the method. Furthermore, "Positioning in the Value Chain" is crucial to define the company's position within its value network. This fragment certainly contains results like service maps that support inter-organizational service integration (Kohlmann and Alt, 2009). The specifications given above determine a situation through their CPVC and SOAIG. Looking at the Situation Identification Matrix (Fig. 4) the most suitable method assigned to this situation can be chosen.

CONCLUSION AND FURTHER RESEARCH

So far, methods for service identification offer only limited support for a situation-specific adaptation. Situational method engineering taken from the realm of information systems development offers many approaches to provide precisely this kind of adaptability. This paper combines existing service identification methods with concepts from SME. In a first step, a meta model for method fragments and the identification of situations was presented. Afterwards, the applicability of this meta model was shown. All steps to develop a service identification method are demonstrated at a running example of a fictitious company. For this purpose, concrete value parameters for context variables and the SOA goal "Provision of services for third parties" are used.

However, there are some limitations to the findings presented in this paper. Context variables, their parameter values and the SOA implementation goals are taken from literature or based on experience. On the one hand the list might not be complete, i.e. important factors might not be included. Here, expert interviews could lead to further evidence about crucial contingency factors and relevant implementation goals. On the other hand, proof of the actual impact is weak for at least some variables and has to be improved through more empirical evidence. Currently, two case studies are evaluated in order to empirically underpin the relevance of contingency factors. A third case study is planned.

Finding a way to reduce the number of possible situations will be subject to further research. Using only five context variables, their respective parameter values and five implementation goals as described previously leads to 96 context parameter value combinations and thus to 480 situations in the Situation Identification Matrix. These combinations have to be limited to reasonable ones to produce fewer situations. Alternatively, methods have to encompass more situations in order to keep the number of methods manageable. Finding the right method for certain variables is not the main problem. Given a repository that contains all methods and a user interface that enables a search with given variables, retrieving a suitable method should be easy. However, the maintenance of such a large number of methods would pose a major problem.

Several examples for method fragments and their respective elements are mentioned in this paper. Method fragments are the building blocks for the composition of situational methods. Before this composition takes place, a repository containing method fragments has to be installed. Thus, reasonable combinations of method elements have to be combined to develop coherent fragments. Finding rules and giving advice on building such fragments is subject to further research. Particularly, an assignment of composition rules to each fragment needs to be elaborated in more detail. Designing method fragments for each case (i.e. every single company) is not feasible due to budget restrictions. Much more likely is the development of best practices for method fragments within certain industries that serve as a blueprint. Moreover, the applicability of method fragments may not be binary but could be marginally, fairly or highly relevant in a situation at hand. This again increases complexity.

Evaluation of the meta model so far rests on the case of one fictitious company. This can only indicate that the ideas of situational method engineering taken from information systems development are transferable to the area of service identification in service-oriented architectures. In order to properly evaluate the applicability and usefulness of the meta model, more evaluation methods should be applied.

REFERENCES

1. Agerfalk, P. J., Brinkkemper, S., Gonzalez-Perez, C., Henderson-Sellers, B., Karlsson, F., Kelly, S. and Ralyté, J. (2007), Modularization Constructs in Method Engineering: Towards Common Ground?, in Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.) *IFIP*, Springer, Boston.
2. Alter, S., Börner, R. and Goeken, M. (2009) Operationalisierung der IT-Governance-Kernbereiche für die Identifizierung und Gestaltung von Services, *Informatik 2009*, Lübeck, Germany, 3660-3674.
3. Anderson, D., Howell-Barber, H., Hill, J., Javed, N., Lawler, J. and Li, Z. (2005) A Study of Web Services Projects in the Financial Services Industry, *Information Systems Management*, 22, 1, 66-76.
4. Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S. and Holley, K. (2008) SOMA: A method for developing service-oriented solutions, *IBM Systems Journal*, 47, 3, 377-396.

5. Aydin, M. N. (2007) Examining Key Notions for Method Adaptation, in Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.) *IFIP Working Conference*, Springer, Geneva, Switzerland, 49-63.

6. Becker, A., Buxmann, P. and Widjaja, T. (2009) Value Potential and Challenges of Service-Oriented Architectures, *17th ECIS*, Verona, Italy.

7. Becker, J., Knackstedt, R., Pfeiffer, D. and Janiesch, C. (2007) Configurative Method Engineering, *13th AMCIS*, Keystone, Colorado, Paper 56.

8. Beverungen, D., Knackstedt, R. and Müller, O. (2008) Entwicklung Serviceorientierter Architekturen zur Integration von Produktion und Dienstleistung, *Wirtschaftsinformatik*, 50, 3, 220-234.

9. Bieberstein, N., Bose, S., Walker, L. and Lynch, A. (2005) Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals, *IBM Systems Journal*, 44, 4, 691-708.

10. Böhmann, T. and Krcmar, H. (2005), Modularisierung: Grundlagen und Anwendung bei IT-Dienstleistungen, in Thomas Herrmann, Uwe Kleinbeck and Helmut Krcmar (Eds.) *Konzepte für das Service Engineering. Modularisierung, Prozessgestaltung und Produktivitätsmanagement*, Physica, Heidelberg.

11. Börner, R. and Goeken, M. (2009a) Identification of Business Services, *15th AMCIS*, San Francisco, California, Paper 162.

12. Börner, R. and Goeken, M. (2009b) Methods for Service Identification, in Daniel Moldt, Juan Augusto and U. Ultes-Nitsche (Eds.) *7th International Workshop on MSVVEIS*, Milan, Italy, 76-84.

13. Börner, R., Looso, S. and Goeken, M. (2009) Towards an Operationalisation of Governance and Strategy for Service Identification and Design, *13th IEEE International EDOC Conference*, Auckland, New Zealand, 180-188.

14. Brinkkemper, S. (1996) Method engineering: engineering of information systems development methods and tools, *Information & Software Technology*, 38, 4, 275-280.

15. Brinkkemper, S., Saeki, M. and Harmsen, F. (1999) Meta-Modelling Based Assembly Techniques for Situational Method Engineering, *Information Systems*, 24, 3, 209-228.

16. Brooksbank, R. (1991) Defining the small business: a new classification of company size, *Entrepreneurship and Regional Development*, 3, 1, 17-32.

17. Bucher, T., Klesse, M., Kurpjuweit, S. and Winter, R. (2007), Situational Method Engineering - On the Differentiation of "Context" and "Project Type", in Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.) *IFIP Working Conference*, Springer, Boston.

18. Cossentino, M., Gaglio, S., Henderson-Sellers, B. and Seidita, V. (2006) A metamodelling-based approach for method fragment comparison, *11th International Workshop on Exploring Modeling Methods in Systems Analysis and Design*, Luxembourg.

19. Erl, T. (2004) Service-Oriented Architecture - A Field Guide to Integrating XML and Web Services, Prentice Hall, Upper Saddle River, NJ.

20. Gutzwiller, T. A. (1994) Das CC-RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen, Physica, Heidelberg.

21. Heuskel, D. (1999) Wettbewerb jenseits von Industriegrenzen: Aufbruch zu neuen Wachstumsstrategien, Campus, Frankfurt/Main.

22. Heutschi, R. (2007) Serviceorientierte Architektur, Springer, Berlin Heidelberg.

23. Heym, M. (1993) Methoden-Engineering - Spezifikation und Integration von Entwicklungsmethoden für Informationssysteme, Institut für Wirtschaftsinformatik, St. Gallen.

24. Karlsson, F. (2002) Meta-Method for Method Configuration - A Rational Unified Process Case, Linköping.

25. Karlsson, F. and Agerfalk, P. J. (2004) Method Configuration - Adapting to Situational Characteristics while Creating Reusable Assets, *Information & Software Technology*, 46, 9, 619-633.

26. Klose, K., Knackstedt, R. and Beverungen, D. (2007) Identification of Services - A Stakeholder-Based Approach to SOA Development and Its Application in the Area of Production Planning, *15th ECI*S, St. Gallen, Switzerland, 1802-1814.

27. Kohlborn, T., Korthaus, A., Chan, T. and Rosemann, M. (2009) Identification and Analysis of Business and Software Services - A Consolidated Approach, *IEEE Transactions on Services Computing*, 2, 1, 50-64.

28. Kohlmann, F. (2007) Service Modelling: A Hybrid Approach In Decomposed Financial Value Chains, *2nd International Workshop on EMISA*, St. Goar, Germany, 205-218.

29. Kohlmann, F. and Alt, R. (2007) Business-Driven Service Modeling, in Witold Abramowicz and Leszek Maciaszek (Eds.) *Business Process & Service Computing: First International Working Conference on BPSC*, Gesellschaft für Informatik e.V., Leipzig, Germany, 180-193.

30. Kohlmann, F. and Alt, R. (2009) Aligning Service Maps, *42nd HICSS*, Hawaii, 1-10.

31. Legner, C. and Heutschi, R. (2007) SOA Adoption in Practice, in Hubert Österle, Joachim Schelp and Robert Winter (Eds.) *15th ECIS*, St. Gallen, Switzerland, 1643-1654.

32. Nadhan, E. G. (2004) Seven Steps to a Service-Oriented Evolution, *Business Integration Journal*, 1, 41-44.

33. Offermann, P. (2009) Eine Methode zur Konzeption betrieblicher Software mit einer Serviceorientierten Architektur, Gito, Berlin.

34. Papazoglou, M. P. (2003) Service-Oriented Computing: Concepts, Characteristics and Directions, *4th International Conference on WISE*, IEEE Computer Society, Rome, Italy, 3-12.

35. Ralyté, J., Brinkkemper, S. and Henderson-Sellers, B. (2007), Situational Method Engineering: Fundamentals and Experiences, in Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.) *IFIP*, Springer, Boston.

36. Ralyté, J. and Rolland, C. (2001) An Approach for Method Engineering, in H.S. Kunii, S. Jajodia and A. Solvberg (Eds.) *20th International Conference on Conceptual Modelling*, Springer, Yokohama, Japan, 471-484.

37. Sedera, D. (2008) Does Size Matter? Enterprise System Performance in Small, Medium and Large Organizations, *2nd Workshop on 3rd Generation Enterprise Resource Planning Systems*, Copenhagen, Denmark.

38. Sunyaev, A., Hansen, M. and Krcmar, H. (2008) Method Engineering: A Formal Approach, *17th International Conference on Information Systems Development*, Paphos, Cyprus.

39. Winkler, V. (2007) Identifikation und Gestaltung von Services, *Wirtschaftsinformatik*, 49, 4, 257-266.

3.4 Zwischenfazit

Konfigurierbare Methoden basieren auf Fragmenten, die bisher häufig intuitiv oder rein auf Basis vorhandener Literatur entwickelt wurden. Grounded Method Engineering bietet eine Ergänzung zu bisher benutzten Forschungsmethoden und ermöglicht eine empirisch fundierte Konstruktion von Fragmenten. Fachwissen (bspw. aus Experteninterviews) kann auf diese Art in entsprechende Methoden transferiert werden. Das Prinzip wurde an einem Beispiel aus dem Bereich IT-Governance demonstriert, ist aber genauso auf den Bereich der Serviceidentifikation übertragbar.

Das Situational Method Engineering konfiguriert Methoden in Abhängigkeit von Situationen. Die Identifikation dieser Situationen und die Vorgehensweise bei der Konfiguration sind Gegenstand dieses Forschungsgebiets. Artikel 3 hat aufgezeigt, wie diese Konzepte auf den Bereich der Serviceidentifikation übertragbar sind. Auf Basis von domänenspezifischen Kontextfaktoren und SOA-Implementierungszielen können Situationen bestimmt werden, die als Grundlage für die Konfiguration von Serviceidentifikationsmethoden dienen.

4. Gebiet III: Integration fachlicher Aspekte in die Serviceidentifikation

4.1 Einleitung

Bei der Konfiguration von Serviceidentifikationsmethoden müssen projektspezifische Charakteristika berücksichtigt werden. Die nächsten drei Artikel demonstrieren, dass neben strategischen Zielen einer SOA-Implementierung oder Rahmenbedingungen wie der Unternehmensgröße auch sehr konkrete Vorhaben unterstützt werden können.

Für fachlich getriebene Identifikationsprojekte können Service Maps, deren Erstellung von bestimmten Methodenfragmenten unterstützt wird, eine große Rolle spielen. Um die Gefahr heterogener Service Maps innerhalb eines Unternehmens zu minimieren, wird in Artikel 4 ein Design-Framework entwickelt. Der fünfte Artikel beschäftigt sich mit der Einbeziehung von IT-Governance und insbesondere der fünf so genannten „Focus Areas". Hierzu werden Techniken wie ein Fragebogen und neue Rollen wie bspw. eine „Service Design Unit" entwickelt. Artikel 6 baut direkt darauf auf und bezieht zusätzlich noch strategische Aspekte in die Serviceidentifikation mit ein.

4.2 Artikel 4: A Framework for the Design of Service Maps

Veröffentlicht: Kohlmann, F., Börner, R., Alt, R. (2010): A Framework for the Design of Service Map, in: Proceedings of the 16th Americas Conference on Information Systems (AMCIS 2010), 12.-15.08.2010, Lima, Peru, Paper 39.

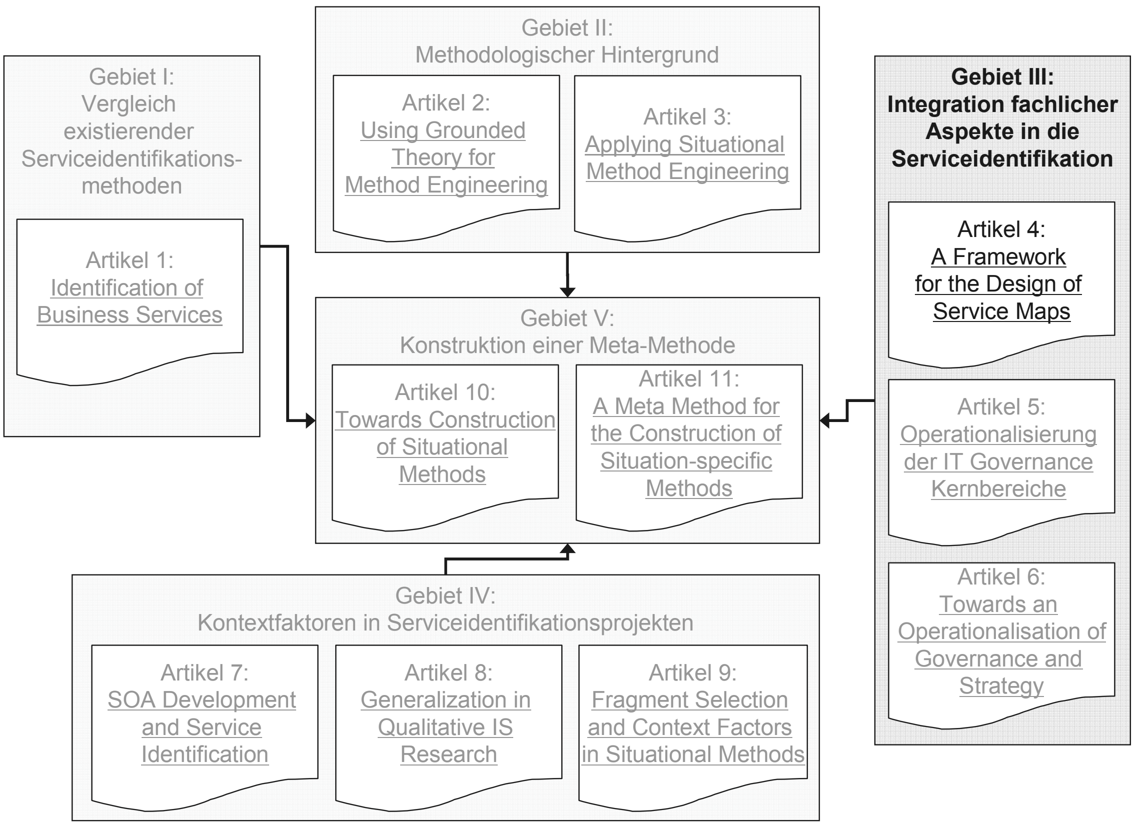

Abbildung 9: Artikel 4 im Kontext der kumulativen Dissertation

A Framework for the Design of Service Maps

Falk Kohlmann
Information Systems Institute
University of Leipzig
Grimmaische Strasse 12, D-04109 Leipzig
kohlmann@wifa.uni-leipzig.de

René Börner
Frankfurt School of Finance & Management
Sonnemannstraße 9-11, D-60314 Frankfurt am Main
r.boerner@frankfurt-school.de

Rainer Alt
Information Systems Institute
University of Leipzig
Grimmaische Strasse 12, D-04109 Leipzig
rainer.alt@uni-leipzig.de

Abstract

The concept of service-oriented architecture (SOA) is recognized as an important enabler for business transformation and application integration. Service maps emerge when individual services are (pre)configured on various architectural levels. For example, business-oriented service maps sustain the communication and coordination among participants within and between businesses. Difficulties occur when, based on different service design strategies, heterogeneous service maps are created which need to be aligned. A methodological approach to establish a systematic design process for such service maps within companies or business networks is needed.

Keywords

Service map, service-oriented architecture, service governance

INTRODUCTION

SOA is recognized as an important concept for business transformation and is discussed from a technological and a business perspective. Whereas the technical view conceives SOA as a "paradigm that supports modularized exposure of existing application functionality to other applications as services" (Nadham 2004, p.41) the business perspective denotes the ability of reusing tasks and processes by providing them at one location (OASIS 2006). SOA presents itself as a suitable option for improving the integration of heterogeneous application environments as well as the sourcing of entire or fractional business processes in a business network (Baskerville et al. 2005). For the implementation and governance of SOA, different instruments emerged. Among them are service maps which structure services by visualizing them along with their relationships. Based on different SOA design strategies the risk of various incompatible service maps increases. Consequently, methodologies integrating these different service maps are required (Kohlmann and Alt 2009a). A contribution to such methodologies would be a framework which incorporates influencing factors for the design of a service map to facilitate the understanding of the basic ideas behind a specific service map, such as its goal or its focus. The Banking Industry Architecture Network (BIAN) e.g. develops a service map whose structure is influenced by the boundaries of common core banking solutions. Therefore (reference) data (information about business partners or market data) is visualized as one out of five parts of the service map. Contrary business departments of a bank would typically organize a service map according to the supported business processes. Integrated in SOA and service design procedures (Erl 2007; Kohlborn et al. 2009) such a framework would further foster a systematic and comprehensible design of service maps for a specific context (e.g. application integration, industry reference). The framework proposed in this paper is a first attempt to obtain such a systematic design support.

For the purpose of this paper a multi-method research design was chosen based on a design science approach (Galliers 1992). Design science is a problem-solving paradigm (Hevner et al. 2004) that creates innovative artifacts such as models, methods, prototypes (Simon 1996) or – in this case – a framework. These artifacts are to-be models and cannot be validated with existing, i.e. historical data. This research follows the seven guidelines for design science research proposed by (Hevner et al. 2004), e.g. it designs an artifact (i.e. a framework) that helps to solve an existing problem (systematic design of service maps) and contributes to research in the field of service-oriented architectures. Finally, the framework is evaluated by a case study which was conducted based upon the recommendations elaborated by (Dubé and Paré 2003) and the four requirements of MIS case study design according to (Lee 1989). As data source for the single case study direct observation in combination with 12 structured interviews, six workshops and project documentation was used.

Section 2 provides the theoretical foundation for SOA and briefly discusses related work. Section 3 proposes a framework for designing service maps. Section 4 presents the case study of a Swiss universal bank exemplifying the applicability of the framework. Section 5 summarizes the results and provides an outlook.

SERVICE-ORIENTED ARCHITECTURE AND SERVICE MAPS

Services, the core of any SOA, provide modular business-oriented functionalities which may be bundled along business processes (Erl 2007; Kohlborn et al. 2009; Kohlmann and Alt 2009b). By using standardized interfaces and centralized repositories, services may be flexibly recombined and efficiently reused. Legacy systems can be integrated or replaced by a modularization brought about by SOA (Papazoglou and Georgakopoulos 2003). This improved *agility* of business processes is generally linked to increased competitive advantage (Schelp and Aier 2009). Combined with standardization of interfaces this contributes to inter-organizational cooperation between service providers and service users. However, as the granularity of services differs, various service layers and typologies emerged to structure SOA (Bell 2008; Erl 2007; Rosen et al. 2008). Well-defined interfaces (i.e. input, output, protocols) enable an outsourcing of services. Consequently, preconfigured service reference models (e.g. from software vendors such as SAP or IBM or communities such as BIAN) are established to reduce initial costs and to indicate reusable services. Still such initiatives provide mainly industry-specific solutions.

In this context service maps are regarded as instruments to structure services by visualizing them within a specific domain. Furthermore, they can be used to exemplify relationships and dependencies. However service maps neither imply the rules for the design nor the specification of the services as an architecture would be. Due to different service design strategies (e.g. top-down, bottom-up) the risk of the emergence of heterogeneous service maps is increased. Nevertheless, service maps on different granularity levels such as presented in (Kohlmann and Alt 2009a) are necessary as they (1) support the analysis and design of business models, (2) provide a comprehensive understanding among IT and business and/or (3) reduce integration costs of applications. As service maps reduce complexity by structuring services and exemplifying relationships, business and IT representatives obtain an instrument that structures business requirements and their implications on IT systems (Kohlmann and Alt 2009a; Rosen et al. 2008).

Despite the broad discussion of SOA, current literature features a lack of contributions which provide in-depth insights into the value of service maps. (Kullvén and Mattsson 1994) propose a (process-oriented) service map as structural basis for an economic model to assign costs and revenues. In turn, (Jiang and Willey 2005) propose a service map as data structure with deployment information to support the discovery of web services. Moreover, several communities and initiatives evolved trying to define service maps for different industries, such as the above mentioned BIAN (Bills 2009). Consequently, prior research (Kohlmann and Alt 2009a) suggested a procedure as part of a possible methodology to align these heterogeneous service maps. However, no attempt has been made so far to provide a framework for a structured service map design.

FRAMEWORK FOR THE DESIGN OF SERVICE MAPS

This section proposes a framework for the design of service maps based on different contingencies. The framework is based on the assumption that an appropriate design of a service map fits external factors such as organizational size. E.g. the classification of service ownerships can be limited in small firms compared to large firms. Similar dependencies have been outlined by the contingency theory of organizational design. Following (Donaldson 2001), an organizational structure (e.g. hierarchical, functional) which “fits” (internal and external) contingency factors, such as size or strategy, is more effective with regard to profitability, efficiency or innovation rate. Based on contingencies an organization is able to adapt its organizational design. Hence, a “misfit” between contingencies and organizational design results in a negative impact on a company’s performance. Accordingly, the design of a service map should “fit” its contingencies. The proposed framework consists of two dimensions: (1) a building block dimension differentiating relevant design elements for building service maps and (2) a design dimension incorporating contingencies on these design elements.

Building block dimension

To enhance existent and prospective service and SOA design methodologies the framework should reflect current perspectives and granularities (Erl 2007). Hence, the building block dimension considers perspective and typology supplemented by relationships such as dependencies between services. Moreover to support governmental issues (Börner et al. 2009) being key for the introduction of SOA (Aier et al. 2008) the framework incorporates an ownership building block.

As the development of the SOA concept as well as any specific SOA is characterized by the attempt to bridge business and technology orientation the structure of a service map can incorporate a business- and/or technology-oriented **perspective**. The basis for a business-oriented service design can be business processes and business networks. Besides using service maps at design time to structure services and their interdependencies, several approaches are aiming at dynamic service integration at runtime using ontologies (Vitvar et al. 2007). Besides business processes (e.g. payments, sales) and business networks (e.g. SWIFT-*Society for Worldwide Interbank Financial Telecommunication*), specific business domains (e.g. financial or automotive industry), or within a firm group commonly expressed as business departments (e.g. trade, human resources), can be used to structure the service map. These business domains can be organized using a shared service center approach (Janssen and Joha 2006). In turn basis for a technological-oriented service map is the application architecture of a company or business network comprising application domains and system boundaries.

Following the distinction between a business and a technological perspective on SOA two service categories may be distinguished (see Figure 1). Business services ('tier 2' services) represent functionality of a specific business activity or transaction (e.g. *PricingRuleService*, *ForeignCurrencySupplyService*) (Bonati et al. 2006; Kohlborn et al. 2009). Due to the entitlement of SOA as intermediate architecture layer between business and technological architecture an alignment with processes and strategy is required. This involves not only cataloguing all services in a directory with little structure but also preconfigured bindings of business services in so-called service clusters (Bell 2008). The clusters or 'tier 1 services' combine several services of finer granularity due to their logical and functional proximity (e.g. *Rating, LoansManagement*). These business service clusters provide a connection to the company's business models. Focusing on technical services a differentiation between services covering domain independent IT functionality and services comprising domain specific basic functionalities would enhance transparency, reusability and governance. Consequently, application services ('tier 3' services) are focusing on independently usable and elaborately specified functional components (e.g. *CreateLoanContract, ManageBusinessPartnerRole* for the introduction of SOA) (Kohlmann and Alt 2009b; Rosen et al. 2008). Infrastructure services ('tier 4' services) finally encapsulate technical capabilities independent of any business domain, such as *accessDatabase* or *checkIPConnection* (Rosen et al. 2008).

Service maps express **relationships.** These service-to-service relations can bridge the service tiers (vertical relation) or remain on the same level (horizontal relation) (see Figure 1). A horizontal relation relates to the fact that a service may need another service in order to execute its function whereas a vertical relation incorporates the concept that a service of higher granularity composes several services of lower granularity. Besides these relationships within a service-layer, a service map is able to exemplify the linkage to systems (implementing a specific service), processes (using a specific service) or business roles (providing a specific service within a business network).

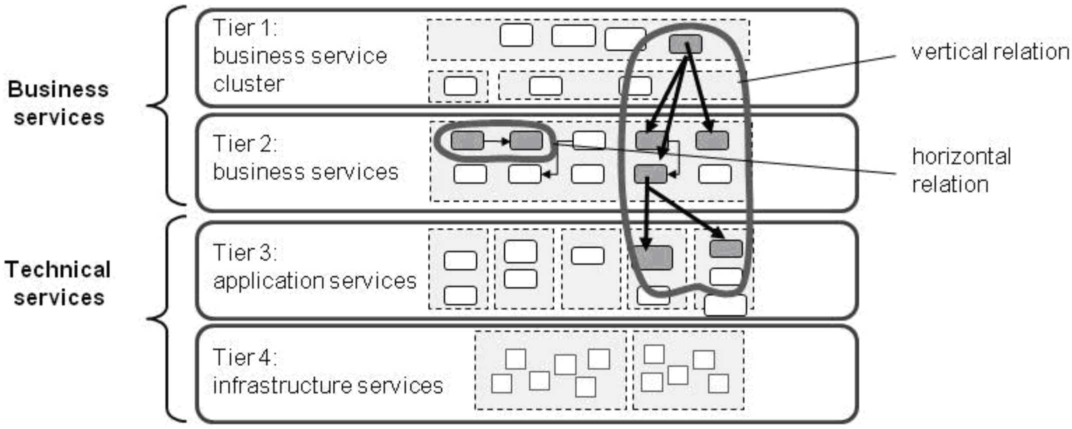

Figure 1. Service typology and service-to-service relationships

A key requirement for SOA is the implementation of a supportive organizational structure (roles, responsibilities) as part of SOA governance (Brown et al. 2006), which should consider both the enterprise and the network view (Kohlmann and Alt 2009b). Current contributions depict high maintenance efforts and lower reusability for service variants (Schelp and Aier 2009). Defined ownerships would enhance transparency in service management and therefore contribute to sustainability. Already a service map as governmental instrument can visualize associated ownerships. The building block **ownership** comprises the four types commonly used within a RACI matrix (responsibility, accountability, consulted, informed) extended by the three types realization, utilization and coordination as proposed by (Kohlmann and Alt 2009b).

Design dimension

Instantiations of the four building blocks of a service map, namely perspective, typology, relationship, and ownership are influenced by contingencies. The latter form the so called design dimension of the matrix shown in Figure 2. In the following, SOA goals, focus, company size and SOA maturity level as the most influential factors (Kohlborn et al. 2009; Papazoglou et al. 2006, p. 3) will be explained and their effect on the design of service maps will be outlined.

		Design dimension (contingencies)			
	Designing service maps	SOA Goal	Focus	Company size	SOA Maturity level
Building block dimension	Perspective				
	Typology / Granularity				
	Relationships				
	Ownership				

Figure 2. Dimensions of the framework

SOA implementation may follow different **goals** (see e.g. (Arsanjani et al. 2008; Erl 2007). A frequent goal is the optimization of the IT architecture in the sense of legacy systems integration (Legner and Heutschi 2007). Using SOA to evolve a standardized integration infrastructure (Zimmermann et al. 2004) includes a documentation of interfaces and integration of platforms as necessary preconditions. Eventually, this leads to enhanced interoperability of existing legacy systems. To optimize IT architecture, SOA can also be applied to decouple application domains. Thus, small changes concerning functionality can be implemented locally without affecting other domains. A higher degree of reusability leads to cost savings and helps to avoid redundancies. On the other side the business perspective focuses on the centralization of business functionalities in order to flexibly bundle business processes (Papazoglou and Georgakopoulos 2003) and to align business processes and IT (Becker et al. 2009). A process-oriented identification of services supports this transition (Börner and Goeken 2009). Agile and flexible business processes are the foundation for an effective handling of process variants. Furthermore, they provide a competitive advantage enhancing a company's sustainability (Becker et al. 2004). Depending on the implementation goal, the design of the building blocks differs (see **Table 1**). Applying the contingency e.g. to the required relationships IT architecture optimization visualizes system-to-services-relations whereas business functionality centralization concentrates on process-to-services-relations.

Companies are increasingly reducing their vertical integration to focus on core competencies by sourcing business functionalities. Hence, a company's business networkability becomes a crucial success factor. For instance, many banks have embarked on sharpening their core competencies and the option of offering services to other banks (Homann et al. 2004). The notion of networkability refers to "both the internal and external capability of organizations to collaborate with each other at the level of both business processes and underlying ICT infrastructure" (Wigand et al. 1997). Depending on the internal or external perspective, the design of a service map may differ. Given an internal focus, the design will concentrate on business domains within a company e.g. to assign responsibilities for certain services. Here, the relationships between business processes and IT infrastructure are crucial. From the technological perspective the concept of SOA represents a promising instrument to implement networkability on the network level. By separating application functionality into modules and using standardized interfaces for invoking these modules (or services) SOA has the potential to link functionality from various providers within a business network. Thus, given an external focus, attention is drawn to the roles companies adopt within a network. (Terlouw et al. 2009) regard "**focus**" as an influential factor on the delivery strategy of SOA. Service maps are influenced by the choice of the focus especially as far as vertical and horizontal relationships are concerned.

The patterns behind the concept of SOA are relevant to companies of all sizes and industries (Aier et al. 2008). Nevertheless, the resources available for implementing and governing SOA as well as the size of the application architecture depend on a **company's size**. (Ciganek et al. 2006) conclude that company size is an influencing factor for the adoption of new technologies. (Sedera 2008) has empirically shown the influence of a company's size at the case of ERP systems. The study reveals that "larger organizations receive higher benefits compared to their small and medium counterparts", whereas small organizations express a higher reliance on their systems. It is common to classify small, medium and large enterprises (Brooksbank 1991). However, the differentiation base differs. (Brooksbank 1991) proposes the following criteria: (1) number of employees (100 to 400 equals medium) and (2) sales turnover ($3.75 million to $30 million equal medium). Thus, the design of a service map depends not only on the fundamental goal of a SOA implementation but also on a company's characteristics such as size. Implementing SOA in small companies requires scaling down leading to different service maps. Issues related to the ownership of services e.g. are less complex and less challenging in small companies (see Table 1).

Finally, **maturity** influences the delivery strategy of SOA (Terlouw et al. 2009). SOA maturity models are used to classify the status of SOA implementations within a company. Broadly accepted models are the SOA Maturity Model (SOAMM) comprising five levels and being introduced by Sonic Software (Bachman 2005) and the Service Integration Maturity Model (SIMM) differentiating seven levels and being published by (Arsanjani and Holley 2006). The latter will be used to demonstrate how the maturity of SOA influences the design of service maps. The first three levels of SIMM focus on function-orientation, whereas the latter four focus on service-orientation (Arsanjani and Holley 2006). Hence, only level 4 to 7 (simple services, composite services, virtualized services and dynamically reconfigurable services) are considered as the contribution of service maps are limited in function-oriented businesses (level 1 to 3). Particularly the perspective and relationships are strongly influenced by the maturity level. E.g. level 4 (simple services) tend to a technological perspective and system-to-service or service-to-service-relations. Table 1 shows a detailed view on how the contingencies influence properties of the building blocks.

Building block dimension \ Design dimension		SOA Goal		Focus		Company size			SOA Maturity level			
		Optimization IT architecture	Business functionality centralization	Company (internal)	Network (external)	Small	Medium	Large	Simple Services	Composite Services	Virtualized Services	Dynamically reconfigurable services
Perspective	Business		x (PM)	x (BD)	x (NM)	x (BD)	x (PM,BD)	x (NM, PM,BD)		x (PM, SCC)	x (PM, CDC)	x (BO)
	Technology (IT systems)	x					verf.	x	x			
Typology	Business services		x (tier2)	x (tier2)	x (tier2)	x (tier1,tier2)	x (tier2)	x (tier1,tier2)		x tier2	x tier2	x (tier2)
	Application services	x (tier3)	x (tier3)	x (tier3)	x (tier3)		x (tier3)	x (tier3,tier4)	x (tier4)	x (tier3)	x (tier3)	x (tier3)
Relationships	Role-service				x	x						
	Process-service		x	x				x		x	x	
	Service-service		x (vert.,hor.)				x (vert.,hor.)	x (vert.,hor.)				x (vert.)
	System-service	x						x	x	x		
Ownership		R,RE	CO,A	RACI, RE,U,CO	RACI, RE,U,CO	RA	RA,CO	RACI, RE,U, CO	R,RE	RA	RA	RACI, RE,U,CO

Legend:
In general: x = applied; verf.= used for verification;
Structure: NM = network model; BD = business departments/business domains; PM = process model; BO = business ontology; SCC = supplemented by shared service center concept; CDC = supplemented by centralized data center
Typology: tier 1 = business service clusters; tier 2 = business services; tier 3 = application services; tier 4 = infrastructure services;
Relationships: hor. = horizontal; vert. = vertical;
Ownership: R = responsible; A = accountable; C = consulted; I = informed; CO = Coordinated; RE = realized; U = utilized

Table 1. Framework for the design of service map

APPLICATION AT A SWISS UNIVERSAL BANK

General setting at the Swiss universal bank

The bank used for the case study is a vertically integrated bank with a multi channel approach. Its business processes include loans, payments and securities as well as most of the transaction related business processes (such as monitoring), transaction spanning processes (such as product development), and support processes. The bank already has elaborated process architecture with business processes documented on different granularity levels. However, consolidated cross-bank business process architecture is still missing. Current challenges are the integration of legacy systems and heterogeneous business processes. A first step towards establishing a service-oriented architecture has been made by identifying business services mainly based on 80 business processes and 37 business objects. Services are seen as a possibility to reduce costs and complexity, to integrate heterogeneous application landscapes, and to standardize and enhance sophisticated pricing models. A service map constitutes the key element in building a business-oriented service architecture in this case.

Designing the service map

Based on the general requirements, a project consisting of IT and business representatives was launched to develop a service map bridging existing IT business boundaries. The application of the framework followed the process exemplified in **Figure 3**. Before designing the service map, the relevant business architecture models such as process, application or sourcing models, were analyzed (step I.1), followed by a consideration of existing strategies (step I.2). The analysis phase revealed several criteria and general conditions for the design of the service map: (1) the bank aims to develop a cross-functional service model by applying a business-oriented service design approach; (2) service design is based upon business processes, business models, regulations and sourcing capabilities; (3) service design includes bottom-up verification by linking the business services with application services or systems functionalities; (4) the design approach supports a hierarchical service typology and (5) the structure of the service map shall facilitate the linkage of services on different granularity levels and/or sources.

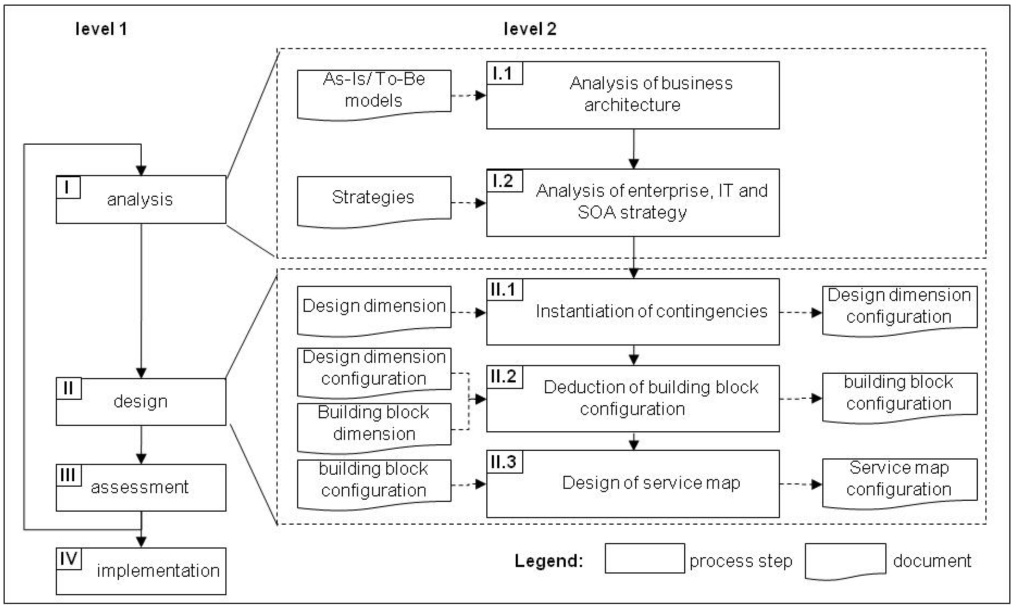

Figure 3. Service map design process

Based on these characteristics the bank was able to instantiate the contingency factors of the design dimension (step II.1). As primary goal centralization of business functionalities excelled IT optimization. The perspective of SOA as well as the service map has a clear internal focus due to the in-house provision of significant functionalities, In terms of size the bank belongs to the largest banks in Switzerland. Maturity finally does not influence the service map as the bank is function-oriented comparable to level 3 of SIMM. Following this setting the bank deduced the configuration of the building block dimension using the matrix in **Table 1** (step II.2). **Table 2** shows the associated design elements.

		Goal	Focus	Company size	SOA Maturity
		Centralization of business functionalities	Company (internal)	Large	
Perspective	Business	● (PM)	○ (BD)	◗ (NM,PM,BD)	
Typology	Business services	● (tier2)	● (tier2)	● (tier1,tier2)	
	Application services	● (tier3)	● (tier3)	◗ (tier3,tier4)	
Relationship	Process-service	◗	◗	◗	
	Service-service	●		●	
	System-service			○	
Ownership		○(CO,A)	◗ (RACI,RE, U,CO)	◗(RACI,RE, U,CO)	
Legend (abbreviations see Table 1):	○ Not applied ◗ Partially applied ● Fully applied			not applicable	

Table 2. Application of the design view to the service map of the Swiss bank

The perspective of the service map is business-oriented based on the bank's goal to develop a cross-functional service model by applying a business-oriented service design approach, Consequently, the structure of the bank's service map reflects the structure of the bank's process architecture: (A) governance/management, (B) sales and (C) support. The remaining parts of the process architecture (1) money transactions, (2) finance and real estate, (3) assets and investment management as well as (4) trade and capital market have been restructured to (D) transaction-specific and (E) transaction-spanning as services can be used in several business processes. Although business department boundaries (BD) should influence the structure of the service map, they had no implications due to the banks current change in the organizational structure (see cell 4.3 in Table 2). According to the proposed framework, business and application services were differentiated. To establish a banking specific SOA framework the business services were identified and specified at first. The identification and specification of the 137 business services took eight months in total. A follow-up project will define application services. Therefore, a comprehensive service map as implied by the framework has not been achieved yet. Nevertheless, the horizontal extent of the service map has been reached as it covers all core business processes of the bank. Contrary to the framework, infrastructure services are not incorporated in the service map (see cell 5.5 in Table 2).

The developed service map was used to structure business services along with their relationships and dependencies. The process-service relations were documented in a separate service catalogue and not in the service map itself (see cells 3.6 to 5.6). Same applies to the different ownerships. **RE**alization and **U**sage are exemplified in the service map whereas RACI is documented in a separate spreadsheet. **CO**ordination has not been used due to the internal focus of the project and the bank's internal SOA initiative. **Figure 4** shows tier 1 of the service map.

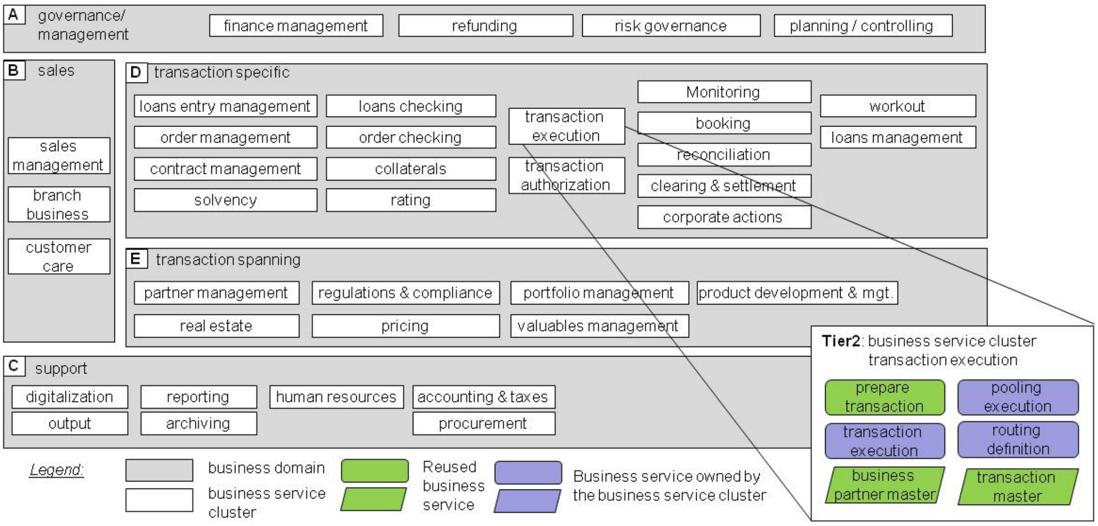

Figure 4. Tier 1 of the bank's service map

CONCLUSIONS

In order to successfully implement service-oriented architectures, service maps are seen as instruments to visualize relationships and interdependencies of services. They can be used as a common basis for business and IT departments to structure services in companies or networks. Structuring these services by using an appropriate service map for the specific context can effectively support agility and flexibility of business processes. However no attempt has been made so far to provide a framework for a structured service map design. Therefore this paper developed a framework to design structure and content of service maps (see s*ection 3*). It incorporates four building blocks for the design of a service map and exemplifies relevant contingencies such as SOA goal, focus, company size and SOA maturity. A case study of a Swiss bank demonstrated the applicability of the framework (see s*ection 4*). Based on different criteria the bank was able to deduce a service map according to the proposed framework. The bank's process architecture is thus reflected in the service map. Nevertheless the case study showed the application from a business-oriented perspective and focused on an internal use of a service map due to the banks characteristics and goals. Thus, further case studies have to be conducted in order to underpin the usability of the proposed framework also for different settings. Such a multi case study has not to be limited to the banking industry. Furthermore the proposed content according to the framework could only be partially considered in the banks final service map. Reasons are the limitations of the subjacent project and contrary interests of different stakeholders. Hence the consideration of visibility (Klose et al. 2007) by applying the framework could improve the applicability. Subsequently, a next step is the integration of the framework into more comprehensive methods for service map integration and design.

REFERENCES

Aier, S., et al. (2008) Economic Justification of Service-Oriented Architecture: Research Study: Experiences and Guidelines on Building SOA Business Cases, St. Gallen.

Arsanjani, A., et al. (2008) SOMA: A method for developing service-oriented solutions, IBM Systems Journal, 47 (3), 377-396.

Arsanjani, A. and Holley, K. (2006) The Service Integration Maturity Model: Achieving Flexibility in the Transformation to SOA, IEEE International Conference on Services Computing (SCC'06) Chicaco, 18-22 September, 515.

Bachman, J. (2005) SOA Maturity Model, Progress Sonic Corporation Homepage.

Baskerville, R., et al. (2005) Extensible Architectures: The Strategic Value Of Service-Oriented Architecture In Banking, Proceedings of the 13th European Conference on In-formation Systems (ECIS'05) Regensburg, May 26-28, 761-772.

Becker, A., et al. (2009) Value Potential and Challenges of Service-Oriented Architectures, 17th European Conference on Information Systems (ECIS) Verona, 8 - 10 June, 2-13.

Becker, J., et al. (2004) Process Management. A Guide for the Design of Business Processes, Springer, Berlin.

Bell, M. (2008) Service-Oriented Modeling: Service Analysis, Design, and Architecture, Wiley, Hoboken.

Bills, S. (2009) Standards Unit Finds an SOA Ally, American Banker, 174 (F306), 6-6.

Bonati, B., et al. (2006) Enterprise Services Architecture for Financial Services, Galileo Press, Boston.

Börner, R. and Goeken, M. (2009) Identification of Business Services Literature Review and Lessons Learned, Proceedings of the Fifteenth Americas Conference on Information Systems (AMCIS'09) San Francisco, August 6th-9th, 1-9.

Börner, R., et al. (2009) Towards an Operationalisation of Governance and Strategy for Service Identification and Design, 13th IEEE International EDOC Conference Auck-land.

Brooksbank, R. (1991) Defining the small business: a new classification of company size, Entrepreneurship and Regional Development, 3 (1), 17-32.

Brown, W. A., et al. (2006) SOA Governance – IBMs Approach, Somers, NY.

Ciganek, A. P., et al. (2006) Horizontal and Vertical Factors Influencing the Adoption of Web Services, Proceedings of the 39th Hawaii International Conference on System Sciences (HICSS'06) Kauai, Hawaii, 04-07 January 109c.

Donaldson, L. (2001) The Contingency Theory of Organizations, Sage Publications, Thousand Oaks, CA.

Dubé, L. and Paré, G. (2003) Rigor In Information Systems Positivist Case Research: Current Practices, Trends, And Recommendations, MIS Quarterly, 27 (4), 597-635.

Erl, T. (2007) SOA - Principles of Service Design, Prentice Hall, Boston.

Galliers, R. D. (1992) Choosing information systems research approaches, In: Information systems research: issues, methods, and practical guidelines (Ed. Galliers, R. D.) Blackwell Scientific Publications, Oxford, 144-162.

Hevner, A. R., et al. (2004) Design Science In Information Systems Research, MIS Quarterly, 28 (1), 75-105.

Homann, U., et al. (2004) Flexible Value Structures in Banking, Communications of the ACM, 47 (5), 34-36.

Janssen, M. and Joha, A. (2006) Motives for Establishing Shared Service Centers in Public Administrations, International Journal of Information Management, 26 (2), 102–115.

Jiang, M. and Willey, A. (2005) Service-Oriented Architecture for Deploying and Integrating Enterprise Applications, Proceedings of the 5th Working IEEE / IFIP Conference on Software Architecture (WICSA'05) Pittsburgh, 6-10 November, 272-273.

Klose, K., et al. (2007) Identification Of Services - A Stakeholder-Based Approach To SOA Development And Its Application In The Area Of Production Planning, ECIS '07: Proceedings of the 15th European Conference on Information Systems (Eds. Österle, H., et al.) University of St. Gallen, St. Gallen, June 07-09, 1802-1814.

Kohlborn, T., et al. (2009) Identification and Analysis of Business and Software Services - A Consolidated Approach,, IEEE Transactions on Services Computing, 2 (1), 50-64.

Kohlmann, F. and Alt, R. (2009a) Aligning Service Maps - A Methodological Approach from the Financial Industry, Proceedings of the 42nd Hawaii International Conference on System Sciences Hawaii, January 5-8, 1-10.

Kohlmann, F. and Alt, R. (2009b) Deducing Service Ownerships in Financial Networks, Proceedings of the 15th Americas Conference on Information Systems (AM-CIS'09) San Francisco, 6-9 August.

Kullvén, H. and Mattsson, J. (1994) A Management Control Model Based on the Customer Service Process, International Journal of Service Industry Manage-ment, 5 (3), 14-25.

Lee, A. S. (1989) A Scientific Methodology For MIS Case Studies, MIS Quarterly, 13 (1), 33-50.

Legner, C. and Heutschi, R. (2007) SOA Adoption in Practice - Findings from Early SOA Implementations, European Conference on Information Systems ECIS'07 (Eds. Österle, H., et al.) St. Gallen, 1643-1654.

Nadham, E. G. (2004) Seven Steps to a Service-Oriented Evolution, Business Integration Journal, 41-44.

OASIS (2006) Reference Model For Service Oriented Architecture 1.0.

Papazoglou, M. P. and Georgakopoulos, D. (2003) Service-oriented Computing, Communications of the ACM, 46 (10), 25-28.

Papazoglou, M. P., et al. (2006) Service-Oriented Computing Research Roadmap, Dagstuhl Seminar 05462.

Rosen, M., et al. (2008) Applied SOA: Service-Oriented Architecture and Design Strategies, Wiley Publishing Inc., Indianapolis.

Schelp, J. and Aier, S. (2009) SOA and EA—Sustainable Contributions for Increasing Corporate Agility, Proceedings of the 42nd Hawaii International Conference on System Sciences: HICSS'42 (Ed. Sprague, R. H.) IEEE Computer Society (Los Alamitos) Waikoloa, January 5-8, 1-8.

Sedera, D. (2008) Does Size Matter? Enterprise System Performance in Small, Medium and Large Organizations, Proceedings of the 2nd Workshop on 3rd Generation Enterpri-se Resource Planning Systems Copenhagen, 17-18 November.

Simon, H. A. (1996) The Sciences of the Artificial, MIT Press, Cambridge, MA London.

Terlouw, J., et al. (2009) An Assessment Method for Selecting an SOA Delivery Strategy: Determining Influencing Factors and Their Value Weights, 4th International Work-shop on Business/IT Alignment & Interoperability (BUSITAL'09) Ams-terdam, 09 June.

Vitvar, T., et al. (2007) Semantic SOA to Promote Integration of Heterogeneous B2B Services, CEC-EEE '07: 9th IEEE International Conference on E-Commerce Technology and 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services 451-456.

Wigand, R., et al. (1997) Information, Organization and Management: Expanding Markets and Corporate Boundaries, Wiley, New York et al.

Zimmermann, O., et al. (2004) Second Generation Web Services-Oriented Architecture in Production in the Finance Industry, 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2004) ACM, Vancouver, Canada, 283-289.

4.3 Artikel 5: Operationalisierung der IT-Governance-Kernbereiche für die Identifizierung und Gestaltung von Services

Veröffentlicht: Alter, S., Börner, R., Goeken, M. (2009): Operationalisierung der IT-Governance-Kernbereiche für die Identifizierung und Gestaltung von Services, in: Lecture Notes in Informatics (LNI), Lübeck, Deutschland, Springer, S. 3660-3674.

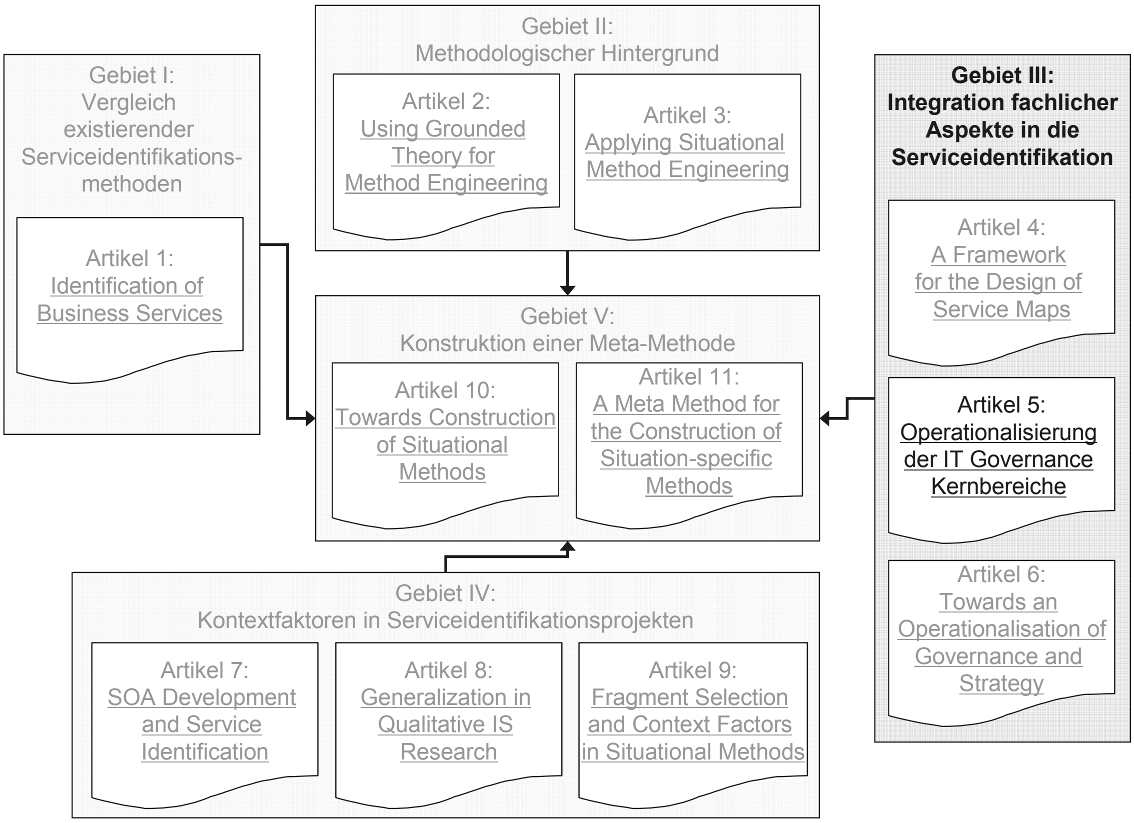

Abbildung 10: Artikel 5 im Kontext der kumulativen Dissertation

Operationalisierung der IT-Governance-Kernbereiche für die Identifizierung und Gestaltung von Services

Stefanie Alter, René Börner, Matthias Goeken

Management Research Centre
Frankfurt School of Finance and Management
Sonnemannstraße 9-11
60314 Frankfurt
[s.alter; r.boerner; m.goeken]@frankfurt-school.de

Abstract

Serviceorientierung ist ein seit einigen Jahren viel diskutiertes Paradigma für Unternehmensarchitekturen. Ein wichtiger Erfolgsfaktor für die Implementierung einer serviceorientierten Architektur (SOA) ist die Berücksichtigung der fachlichen Perspektive basierend auf Geschäftsprozessen [DD07]. Zu dieser fachlichen Sicht auf eine SOA gehören auch Governance-Aspekte, weshalb analog zur IT-Governance der noch recht unpräzise Begriff SOA-Governance geprägt wurde. Ein Bestandteil der SOA-Governance ist, nach Ansicht der Verfasser, die Service-Governance, d.h. die fachliche Sicht auf den einzelnen Service. Notwendig sind demnach Verfahren, die eine Berücksichtigung der Governance-Perspektive bereits zu einem frühen Zeitpunkt im Service-Lebenszyklus ermöglichen. Ziel dieses Beitrages ist daher die Entwicklung eines Fragebogens zur methodischen Unterstützung der Identifikation und Gestaltung von Services. Hierfür wurden aus den fünf Kernbereichen der IT-Governance, den sogenannten IT Governance Focus Areas, Dimensionen der Service-Governance abgeleitet und für eine Verwendung während der Service-Identifikation operationalisiert.

1 Governance in serviceorientierten Architekturen

Nur Unternehmen mit der Flexibilität, sich schnell an ein neues Marktumfeld anzupassen, können langfristig überleben [BKR04]. In diesem Zusammenhang ist die Serviceorientierung ein viel genanntes Paradigma für die Unternehmensarchitektur. Die in Verbindung mit Serviceorientierung am häufigsten genannten Aspekte sind eine schnellere Anpassung an Veränderungen der Geschäftsprozesse durch eine größere Flexibilität und Agilität, eine höhere Wiederverwendbarkeit von Services und die lose Kopplung verbunden mit einer hohen inneren Kohäsion der Services [Pa03]. Die Wiederverwendbarkeit von Services verhindert das unnötige Vorhalten redundanter Funktionen und senkt somit Entwicklungs- und Unterhaltungskosten der IT-Infrastruktur. Die lose Kopplung ermöglicht eine nahezu virtuose Komposition von Services und eröffnet Sourcing-Potentiale, bspw. durch die Nutzung von Web-Services [BBW09].

Die Implementierung serviceorientierter Architekturen (SOA) bringt aber auch eine Reihe von Herausforderungen mit sich. Die flexible Orchestrierung der Services erhöht die Komplexität des gesamten Systems deutlich und kann sogar zu einer Verschlechterung der Performance führen [Ra06]. Eine SOA-Governance, d.h. ein holistisches Management der Technologie sowie der Geschäftsprozesse ist daher notwendig [Be06]. Ein Ziel dieser SOA-Governance ist es, „to check services concerning capability, security and strategic business alignment” [Ni08]. Neben der Bewältigung der erhöhten Komplexität entstehen auf Ebene der einzelnen Services auch SOA-spezifische Gefahrenpotentiale [o.V.08].

In der Literatur finden sich bereits eine ganze Reihe von Beiträgen, die das Thema SOA-Governance behandeln und entsprechende Vorgehensweisen vorschlagen [MB06; SS07; BBW09; Bi05a; Lo08; KSH08; JG07]. [Ni08] entwickeln nach Analyse einiger der genannten Quellen ein generisches Governance-Modell. Alle genannten Ansätze adressieren dabei die Governance für eine gesamte serviceorientierte Architektur. Aspekte wie der Lebenszyklus einer SOA, die Integration neuer Services sowie Pflege und Entsorgung bestehender Services sind bei dieser Betrachtung von Bedeutung. Ein wichtiger Bestandteil des SOA-Lebenszyklus ist die Identifizierung und Gestaltung von Services, die stets am Beginn steht. Obwohl dieser Teil eine entscheidende Rolle für den Erfolg einer SOA spielt, bleibt die Auswirkung von Governance-Aspekten auf die Gestaltung einzelner Services bei den betrachteten Ansätzen ungeklärt [BG09].

In der Diskussion über Gestaltungsprinzipien von Services werden Aspekte wie Schnittstellenorientierung, Interoperabilität, lose Kopplung, Modularität und Wiederverwendbarkeit von vielen Autoren als entscheidende Merkmale einer SOA dargestellt [Er04; Jo08; Bi05b; LH07]. Zweifelsohne sind diese Merkmale kennzeichnend für eine SOA, allerdings spiegeln sie ausschließlich eine technische Betrachtungsweise einer solchen Architektur wider. Die fachliche Perspektive (basierend auf Geschäftsprozessen) ist jedoch ein wichtiger Erfolgsfaktor für die Implementierung einer SOA [DD07]. Dazu gehören auch Governance-Aspekte, die nach Ansicht der Verfasser bereits bei der Identifizierung und Gestaltung von Services beachtet werden müssen. Die Berücksichtigung von Governance-Merkmalen im Lebenszyklus eines einzelnen Services wird im Folgenden als „Service-Governance“ bezeichnet (siehe Abbildung 1). Letztere bedient sich dazu einiger Elemente der IT- und SOA-Governance, stellt aber keine direkte Ableitung dar und erhebt daher auch nicht den Anspruch auf vollständige Abdeckung aller Aspekte erstgenannter Governance-Arten. Der bisher häufig verfolgten, technischen Bottom-up-Herangehensweise bei der Identifizierung von Services [Na04] muss ein Top-down-Ansatz vorangestellt werden, um eine wirtschaftliche und unternehmensstrategisch adäquate Service-Identifikation sicherzustellen. Die im Folgenden näher betrachtete Identifikation ist der erste Schritt einer auch von anderen Autoren geforderten, hybriden Vorgehensweise zur Identifikation von Services [IS05; Za05].

Kapitel 2 gibt einen Literaturüberblick und erläutert vorhandene Methoden zur Service-Identifikation und zur Unterstützung der SOA-Governance. In Kapitel 3 werden aus den Kernbereichen der IT-Governance, den sogenannten IT Governance Focus Areas, *Dimensionen* einer Service-Governance abgeleitet. In Kapitel 4 wird anhand dieser Dimensionen ein Fragebogen entwickelt, der die Berücksichtigung von Governance-Aspekte bereits während der Service-Identifikation und -Gestaltung unterstützen soll. Dieser Fragebogen dient der methodischen Unterstützung der Service-Identifikation und ist ein erster Schritt hin zu einer Methode zur fachlichen Identifikation von Services. Kapitel 5 zieht ein Fazit und erörtert Möglichkeiten weiterer Forschung.

2 Bisherige Ansätze zur Service-Identifikation und -Gestaltung

In der Literatur werden bereits einige Ansätze sowohl zur Identifikation als auch zur Gestaltung von Services diskutiert. Einige von ihnen haben ihre Wurzeln im stark ingenieurtechnisch geprägten Bereich der Fertigungsindustrie [KKB07], andere betrachten die Modularisierung im IT-Dienstleistungsbereich [BK05]. Diese Ansätze sind oft primär technisch geprägt. Auch unter den Ansätzen, die sich mit Finanzdienstleistungen beschäftigen, gibt es neben deutlich fachlich orientierten Varianten [Ar08; KA07] auch stark technische (objektorientierte) Ansätze [Wi07].

Klose et al. [KKB07] betonen die große Bedeutung des „business point of view“. Dementsprechend sind das Business Process Modeling (BPM) und die daraus resultierenden Geschäftsprozesse Grundlage ihrer Analyse. Die von ihnen betrachteten Aspekte wie beispielsweise IT-Unterstützung der Prozesse, beteiligte Organisationseinheiten, interne und externe Stakeholder sowie die Einbeziehung mehrerer Ebenen (von der Geschäftsprozesssicht bis zur Aktivitätensicht) sind von entscheidender Bedeutung für eine Identifikation von Services. Die Betrachtung der Interaktions- und Sichtbarkeitslinie, die ursprünglich aus der Marketing-Literatur stammt [Sh81], geschieht auch unter fachlichen Gesichtspunkten. Sie ist weniger für die Identifikation selbst, als vielmehr für den späteren Umgang mit den einzelnen Services entscheidend. Beide Merkmale sind für Sourcing-Entscheidungen bezüglich der Services von großer Bedeutung. Die Autoren betrachten zwar Services eines Fertigungsbetriebes, jedoch ist die Berücksichtigung der Interaktions- und Sichtbarkeitslinie für Dienstleistungsprozesse nicht minder bedeutungsvoll. Eine Stärke des Ansatzes von [KKB07] ist, dass nach der bisherigen Top-down-Vorgehensweise auch die Umsetzbarkeit der identifizierten Service auf Basis technischer Kriterien (also „Bottom-up“) geprüft wird. Sie verfolgen demnach einen hybriden Ansatz, wie er beispielsweise von [IS05] und [Za05] vorgeschlagen wird. Diese Machbarkeitsprüfung ist im Rahmen der Gestaltung von Services für eine betriebswirtschaftlich sinnvolle Implementierung der gleichen unverzichtbar. Die Notwendigkeit von IT-Governance ist in diesem Ansatz nur implizit beschrieben. Eine SOA-Governance im Speziellen fehlt vollständig. Verbunden damit werden auch bei der Identifikation von Services keinerlei Governance-Aspekte beachtet.

Ein herausragendes Merkmal bei **Böhmann und Krcmar [BK05]** ist die ausgeprägte Zielbestimmung, deren Ergebnis allerdings den allgemeinen SOA-Zielen gleicht [Pa03]. Aus ihren Ausführungen wird nicht deutlich, inwieweit diese Zielbestimmung der Service-Identifikation nützt. Es gibt keinen expliziten Bezug zu BPM. Die Autoren erwähnen lediglich, dass vorhandene Informationen wie beispielsweise Prozessmodelle für die weitere Vorgehensweise bereitzustellen sind. Tatsächlich erfolgt später eine Dokumentation der Serviceprozesse, die zumindest implizit auf die Bedeutung von Geschäftsprozessen hinweist. Mit der Anwendung ihrer Modularisierungsmatrix verfolgen [BK05] eine hybride Herangehensweise, die allerdings deutlich mehr durch die vorhandene Technik geprägt wird als der zuvor diskutierte Ansatz von Klose et al. Mit Hilfe der Matrix soll eine komplette Architektur entworfen werden, ohne dass die Identifikation der einzelnen Serviceprozesse näher erläutert wird. Insgesamt liegt der Schwerpunkt auf der Modularisierung von IT-Dienstleistungen, so dass die hier beschriebenen Services sehr komplex sind. Auch bei [BK05] fehlt eine explizite Behandlung von IT- und SOA-Governance-Aspekten. Insofern lassen sich nur wenige Aspekte für weitere Überlegungen im Rahmen der Service-Identifikation nutzen. Einer davon ist sicherlich die große Bedeutung, die die Autoren der Nachfragerintegration zukommen lassen, denn die Interaktion mit Kunden hat erheblichen Einfluss auf verschiedene Governance-Aspekte eines Services.

Obwohl **Winkler [Wi07]** ausgehend von Geschäftsprozessen einen Top-down-Ansatz verfolgt, weist sie nicht ausdrücklich auf die Bedeutung des BPM hin. Implizit werden dennoch modellierte Prozesse aus dem Finanzdienstleistungsbereich für die Identifikation der Services verwendet. Die Autorin verzichtet im Gegensatz zu den vorher beschriebenen Ansätzen auf einen Bottom-up-Ansatz bzw. eine hybride Herangehensweise. Umso erstaunlicher ist es, dass die von ihr identifizierten Services mit Abstand am granularsten sind. Nach Ansicht des Verfassers handelt es sich bei der von Winkler beispielhaft angeführten Nullstellenberechnung nicht um einen fachlichen Service im Rahmen einer SOA, sondern vielmehr um ein Objekt im Sinne der objektorientierten Programmierung [Za05; El07]. Die Anforderungen sind sehr allgemein gehalten und beziehen sich eher auf technische Merkmale denn auf fachliche oder betriebswirtschaftliche Anforderungen. Die Erfüllung ihrer ersten Anforderung – der übergreifenden Einsetzbarkeit des Services in einer Vielzahl heterogener Anwendungen – macht [Wi07] nicht von dem (fachlich) identifizierten Service selbst, sondern von seiner (eher technisch getriebenen) Gestaltung abhängig. [Wi07] betrachtet weder IT- noch SOA-Governance bei ihrer Methode zur Service-Identifikation.

Arsanjani et al. [Ar08] verfolgen einen hybriden Ansatz und demonstrieren die beispielhafte Anwendung ihrer Methode an einem Finanzdienstleistungsunternehmen. Bei ihrem Goal Service Modelling und der darauf folgenden Zerlegung handelt es sich ebenfalls um eine hierarchische Dekomposition der Geschäftsprozesse. Es ist also zunächst ein klassischer Top-down-Ansatz, der mit den zuvor präsentierten Verfahren vergleichbar ist. Spätestens im Rahmen der Spezifikation bewegt sich der Fokus aber hin zur technischen Umsetzung und bezieht die bereits bestehende Infrastruktur mit ein (Bottom-up). Zwar erinnert die Aufteilung in Identifikations- und Spezifikationsphase stark an Winklers Vorgehensweise, jedoch wird der Begriff „Service“ hier stets mit Blick auf Geschäftsprozesse, d.h. deutlich grobgranularer verwendet. Die Autoren verwenden die Begriffe „Service“ und „Web Service“ synonym, ohne explizit darauf einzugehen. Im Rahmen des Business Process Modeling führen [Ar08] eine „Rules and Policies Analysis“ durch. Diese internen Regeln und Vorschriften sind Bestandteil der IT-Governance. Die Autoren stellen auch ein fraktales Lebenszyklusmodell serviceorientierter Architekturen vor, zu dem auch die Service-Identifikation zählt. Nichtsdestotrotz findet eine Bezugnahme auf Governance-Aspekte während der Service-Identifikation nicht statt.

Die von **Kohlmann und Alt [KA07]** vorgeschlagene Vorgehensweise gehört ebenfalls zu den hybriden Varianten der Service-Identifikation. Zusätzlich zur Business-Sicht mittels Business Process Modeling wird eine Asset Analysis für das Clustering der Services verwendet. Die Autoren unterscheiden explizit drei Service-Granularitäten und ordnen diese verschiedenen Hierarchieebenen zu. Tendenziell unterstützen ihre Services ganze Geschäftsprozesse. Im Rahmen der IT-Governance betonen [KA07] die Bedeutung der konsistenten Benennung von Services. Es findet eine Unterscheidung zwischen ausschließlich intern genutzten Services und extern bereitgestellten Services statt. Somit werden Sourcing-Optionen explizit berücksichtigt. Die Einbeziehung der funktionalen und semantischen Ähnlichkeit beim Clustering dient der betriebswirtschaftlichen Betrachtung beim Zuschnitt der Services. Neben der bereits erwähnten internen Policies finden auch gesetzliche Bestimmungen bei der Ausgestaltung der Services Berücksichtigung. [KA07] thematisieren den Umgang mit Kundendaten, die unter Umständen gesetzlich geschützt sind und daher spezieller Behandlung bedürfen. Neben gesetzlichen Regelungen spielen bei ihnen auch IT-Governance-Aspekte eine Rolle. Vor allem bei der Benennung der Services wird die Bedeutung der Governance deutlich. Bei der Gestaltung der Services werden allerdings keine dieser Aspekte berücksichtigt.

Kohlborn et al. [KKCR09] unterscheiden in ihrem Ansatz Business Services und Software Services. Diese Zweiteilung greift im Grunde die auch in anderen Ansätzen vorhanden zweistufige Hierarchie von Services auf. Allerdings gelingt es den Autoren durch diese explizite Trennung deutlich besser, zunächst Business Services mit einem klaren Fokus auf Geschäftsprozesse und Berücksichtigung der Unternehmensstrategie in drei Phasen zu identifizieren. Erst anschließend werden in entsprechenden drei Phasen die unterstützenden Software Services identifiziert und durch eine „Verbindungsphase“ mit den Business Services zusammengeführt. Dieser Ansatz berücksichtigt und beseitigt eine Reihe von Schwächen anderer Herangehensweisen, bezieht aber auch ausdrücklich Ideen aus diesen mit ein. So nutzen die Autoren eine Analysetechnik, die alle Stakeholder und deren Integration berücksichtigt und lehnen sich dabei eng an [KKB07] an. Eine Berücksichtigung von Governance-Aspekten findet allerdings auch in dieser Methode nicht statt.

Schelp und Stutz [SS07] sehen den Aufbau einer ganzheitlichen SOA-Governance als unumgänglich um der Steigerung der Komplexität Rechnung zu tragen. Sie stellen fest, dass ohne geeignete Governance-Instrumente, die aus einer SOA resultierende Komplexität „zu Strukturen führt, deren Wartung und Pflege ähnlich aufwendig wird wie die der bestehenden evolutionären Applikationslandschaften, die sie ablösen sollen“ [SS07, S.66]. Das von [SS07] entwickelte SOA-Governance-Modell enthält Aktivitäten für eine erfolgreiche SOA-Umsetzung, etwa Planung, Implementierung, Wartung oder Controlling. Sie benennen unter anderem die Aktivitäten SOA Service Design und SOA Service Build. Dort werden zunächst Designrichtlinien festgelegt, bevor der Service gemäß den bestehenden Entwicklungsparadigmen umgesetzt wird. Jedoch ist Governance in ihrem Verständnis auf einer höheren Ebene angesiedelt. Die Umsetzung von Governance für Services wird nicht thematisiert.

Johannsen und Goeken [JG07] betonen, dass für eine SOA nahezu die gleichen Anforderungen und Herausforderungen gelten, wie sie in der IT-Governance, insbesondere im Umfeld des COBIT-Referenzmodells, mit dem Begriff „IT Governance Focus Areas“ beschrieben werden. Sie erläutern kurz die fünf Focus Areas aus SOA-Sicht, bspw. für den Bereich „Value Delivery“: „Die Erwartung an die jeweiligen Wertbeträge der Servicekomponenten unter Berücksichtigung unterschiedlicher Sourcing-Optionen ist kontinuierlich zu überprüfen.“ [JG07, S.193]. Für eine SOA-Governance zeichnen sich laut [JG07] die zwei zentralen Aufgabenbereiche SOA-Conformance und SOA-Lifecycle-Governance ab. Der Bereich SOA-Conformance beinhaltet Aufgaben die zeitlich betrachtet vor der Identifikation und Gestaltung von Services liegen, etwa inwieweit ein Unternehmen organisatorisch, prozessual und technisch auf die Umstellung zur Serviceorientierung vorbereitet ist. SOA-Lifecycle-Governance beschäftigt sich mit der Gewährleistung von Governance im laufenden Betrieb. Hierfür sind die Service-Komponenten während des SOA-Lebenszyklus gemäß den IT Governance Focus Areas (siehe Kapitel 3) einer Überprüfung zu unterziehen. Eine konkrete Ausgestaltung dieser Überprüfung wird jedoch nicht vorgenommen.

[JG07] thematisieren den Zusammenhang zwischen IT-Governance und der Governance einer serviceorientierten Architektur. Eine konkrete Operationalisierung für die Umsetzung von Governance für SOA sowie eine explizite Verbindung zu einer Service-Governance fehlt jedoch in den genannten Ansätzen [BG09]. An dieser Stelle setzt der vorliegende Beitrag an und unterscheidet hierfür zwischen SOA-Governance und Service-Governance. Der Scope dieses Beitrags ist in Abbildung 1 hervorgehoben.

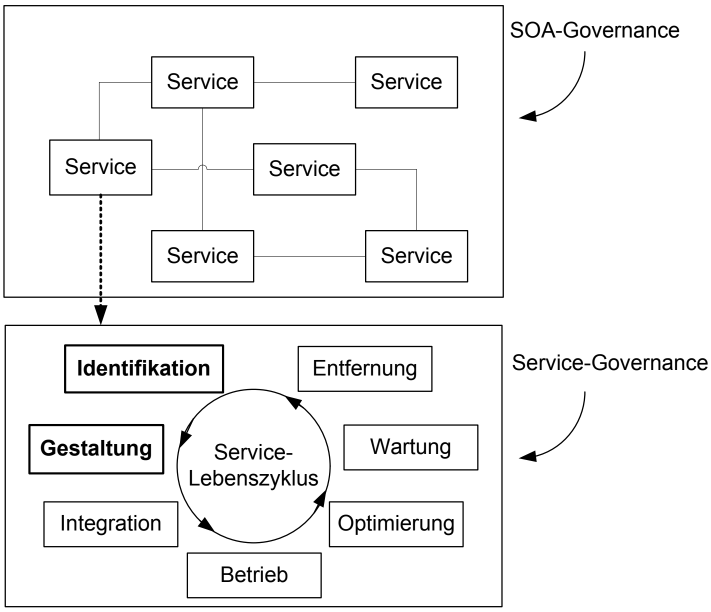

Abbildung 1: Vergleich der Bestandteile von SOA- und Service-Governance

Wie in Abbildung 1 gezeigt bezieht sich die SOA-Governance auf die serviceorientierte Architektur als Ganzes. Dies beinhaltet nach [JG07] die SOA-Conformance und die SOA-Lifecycle-Governance. Letztgenannte ist jedoch nicht zu verwechseln mit der hier adressierten Service-Governance, die sich mit dem Lebenszyklus (Identifikation, Gestaltung,…, Entfernung) jedes einzelnen Services beschäftigt. Aufgrund der in der Abbildung dargestellten Beziehung zwischen SOA- und Service-Governance ist die Service-Governance ist nach Ansicht der Verfasser unumgänglich für die Umsetzung einer SOA-Governance. Die Adressierung betriebswirtschaftlicher sowie fachlicher Aspekte bereits während der Identifikation und Gestaltung von Services sind ein wichtiger Erfolgsfaktor für die strategieorientierte Steuerung der gesamten IT-Systemlandschaft.

Dem Ansatz von [JG07] folgend wird daher im nächsten Abschnitt die Umsetzung von Service-Governance während der Identifikation und Gestaltung von Services beschrieben. Hierfür werden die fünf IT Governance Focus Areas für eine Anwendung auf Services operationalisiert. Aus den fünf Kernbereichen der IT-Governance werden Dimensionen der Service-Governance abgeleitet. Diese Dimensionen werden dann, wie in Kapitel 4 erläutert, mit konkreten Merkmalen sowie deren Ausprägungen versehen und bilden die Grundlage eines konkreten Fragebogens zur methodischen Unterstützung der Service-Identifikation.

3 IT Governance Focus Areas und ihre Anwendung auf Services

Die Kernbereiche der IT-Governance sind die fünf sogenannten IT Governance Focus Areas: Strategic Alignment, Value Delivery, Risk Management, Resource Management und Performance Measurement [JG07; ITGI07]. Diese Kernbereiche adressieren Aspekte, die die Aufmerksamkeit des Managements erfordern, um die IT aus Geschäftssicht adäquat zu steuern. Im Folgenden werden diese fünf Kernbereiche einzeln erläutert und ihre Verwendung für Services und im Rahmen einer SOA diskutiert. Ziel dieses Abschnitts ist es, Dimensionen der Service-Governance aus den etablierten Kernbereichen nach COBIT abzuleiten.

Strategic Alignment

Der Abgleich zwischen Geschäftsseite und IT ist ein vielschichtiges Problem, welches neben der Strategie beider Bereiche auch Prozesse, Architekturen, Infrastrukturen und kulturelle Aspekte umfasst. Häufig wird davon ausgegangen, dass Alignment auf der Ebene der Strategie beginnt [Lu08]. Zunächst sollte ein Abgleich zwischen Geschäfts- und IT-Strategie vorgenommen werden. Das Strategic Alignment Modell (SAM) von [HV93] unterteilt ein Unternehmen hierfür zunächst in vier Domänen. Unterschieden werden Business und IT sowie eine strategische (externe) und eine infrastrukturelle (interne) Sichtweise. Nach [HV93] ist Alignment “a balance among the choices made across all four domains”. [GJP09] beschreiben darauf aufbauend einen Zusammenhang zwischen Standard-Geschäfts- und Standard-IT-Prozessen [vgl. auch AG09]. Sie verdeutlichen damit, dass ein Abgleich zwischen Business und IT auch auf den dem Strategie-Abgleich nachgelagerten Ebenen notwendig ist. [RB00] nennen ein gemeinsames Verständnis von Weg und Ziel als einen zentralen Erfolgsfaktor für diesen Abgleich zwischen Business und IT. Schlüssel zu diesem gemeinsamen Verständnis ist die gezielte Kommunikation zwischen IT- und Business-Mitarbeitern. Dies fördert auch die von [BK05] geforderte Nachfragerintegration, denn die Fachabteilung kann natürlich als interner Kunde interpretiert werden. Auch [Si07] betont die Fokussierung auf die Geschäftsseite, um die Finanzierung von Entwicklung und Pflege sicherzustellen und den Service unternehmensweit einsetzen zu können. Eine konsistente Benennung der Services, wie sie von [KA07] gefordert wird, ist nicht nur eine wesentliche Voraussetzung für eine hohe Wiederverwendbarkeit, sondern reflektiert auch das Alignment von Business und IT. Diesen Ansätzen folgend sollte das Strategic Alignment durch ein Alignment auf nachgelagerten Ebenen forciert werden. Bei der Service-Identifikation sollte dies nach Meinung der Verfasser durch eine geeignete Zusammensetzung des

„Identifikationsteams“ organisatorisch unterstützt werden. Eine dafür zu schaffende Service Design Unit (SDU) setzt sich daher aus paritätisch aus Fachanwendern und IT-Mitarbeitern zusammen (siehe Abbildung 2).

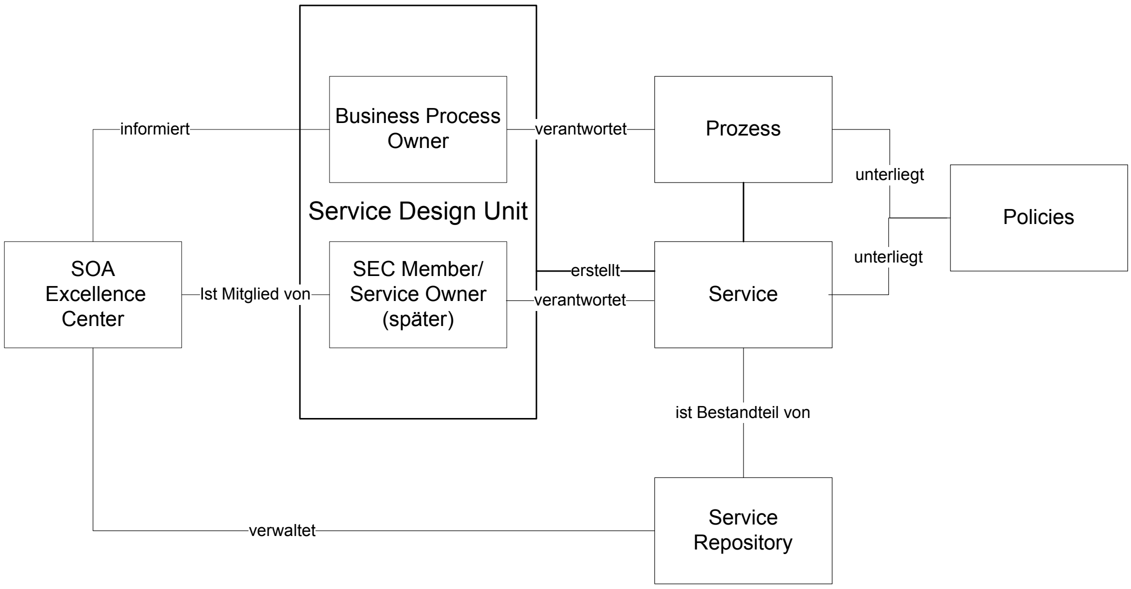

Abbildung 2: Einordnung der Service Design Unit

Ein Mitglied dieses Teams ist der Business Process Owner (BPO), der einen potentiellen Service-Kandidaten der SOA-Einheit des Unternehmens (hier: SOA-Excellence-Center SEC) meldet. Zusammen mit einem Mitglied des SEC (dem späteren Service Owner) bildet der BPO die SDU. Dadurch, dass der BPO den Prozess der Service-Identifikation in Gang setzt, sollte die fachliche Eignung des Service-Kandidaten zumindest grundsätzlich gesichert sein. Durch die gemeinsame, detaillierte Gestaltung dieses Service-Kandidaten zu einem Service, werden technische Restriktionen und vor allem das Kriterium „Strategische Bedeutung“ berücksichtigt. Somit wird ein Strategic Alignment durch ein Alignment auf Ebene der Services unterstützt.

Value Delivery

Dieser Kernbereich beinhaltet die Forderung nach einem deutlichen und nachweisbaren Wertbeitrag der IT zum Unternehmenserfolg. Eine wesentliche Aufgabe ist dabei die bedarfsgerechte Gestaltung von IT-Produkten im Sinne von wertorientierten IT-Services. Als wesentlich für die bedarfsgerechte Bereitstellung wertorientierter IT-Services stellt sich die Abstimmung zwischen dem Abnehmer und dem Lieferanten von IT-Services dar. Diese Abstimmung ist zugleich eine zentrale Aufgabe der bereits beschriebenen Focus Area Strategic Alignment. [TK03] stellen einen Zusammenhang zwischen dem Grad des strategischen Alignments und dem Wertbeitrag der IT her. Weiterhin ergibt eine Studie von [BBW09], dass eine SOA das Business/IT-Alignment stärkt. Dies gelte jedoch nur, wenn Business-IT-Alignment darüber definiert wird, „dass die IT-Lösungen besser zum Geschäft passen“ [BBW09]. [Gr04] hält die Generierung eines Wertbeitrages durch Alignment von Business und IT für die wichtigste Aufgabe der IT-Governance.

Die wechselseitige Beziehung zwischen Value Delivery und Strategic Alignment wird auch auf Service-Ebene deutlich. Dimensionen wie Änderungshäufigkeit oder Wiederverwendbarkeit können nur sinnvoll ermittelt oder geschätzt werden, wenn Business- und IT-Erfahrungswissen zusammengefasst werden. Wiederverwendbarkeit der Services und die damit realisierbaren ökonomischen Potentiale, insbesondere die Eliminierung von Redundanzen [KA07; Si07], sind ein entscheidender Erfolgsfaktor im Bereich Value Delivery. Eine wesentliche Voraussetzung für eine hohe Wiederverwendbarkeit ist die einheitliche Benennung von Services. Wie bereits beschrieben ist die von [BK05] explizit geforderte Nachfragerintegration oftmals entscheidend, um einen Nutzen für den Kunden zu schaffen. Die Gestaltung der Kundenschnittstelle (auch „Line of interaction“ [Sh81]) muss beim Design der Services beachtet werden. Je mehr ein Kunde eingebunden ist, umso häufiger sind Änderungen bezüglich der Schnittstellen und Funktionalitäten des Services zu erwarten, da sich Kundenanforderungen u.U. häufig ändern können.

Risk Management

Ziel des Risikomanagements ist die Identifikation und Analyse von Risiken. Außerdem sollte ein Unternehmen ein klares Verständnis der eigenen Risikopräferenz haben. Dazu kommt die Kenntnis einschlägiger Regularien und Gesetze sowie die Verankerung von Verantwortlichkeiten in einer Organisation [vgl. JG07, S.42ff]. Im Folgenden sollen die wichtigsten Sicherheitsaspekte einer SOA [vgl. auch o.V. 08] diskutiert werden.

Auch für einen konkreten Services muss die Authentifizierung sichergestellt sein, d.h. Benutzer (oder andere Entitäten) müssen – bspw. anhand von Benutzername und Passwort – eindeutig identifiziert werden. Die Autorisierung stellt in der Regel durch die Verwendung von Rollenkonzepten sicher, dass nur Nutzer mit entsprechender Berechtigung Daten abfragen oder verändern dürfen. Sind Authentifizierung und Autorisierung entsprechend implementiert, kann bei adäquater technischer Umsetzung – durch nicht kompromittierbare Protokolle – auch die Integrität und Vertraulichkeit der in den Services verarbeiteten Daten gewährleistet werden. Die Verfügbarkeit von Services spielt einerseits für die Kundenzufriedenheit eine entscheidende Rolle. Die Konsequenzen hieraus werden in den Focus Areas „Resource Management“ und „Performance Measurement“ betrachtet. Andererseits fällt die Verfügbarkeit vor allem im Hinblick auf regulatorische Anforderungen in die Kategorie „Risk Management“. Insbesondere Finanzdienstleister müssen bspw. gemäß den Anforderungen der MARisk oder Basel II die Verfügbarkeit von Services sicherstellen.

Resource Management

Resource Management zielt auf die konsequente Steuerung der Ressourcen einer Organisation. Die Optimierung der Investitionen und deren zweckmäßiges Management stehen dabei im Mittelpunkt. Das IT-Governance-Referenzmodell COBIT (Control Objectives for Information and Related Technology) [ITGI07] beschreibt Anwendungen, Informationen, Infrastruktur und Personal als die wesentlichen IT-Ressourcen.

Für einen Service muss dazu festgestellt werden, welche Inputs in Form von Daten, Hardware und ggf. Personal er benötigt und ob diese Inputs stets gleich sind. Möglicherweise kann ein Service in Abhängigkeit von den Inputfaktoren auf unterschiedliche Art und Weise genutzt werden. Dies hat wiederum einen Einfluss auf den vom Service gelieferten Output, der ebenfalls in Hinblick auf alle genannten Faktoren definiert sein muss. Je genauer die Schnittstellen der Services definiert sind, desto exakter können benötigte Ressourcen bestimmt werden. Ein hoher Grad an Autonomie, d.h. eine möglichst große Kohäsion innerhalb des Services, verbunden mit einer losen Kopplung zu anderen Services, unterstützt somit das Resource Management. Die Ausdehnung eines Services beeinflusst ebenfalls das Resource Management. Hierbei wird betrachtet, ob ein Service geschäftsrollenübergreifend agiert. Ist dies der Fall, d.h. werden mehrere Rollen eingebunden, erhöht dies die Komplexität der Kapazitätsplanung. Hinzu kommt, dass ein solcher Service in der Regel für ein Outsourcing nicht in Frage kommt. Durch die Beteiligung mehrerer Rollen wäre nur die Auslagerung eines kompletten Geschäftsprozesses möglich, nicht jedoch eines einzelnen Services. Die notwendige Verfügbarkeit eines Services beeinflusst natürlich das Ausmaß der vorzuhaltenden Ressourcen. Für kritische Prozesse müssen dementsprechend Puffer vorgehalten werden, die auch in Notfällen einen reibungslosen Betrieb garantieren (vgl. Abschnitt „Risk Management“).

Performance Measurement

Das Performance Measurement verfolgt und überwacht die Umsetzung der Strategie, von Projekten, von Prozessen, die Verwendung von Ressourcen etc. Durch Operationalisierung und Messung von Maßnahmen und Aktivitäten wird die Zielerreichung geprüft und unterstützt. Dies geschieht beispielsweise mit einer Balanced Scorecard oder einer anderen Methode zur Übersetzung von Strategie in „messbare“ Einheiten. Die Messung geht hier über die Anforderung des Rechnungswesens hinaus, da auch sogenannte „weiche“ Faktoren in die Messung miteinbezogen werden [KN93]. COBIT empfiehlt, dass Ziele und deren Metriken auf drei Ebenen festgelegt werden sollten: IT-Ziele und Metriken, die definieren, was die Geschäftsbereiche von der IT erwarten. Prozessziele und Metriken, die definieren, was der IT-Prozess liefern muss, um die Ziele der IT zu unterstützen, sowie Aktivitätsziele und deren Metriken [vgl. AG09].

Die Messung der Performance ist grundsätzlich auch für Services möglich, allerdings müssen dabei einige Eigenheiten beachtet werden. Zunächst müssen geeignete Metriken bzw. KPIs für den jeweiligen Service identifiziert werden. Die Zeit für die Ausführung des Services oder auch die Reaktionszeit bei Aufruf können als KPI herangezogen werden. Letztere steht wieder in engem Zusammenhang mit der Verfügbarkeit des Services, denn die Reaktionszeit ist hier die entscheidende Größe. Alle qualitätskritischen Merkmale sollten bestimmt und gemessen werden, unabhängig davon ob der Service für interne oder externe Kunden erbracht wird. Im Falle eines Outsourcings ist die Formalisierung der Messung im Sinne eines Service Level Agreements unerlässlich. Die Messung der Service-Performance muss allerdings differenziert betrachtet werden, denn ein Service umfasst unter Umständen nicht nur eine Aktivität mit relativ kleinem Umfang. Abhängig von der Granularität kann er auch einen ganzen Geschäftsprozess oder zumindest mehrere Teile davon abdecken. Misst man nun Kennzahlen auf Aktivitäten-ebene muss bspw. geklärt werden, ob die Kennzahlen ohne weiteres aggregierbar sind. Die Gestaltung von Services muss also auch auf die Messbarkeit der Performance Rücksicht nehmen.

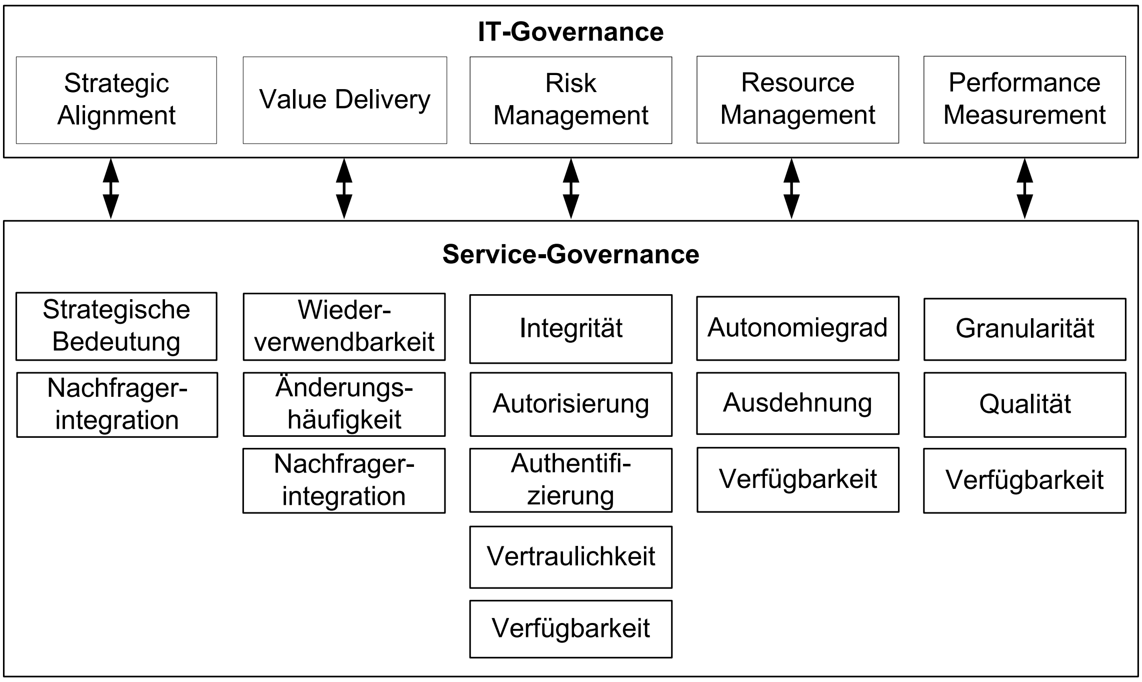

Abbildung 3: Dimensionen der Service-Governance

Abbildung 3 zeigt zusammenfassend die Zuordnung der Dimensionen der Service-Governance zu den IT Governance Focus Areas. Die Bedeutung der abgebildeten Dimensionen für die Service-Identifikation wurde in diesem Kapitel dargestellt.

4 Entwicklung eines Fragebogens als methodische Unterstützung der Service-Identifikation

Um die Governance-Aspekte bereits während der Service-Identifikation zu berücksichtigen, werden die in Abbildung 3 dargestellten Dimensionen operationalisiert. Hierbei muss zunächst entschieden werden, welche Merkmale die jeweiligen Dimensionen sinnvoll repräsentieren. Außerdem müssen die möglichen Ausprägungen der Merkmale beschrieben werden. Einige Ausprägungen sind beispielsweise binär, d.h. mit „vorhanden" und „nicht vorhanden" bzw. mit „Ja" oder „Nein" zu beschreiben. Andere kommen in Abstufungen auf einem Kontinuum vor und können bspw. auf einer 3-stufigen Likert-Skala eingeordnet werden. Freitexte sind für eine standardisierte Auswertung sehr schwierig und sollten daher vermieden werden. Die Frage nach einer Anzahl (bspw. der Schnittstellen) kann indes sehr sinnvoll sein. Einige Ausprägungen können auch K.O.-Kriterien oder zumindest dominierende Kriterien bei der Gestaltung eines Services sein.

Organisatorisch obliegt die Bewertung der in Abbildung 3 aufgeführten Dimensionen der Service Design Unit, also dem Process Owner gemeinsam mit dem späteren Service Owner. Diese benötigen dazu einen strukturierten Fragebogen, der schließlich eine

Auswertung dahingehend zulässt, ob der vom Process Owner vorgeschlagene Service-Kandidat tatsächlich als Service implementiert werden sollte. Die beiden Mitglieder der SDU einigen sich auf die Ausprägungen der verschiedenen Aspekte und werden dadurch in die Lage versetzt, einen Service Governance-konform zu gestalten.

Frage 1.1	Ist der Service geschäftsprozessspezifisch oder kann er auch für andere Prozesse verwendet werden (bspw. Posteingangsscanning)?	Sehr spezifisch	Eher spezifisch	unspezifisch
Frage 2.1	Ist der Service wandelnden Kunden- oder Marktanforderungen ausgesetzt?	Stark	Mittel	Kaum
Frage 3.1	Wie wichtig ist der Service Ihrer Meinung nach, um sich von Wettbewerbern abzusetzen?	Sehr wichtig	Mittel	Unwichtig
Frage 3.2	Würden Sie den Service bedenkenlos an einen externen Provider abgeben?	Niemals	Unter Umständen	Bedenkenlos
Frage 4.1	Wie viele Schnittstellen hat der Service	[Zahl]		
Frage 4.2	Ist der Input für den Service stets gleich?	Immer	Teilweise	Nie
Frage 4.3	Liefert der Service stets den gleichen Output?	Immer	Teilweise	Nie
Frage 5.1	Sind Kunden in den Service eingebunden?	Immer	Manchmal	Nie
Frage 5.2	Sind Mitarbeiter in den Service eingebunden?	Immer	Manchmal	Nie

Abbildung 4: Fragebogen-Auszug

Abbildung 4 zeigt einen Auszug aus einem Fragebogen, der der SDU eines Unternehmens zur methodischen Unterstützung der Service-Identifikation dienen kann. Der Fragebogen wird anhand der abgeleiteten Dimensionen unterteilt (zur Methodik siehe bspw. [Bü06]), d.h. dass bspw. alle die Autonomie (Dimension Nr. 4) des Services betreffenden Fragen in einem Fragenblock zusammengefasst sind. Die Nummerierung der Fragen ordnet diese durch die erste Ziffer den jeweiligen Dimensionen zu. So dienen die Fragen 4.1 bis 4.3 beispielsweise dazu, die Dimension „Autonomie" zu untersuchen.

Ein solcher Fragebogen ist ein integraler Bestandteil einer Methode zur fachlichen Identifikation von Services. Bei der Entwicklung eines solchen Fragebogens sollte darauf geachtet werden, dass dieser einerseits nicht zu viele Spezifika enthält, so dass er für viele Abteilung oder sogar unternehmensübergreifend verwendet werden kann. Allerdings sind zu generische Fragen auch nur bedingt geeignet. Es muss also eine Balance gefunden werden, die den Einsatz in vielen Situationen zulässt und dennoch aussagekräftige Resultate generiert.

Die sachgemäße Auswertung des Fragebogens ist im nächsten Schritt die Grundlage für eine adäquate Verwendung der Ergebnisse. Die SDU muss daher durch einen Leitfaden bei der Bewertung der Antworten unterstützt werden. Eine zweistellige Zahl von Schnittstellen deutet bspw. auf einen für einen Service mangelhaften Autonomiegrad hin und sollte von der SDU entsprechend interpretiert werden.

5 Fazit und Ausblick

Das Paradigma serviceorientierter Architekturen stellt zweifellos ein viel versprechendes Konzept zur flexiblen und agilen Anpassung der IT-Landschaft eines Unternehmens an neue Geschäftsanforderungen dar. Eine solche Architektur führt jedoch zu einem Anstieg der Komplexität und daher auch zu der verstärkten Forderung, eine SOA durch den Aufbau von Governance-Strukturen steuerbar zu gestalten. Während bisherige Beiträge lediglich eine SOA-Governance bezogen auf die gesamte Infrastruktur betrachteten, fokussiert dieser Beitrag die Governance einzelner Services. Dazu wurden die fünf IT-Governance-Kernbereiche operationalisiert und ihre Bedeutung bereits bei der Identifikation und Gestaltung von Services herausgearbeitet.

Aus den Kernbereichen der IT-Governance, den sogenannten IT Governance Focus Areas, wurden daher die für eine Service-Governance relevanten Dimensionen abgeleitet. Diese Dimensionen wurden zu einem Fragebogen konsolidiert, der die Service Design Unit bei der Identifikation und Gestaltung von Services methodisch unterstützt. Die Verwendung des Fragebogens als Leitfaden während der Service-Identifikation soll in erster Linie dazu dienen, Governance-Gesichtspunkte bereits bei der Gestaltung von Services zu berücksichtigen. Weiterer Forschungsbedarf besteht bezüglich der Auswertung eines solchen Fragebogens mit Blick auf die Tauglichkeit eines Service-Kandidaten. Nach Meinung der Verfasser kann die systematische Auswertung und Beurteilung der Governance-Dimensionen eines Services-Kandidaten mithilfe des vorgestellten Fragebogens unterstützt werden. Die Governance-Dimensionen sind allerdings nur ein Teil der Kriterien, die die Tauglichkeit determinieren. Die fachliche und technische Eignung muss im Rahmen der Service-Identifikation und -Gestaltung ebenso beachtet werden. Weiterer Forschungsbedarf besteht daher im Bereich der Wechselwirkungen und Abhängigkeiten zwischen fachlicher und technischer Eignung sowie der Tauglichkeit eines Service-Kandidaten aus Governance-Sicht.

Literaturverzeichnis

[AG09] Alter, S.; Goeken, M.: Konzeptionelle Metamodelle von IT-Governance-Referenzmodellen als Basis der Kombination und Integration in einer Multi-Modell-Umgebung, Proceedings der WI 2009, Wien, 2009.

[Ar08] Arsanjani, A.; Ghosh, S.; Allam, A.; Abdollah, T.; Ganapathy, S.; Holley, K.: SOMA: A method for developing service-oriented solutions. IMB Systems Journal, 47, 2008, S. 377-396.

[BBW09] Becker, A.; Buxmann, P.; Widjaja, T.: Value Potential and Challenges of Service-Oriented Architectures - A User and Vendor Perspective. Erscheint in: 17th European Conference on Information Systems, Verona (Italien), 2009.

[Be06] Berbner, R.; Johannsen, W.; Eckert, J.; Goeken, M.; Repp, N.: SOA Governance - Management of Opportunities and Risk. efinance lab quarterly, 4, 2006, S.4-5.

[Bi05a] Bieberstein, N.; Bose, S.; Fiammante, M.; Jones, K.; Shah, R.: Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap. Prentice Hall, Upper Saddle River, NJ, 2005.

[Bi05b] Bieberstein, N.; Bose, S.; Walker, L.; Lynch, A.: Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals. IBM Systems Journal, 44, 2005, S. 691-708.

[BK05] Böhmann, T.; Krcmar, H.: Modularisierung: Grundlagen und Anwendung bei IT-Dienstleistungen. In (Herrmann, T.; Kleinbeck, U.; Krcmar, H. (Hrsg.)) Konzepte für das Service Engineering. Modularisierung, Prozessgestaltung und Produktivitätsmanagement. Physica, Heidelberg, 2005.

[BKR04] Becker, J.; Kugeler, M.; Rosemann, M.: Process Management. A Guide for the Design of Business Processes, Berlin, Springer, 2004.

[BG09] Börner, R.; Goeken, M.: Identification of Business Services - Literature Review and Lessons Learned. In Proceedings of the 15th Americas Conference on Information Systems (AMCIS), 2009.

[Bü06] Bühner, M.: Einführung in die Test- und Fragebogenkonstruktion. Pearson, München [u.a.], 2006.

[DD07] Durst, M.; Daum, M.: Erfolgsfaktoren serviceorientierter Architekturen. HMD - Praxis der Wirtschaftsinformatik, 253, 2007, S. 18-27.

[El07]Elfatatry, A.: Dealing with Change: Components versus Services. Communications of the ACM, 50, 2007, S. 35-39.

[Er04] Erl, T.: Service-Oriented Architecture - A Field Guide to Integrating XML and Web Services. Prentice Hall, Upper Saddle River, NJ, 2004.

[GJP09] Goeken, M.; Johannsen, W.; Pfeiffer J.C.: Linking IT and Business Processes for Alignment - A Meta Model based Approach. Conference on Enterprise Information Systems, Mailand, Italien, 2009.

[Gr04] van Grembergen, W: Strategies for information technology governance. Idea Group Publishing, Hershey, PA, 2004.

[HV93] Henderson, J.C.; Venkatraman, N.: Strategic alignment: leveraging information technologyfor transforming organizations. IBM Systems Journal, IBM Corporation, Riverton New Jersey, USA, 32, Nr. 1, 1993. S. 4-16.

[IS05] Ivanov, K.; Stähler, D.: Prozessmanagement als Voraussetzung für SOA. OBJEKTspektrum, 12, 2005, S. 60-64.

[ITGI07] IT Governance Institute: COBIT 4.1, o.O., 2007.

[JG07] Johannsen, W.; Goeken, M.: Referenzmodelle für IT-Governance, dpunkt.verlag, Heidelberg, 2007.

[Jo08] Josuttis, N.: SOA in der Praxis - System-Design für verteilte Geschäftsprozesse. dpunkt.verlag, Heidelberg, 2008.

[KA07] Kohlmann, F.; Alt, R.: Business-Driven Service Modeling - A Methodological Approach from the Finance Industry. In (Abramowicz, W. & Maciaszek, L. (Hrsg.)), Business Process & Service Computing: First International Working Conference on Business Process and Services Computing, Bonn Gesellschaft für Informatik e.V., Leipzig, Germany, 2007.

[KKB07] Klose, K.; Knackstedt, R.; Beverungen, D.: Identification of Services - A Stakeholder-Based Approach to SOA Development and Its Application in the Area of Production Planning. In Proceedings of the 15th European Conference on Information Systems, St. Gallen, Switzerland, 2007.

[KKCR09] Kohlborn, T.; Korthaus, A.; Chan, T.; Rosemann, M. (2009). Identification and Analysis of Business and Software Services - A Consolidated Approach. IEEE Transactions on Services Computing, 2. Jg., Nr. 1, S. 50-64.

[KN93] Kaplan, R.; Norton, D.P.: Putting the Balances Scorecard to work. Harvard Business Review, 1993, September-October, S. 134-147.

[KSH08] Kohnke, O.; Scheffler, T.; Hock, C.: SOA-Governance - Ein Ansatz zum Management serviceorientierter Architekturen. Wirtschaftsinformatik, 50, 2008, S. 408-412.

[LH07] Legner, C.; Heutschi, R.: SOA Adoption in Practice - Findings From Early SOA Implementations. In (Österle, H., Schelp, J. & Winter, R. (Hrsg.)), 15th European Conference on Information Systems (ECIS 2007). St. Gallen, 2007.

[Lo08] Lotz, V.; Pigout, E.; Fischer, PM.; Kossmann, D.; Massacci, F.; Pretschner, A.: Towards Systematic Achievement of Compliance in Service-Oriented Architectures: The MASTER Approach. Wirtschaftsinformatik, 50, 2008, S.383-391.

[Lu08] Luftmann, J.; Dorociak, J.; Kempaiah, R.; Rigoni, E.: Strategic Alignment Maturity: A Structural Equation Model Validation. In Proceedings of the 14th Americas Conference on Information Systems (AMCIS), 2008.

[MB06] Mark, E. A.; Bell, M.: Executive's Guide to Service-Oriented Architecture. Wiley & Sons, 2006.

[Na04] Nadhan, E.G.: Seven Steps to a Service-Oriented Evolution. Business Integration Journal, 2004, S. 41-44.

[Ni08] Niemann, M.; Eckert, J.; Repp, N.; Steinmetz, R.: Towards a Generic Governance Model for Service-oriented Architectures. Fourteenth Americas Conference on Information Systems. Toronto, Canada, 2008.

[o.V.08] SOA-Security-Kompendium. Bundesamt für Sicherheit in der Informationstechnik, Bonn, 2008.

[Pa03] Papazoglou, M.P.: Service-Oriented Computing: Concepts, Characteristics and Directions. Proceedings of the Fourth International Conference on Web Information Systems Engineering. IEEE Computer Society, Rome, Italy, 2003.

[Ra06] Rabhi, F. A.; Yu, H.; Dabous, FT.; Wu, SY.: A service-oriented architecture for financial business processes: A case study in trading strategy simulation. Information Systems and e-Business Management, 5, 2006, S. 185-200.

[RB00] Reich, B. H.; Benbasat, I.: Factors that Influence the Social Dimension of Alignment between Business and Information Technology Objectives. MIS Quarterly, Management Information Systems Research Center, University of Minnesota, Minneapolis, USA, 24, Nr. 1, 2000; S. 81–113.

[Sh81] Shostack, G.L.: How to Design a Service. In (Donelly, J.H. & George, W.K. (Hrsg.)) Marketing of Services, Amer Marketing Assn., 1981.

[Si07] Simmons, S.: Don't let the greatest benefits of SOA elude you. IBM developerWorks, 2007, [URL: http://www.ibm.com/developerworks/websphere/

techjournal/0706_col_simmons/0706_col_simmons.html], Zugriff: 08.04.2009.

[SS07] Schelp, J.; Stutz, M.: SOA-Governance. HMD 253, 2007, S. 66-73.

[TK03] Tallon, P.P.; Kraemer, K.L.: Investigating the relationship between strategic alignment and IT business value: the discovery of a paradox. In (Shin, N. (Hrsg.)), Creating business value with information technology challenges and solutions, Idea Group, Hershey Pennsylvania, USA, 2003, S. 1-22.

[Wi07] Winkler, V.: Identifikation und Gestaltung von Services. Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich. Wirtschaftsinformatik, 49, 2007, S. 257-266.

[Za05] Zacharias. R.: Serviceorientierung: Der OO-König ist tot, es lebe der SOA-König. OBJEKTspektrum, 12, 2005; S. 43-52.

4.4 Artikel 6: Towards an Operationalisation of Governance and Strategy for Service Identification and Design

Veröffentlicht: Börner, R., Goeken, M., Looso, S. (2009): Towards an Operationalisation of Governance and Strategy for Service Identification and Design, in: Proceedings of the 13th IEEE International EDOC Conference 2009, 01.-04.09.2009, Auckland, Neuseeland, S. 180-188.

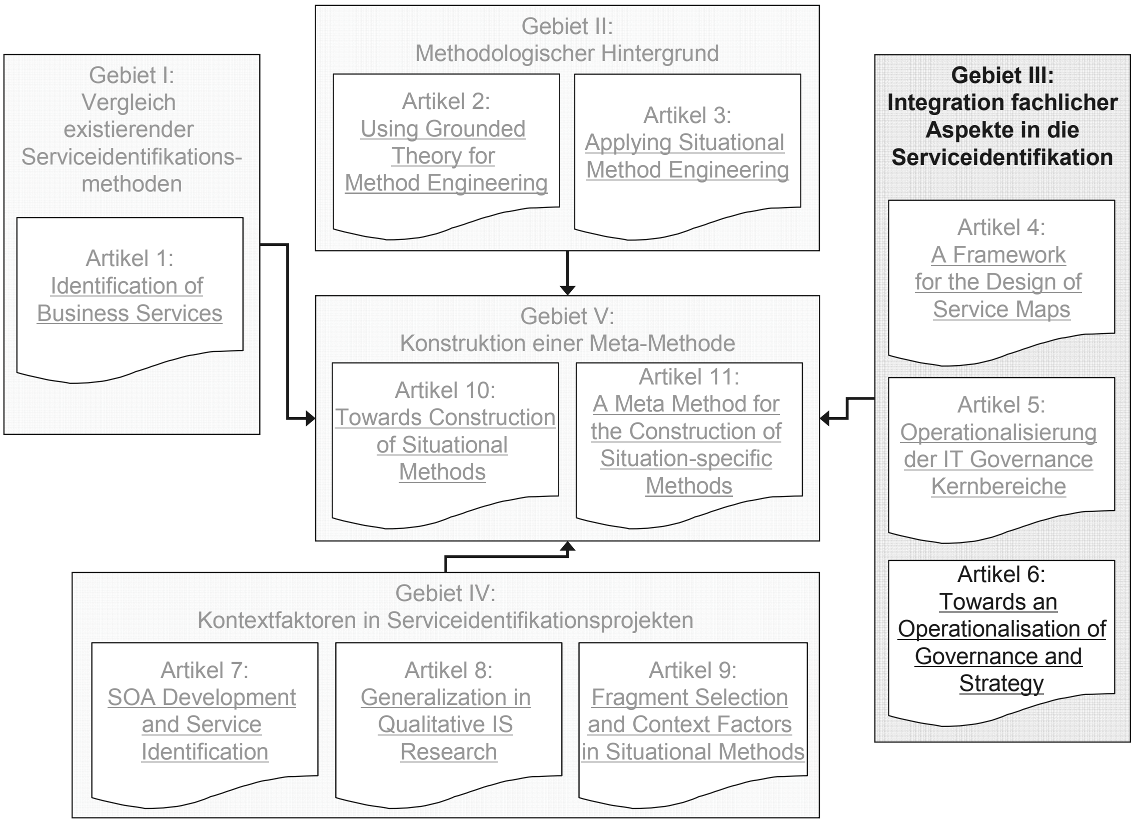

Abbildung 11: Artikel 6 im Kontext der kumulativen Dissertation

Towards an Operationalisation of Governance and Strategy for Service Identification and Design

René Börner
ProcessLab
Frankfurt School of Finance & Management, Germany
r.boerner@frankfurt-school.de

Stefanie Looso
IT-Governance-Practice-Network
Frankfurt School of Finance & Management, Germany
s.looso@frankfurt-school.de

Matthias Goeken
IT-Governance-Practice-Network
Frankfurt School of Finance & Management, Germany
m.goeken@frankfurt-school.de

Abstract

Service orientation is a promising paradigm for business architectures. Implementing a service-oriented architecture (SOA) promises increasing flexibility as well as agility and decreasing development and maintenance costs of IT landscapes. Simultaneously with these advantages, the implementation of an SOA entails some inherent challenges. The flexible orchestration of services increases the complexity of the whole system significantly and can even result in decreasing performance [1]. A holistic management of technology and business processes is therefore necessary [2]. We believe that the essential SOA management tasks include topics such as strategy, governance, processes as well as infrastructure. These management topics can be addressed on different levels of granularity, e.g. at an SOA governance level or a single service level.

This article focuses on individual services and shows how to support their identification and design. Therefore, we present specific deduced dimensions of the generic topics governance and strategy. In addition, we present roles and techniques in order to consider these topics already during the early phases of an SOA implementation, i.e. in the identification and design process.

Keywords

Service-oriented Architecture; SOA Governance; Service Governance; Service Strategy; Service Identification; Service Design; Method

I. Success Factors of Service-Oriented Architectures

Only companies with the flexibility to adjust quickly to a changing environment may survive in the long-run [3]. In this context, service orientation is a broadly discussed paradigm for business architectures. Frequently mentioned issues in conjunction with service orientation are a quicker adaptation to changes in business processes through greater flexibility and agility, greater reusability of services through loose coupling, which is associated with a high degree of internal cohesion of services [4]. The reusability of services prevents the unnecessary provision of redundant functions and thus reduces development and maintenance costs of the IT infrastructure. The loose coupling allows an almost virtuosic composition of services and discloses sourcing potentials, for example through the use of web services [5]. But the implementation of a service-oriented architecture also results in a number of challenges. The flexible orchestration of services increases the complexity of the whole system significantly and can even result in a deterioration of performance [1]. But how to cope with these challenges?

It is commonly accepted that processes should be derived from strategy and that their execution needs some sort of infrastructure [6, 7]. Goeken and Johannsen present governance as an additional topic in their business architecture as shown in Fig. 1.

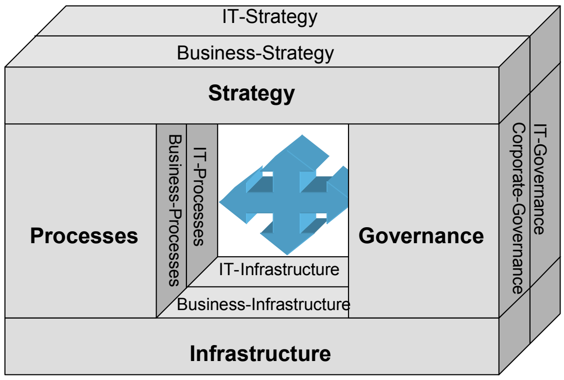

Figure 1. Business Architecture [21]

To deal with the inherent challenges of an SOA the organization has to consider the same aspects in a complex environment, namely technical, business, strategic and governance aspects. These four management topics have to be at the centre of attention in the process of an SOA implementation.

Furthermore, we believe that these four topics should already play an important role during the identification and design phases of each individual service. Governance aspects, for example, should influence the service identification process. Fig. 2 illustrates exemplary the differences between SOA governance and service governance. SOA governance refers to the service-oriented architecture as a whole and should not be mixed up with service governance which is concerned with the life cycle (identification, design,…, deletion) of each individual service.

In this paper we present our contribution to the management topics governance and strategy with a focus on individual services. We particularly concentrate on service identification and service design which belong to the early phases of the SOA lifecycle and are highlighted in Fig. 2.

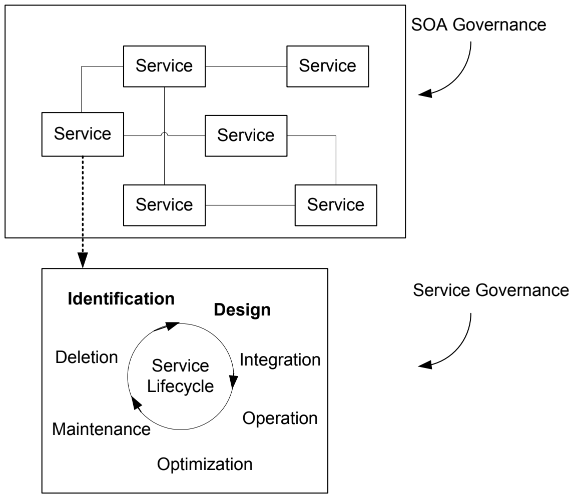

Figure 2. Comparison of the properties between SOA and Service Governance

Service identification is the task of the business process owner (BPO). Usually, a Business Process Modeling project precedes such identification. In the following, the result of this first step is referred to as a "service candidate" (see Fig. 5). Only after the service candidate has been scrutinised by the Service Excellence Centre (SEC) the design process is conducted by the Service Design Unit (see section 5 for details). Finally, this leads to a service that can be implemented subsequently.

For most existing approaches, a concrete operationalisation of the aforementioned management topics at the service level is unsatisfactory or completely missing. This article focuses on governance and strategy during the identification and design phases of services. In order to guarantee an economically advantageous implementation of an SOA, this article shows how to support a top-down approach for service identification. This is considered a first step that has to be complemented by proving the technical feasibility of identified services [8] which is out the scope of this paper. Finally, this leads to a hybrid approach for the identification and design of services which is recommended by many authors [9; 10].

The remainder of the paper is organised as follows. Section 2 reviews related literature concerning both identification methods and existing approaches to relevant management task of an SOA. Chapter 3 shows how the IT governance focus areas can be applied on services. The utilization of strategic aspects for the identification and design of services will be discussed in section 4. For both areas section 5 will – in the sense of method engineering – develop roles and techniques to assist the construction of a method for service identification. We suppose that service governance und service strategy are essential for the generic management topics governance and strategy. Addressing both aspects during the phases of identification and design of services is crucial for a successful business-oriented SOA implementation. Section 6 will draw conclusions and give an outlook on further research.

II. EXISTING APPROACHES TO SERVICE IDENTIFICATION AND SERVICE DESIGN

A couple of approaches for service identification and design can be found in related literature. Although common patterns can be identified among these approaches, they do differ considerably in detail. This is partly due to their origination from different industry sectors. A detailed analysis of the five approaches briefly presented in the following can be found in Börner and Goeken [11].

Klose, Knackstedt and Beverungen [12] emphasise the importance of the “business point of view”. They derive services on the basis of Business Process Modelling and stress issues such as customer interaction. Their approach stems from the manufacturing industry. Strategic aspects like sourcing are not discussed explicitly. The need for IT governance can only be found implicitly in this approach. An SOA governance in particular is missing entirely. Thus, governance or strategic aspects are ignored in the identification process.

Böhmann and Krcmar [13] show a distinct commitment to goal orientation. The goals they identify are, however, very similar to the general SOA goals [4]. The application of their modularisation matrix within the IT services industry is the outstanding feature of this approach. Threats and opportunities of modular service architectures in general are discussed. An impact on the identification of single services is not considered though. An explicit treatment of IT and SOA governance issues is also missing.

Three more approaches are taken from the financial services industry. Arsanjani et al. [14] as well as Kohlmann and Alt [15] follow a clearly business oriented identification on services based on Business Process Modelling. Although Winkler [16] also grounds her identification of services on Business Process Modelling, her approach is much more technical and leads to a more object-oriented than service-oriented view. Whereas the latter approach does not consider strategic aspects at all, the others include at least short discussions on sourcing and reference models. By stressing the importance of consistently naming services, only Kohlmann and Alt look at one aspect of SOA governance that directly affects the identification process.

Kohlborn et al. [43] distinguish between business services and software services. Due to this differentiation (that is only implicitly addressed in other approaches by hierarchies) they manage to focus on business processes in the first place. Subsequently, a prioritization phase is used to link the first three phases of their method with the last three – technically oriented – phases. By building on approaches such as [12] and overcoming weaknesses of others, strategic aspects are incorporated fairly well. However, governance issues are not at the core of their approach.

Many authors describe issues like interface orientation, interoperability, loose coupling, modularity and reusability as key characteristics of an SOA [17-20]. Undoubtedly, these characteristics are typical for an SOA, but they reflect only a technical point of view. An explicit connection to management topics is missing in these approaches. Internal policies are marginally mentioned in [14] and [15]. IT governance in the broadest sense is implicitly mentioned by other authors [12; 13; 15]. [21] discuss the relationship between IT governance and the governance of a service-oriented architecture.

Following the approach of [21], the next section therefore describes the implementation of service governance during the identification and design phases of services. This entails operationalising the five IT Governance Focus Areas for an application to services. In other words, the dimensions for service governance are derived from the five key areas of IT governance. Section 4 focuses on service strategy. In addition to existing more technical oriented approaches we deduce and operationalise dimensions of service strategy as well as dimensions of service governance in the following sections.

III. IT GOVERNANCE FOCUS AREAS AND THEIR APPLICATION TO SERVICE LEVEL

The key areas of IT governance comprise the five so-called IT Governance Focus Areas: strategic alignment, value delivery, risk management, resource management, and performance measurement [21; 22]. The key areas focus on aspects which require management's attention in order to appropriately manage and direct IT from a business perspective. In the following, these five key areas will be explained individually, and their application for services and their relevance within the context of an SOA will be discussed. The objective of this section is to derive dimensions for service governance from the established key areas within the COBIT (Control Objectives for Information and Related Technology) frameworkEase of Use

A. Strategic Alignment

Aligning the business management perspective and IT represents a multi-layered problem, which encompasses not only the strategies of these two areas but also processes architectures, infrastructure, and cultural aspects. A common assumption is that the alignment starts at the strategic level [23]. This requires as a first step to align the business and the IT strategies. In order to accomplish this, the Strategic Alignment Model (SAM) by [6] divides a business into four domains: business and IT, as well as a strategic (external) and an infrastructural (internal) domain. [6] regard alignment as, "a balance among the choices made across all four domains." Based on this, [24] describe a relationship between standard business processes and standard IT processes [cf. 25]. The authors thus illustrate that the alignment of the business perspective and IT necessitates aligning levels subsequent to strategy alignment. [26] consider a mutual understanding of the objective and the pursued avenue for reaching the objectives as a critical success factor for aligning the business perspective and IT. Key to reaching such a common understanding is efficient and pro-active communication between employees in business operations and IT. Such communication fosters customer integration, as advocated by [13], since the functional departments can be viewed as internal customers of the IT department. [27] emphasises a business management focus as well in order to secure the financing required for IT development and maintenance and to deploy the services throughout the organisation. A consistent and uniform use of terminology for services, as postulated by [15], serves not only as a pre-requisite for a high degree of reusability but reflects the alignment of the business management perspective and IT as well.

B. Value Delivery

This key area comprises the stipulation that IT has to make substantial and verifiable contributions to the success of the business. This entails as one important aspect the demand-driven development of IT products in terms of value-oriented IT services. In order to provide value-oriented IT services that actually meet customer demand, it is critically important to align customers and IT services suppliers. Such an alignment represents the main task of the previously described focus area strategic alignment. [28] establish a relationship between the degree of strategic alignment and the value added by IT. Moreover, a study by [5] substantiates that an SOA indeed enhances the alignment between the business perspective and IT. However, this holds true only if such an alignment between the business perspective and IT is defined within the scope that "the IT solutions provide a strong fit with the business" [5]. [29] considers the value generation through an alignment of the business perspective with IT as the most important task of IT governance.

The reciprocal nature of the relationship between value delivery and strategic alignment is also visible at the service level. Aspects such as update frequency and reusability can only be measured or forecasted in a meaningful way if business knowledge and IT know-how are combined. The reusability of services and the resulting economic benefits that may be generated, specifically eliminating redundancies [15; 27], represent a decisive success factor in the key area value delivery. An essential pre-requisite for a high reusability rate is the standardised use of terminology for services. As previously explained, the integration of the customer, as explicitly advocated by [13], is often a decisive factor in order to create a benefit for an external or internal customer. The configuration of the customer interface (also referred to as "line of interaction"[30]) has to be considered during the service design phase. The higher the degree of customer integration the more likely are changes to the customer interfaces and the service functionality, as customer requirements may change frequently.

C. Risk Management

The objective of risk management is the identification and analysis of risks. In addition, a business should possess a clear understanding of its own risk preferences and attitude. Finally, risk management considers knowledge of applicable regulations and laws as well as the division of responsibility within an organisation [cf. 21]. The most important security aspects of an SOA [cf. also 31] are discussed in the following.

A specific service often requires authentification, i.e. users (or other entities) have to be clearly identified, for example, through a user name and password. The authorisation process ensures through the use of role-based access control that only users with an appropriate authorisation are able to retrieve or change data. Depending on the sensitivity of the data and the importance of the process, a determination has to be made concerning which users are entitled to which type of access. Once the authentification and authorisation procedures have been implemented, and provided that an adequate, technological solution, such as protocols that cannot be compromised, has been developed, the integrity and confidentiality of the data inherent in the service can be assured. Service availability plays, on the one hand, a vital role for customer satisfaction. The resulting consequences of this will be examined in the discussions of the focus areas “resource management” and “performance management.” On the other hand, within the context of regulatory requirements, service availability also has to be considered within the focus area “risk management.” Particularly financial services providers have to ensure service availability in compliance with the requirements in MARisk or Basel II. Service availability must be ensured especially for the highly mechanised and automated trading systems.

D. Resource Management

Resource management aims to achieve the consistent management of resources within an organisation. This entails paying particular attention to optimising investments and managing these appropriately. The IT governance reference model COBIT [22], identifies software applications, information, infrastructure, and employees as the fundamental IT resources.

Applied to a service, this requires a determination of the needed inputs in terms of data, hardware, and personnel, and whether these inputs always remain constant. It is entirely possible that a service may be differently utilised, depending on the input factors. This, in turn, impacts the service output, which therefore has to be defined in terms of the resource factors as well. A more precise definition of the service interfaces allows for better and more exact determination of the required resources. A high degree of autonomy, i.e. as much inner-service cohesion as possible, combined with loose relationships to other services, supports the resource management activity. A service expansion also impacts resource management. In such a case, one has to examine whether a service affects multiple business functions. If this is the case, i.e. various different functional departments are involved, then the complexity of resource capacity planning increases. Moreover, such a service is typically not suited to be outsourced. The involvement of several functional roles would necessitate outsourcing an entire business process rather than a single service. The required level of service availability obviously impacts the quantity of resource reserves. Critical processes require sufficient resource buffers that guarantee even during emergency situations flawless and uninterrupted operation (cf. section “Risk Management”).

E. Performance Measurement

The purpose of performance management is to measure and control the implementation of strategy, projects, processes, and resource utilisation among others. The operationalisation and measurement of strategic initiatives and activities verifies and supports the degree of goal attainment. This is accomplished using, for example, a Balanced Scorecard or any other methodology that provides for the quantification of strategy in “measurable units.” The measurement goes beyond the mere accounting requirements, as so-called “soft“ factors are considered as well [32]. COBIT recommends determining objectives and their measurements at three levels: IT objectives and measurements, which define the expectations the functional departments have of IT; process objectives and measurements, which define the requirements for the IT process so that IT objectives can be achieved; and, finally, activity objectives and corresponding measurements [cf. 25].

In general, performance measurement applies to services as well, but certain unique aspects have to be considered. Firstly, applicable metrics or Key Performance Indicators (KPIs) have to be identified for the particular service. The complete service delivery time or the response time to the service request represent suitable KPIs. The latter is closely correlated to service availability, as response time serves as the critical measure for service availability. All quality-critical attributes should be determined and measured, regardless of whether the service is intended for internal or external customers. In outsourcing arrangements, it is essential to establish formal measurements through service-level arrangements. Measuring service performance requires, however, careful and discerning examination, since a service may comprise more than one activity with relatively limited scope. Depending on the degree of granularity, a service may encompass an entire business process or, at least, several parts of such a process. If, for example, measurements are conducted at the activity level, one has to determine whether the individual measurements can be easily aggregated. The design of services therefore has to take the measurement of performance into consideration.

Fig. 3 provides a summary of how the dimensions of service governance relate to the IT governance focus areas. The significance of the depicted dimensions for the service identification phase has been illustrated in this section.

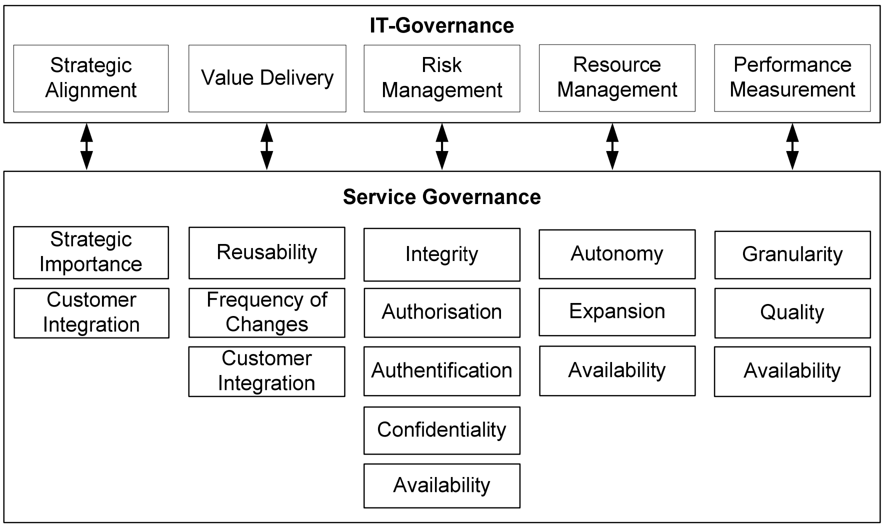

Figure 3. Dimensions of Service Governance

IV. SELECTED STRATEGIC ASPECTS AND THEIR APPLICATION ON SERVICE LEVEL

A consideration of strategic aspects is very important because implementation of an SOA is not done for its own sake but seeks tangible benefits for the company. Due to this, strategic relevance must not be omitted from the early stages of the SOA lifecycle, i.e. the identification and design of services. A categorization by Allen [33] therefore differentiates between three types of services:

- Commodity services are stable, sufficiently established services every market player must have. They are suitable for outsourcing and standardization.
- Territory services are fairly widespread but less stable and usually represent business rules.
- Value-added services constitute the special value of a company's product or service in the market, i.e. a company's core competence. It is this highly innovative service that gives distinction to the company.

Assigning a service candidate correctly to one of these categories is crucial for subsequent sourcing strategies. A thorough analysis of all kinds of transaction costs must give reasons for a make or buy decision for certain services. Specificity, frequency and uncertainty of transactions (and the services representing them) determine the suitability and costs for adequate coordination forms [34]. Commodity services are more likely to be purchased in the marketplace whereas value-added services are probably found in organisational hierarchies.

A. Maintenance and Operation Costs

These costs correlate strongly with the quantity and heterogeneity of IT systems in an enterprise. Although an SOA can lead to an increase of complexity due to the need for coordination of services [1], the number of different systems and applications is generally reduced. Thus, costs for maintenance are decreased [35; 36]. Moreover, well-defined functions and interfaces contribute to the robustness of IT systems which in turn lessens operation costs.

On a service level, this implies that inputs and outputs (i.e. the interfaces) have to be defined concisely. Taken this for granted, the functionality of any service can be adapted to changing customer or legal requirements using the same inputs and delivering equal outputs. Hence, other services are not influenced by these changes. Testing efforts for the new functionality can be reduced to a minimum. Subsequently, overall costs for a new functionality are reduced significantly. Thus, the autonomy of a service is a determining factor for both maintenance and operation costs as well as resource management described in section 3.

B. Vendor Dependency

Implementation of an SOA decreases vendor dependency because such architecture is platform independent. Firstly, this gives a company the opportunity to use open source software. Thus, a necessity of buying licenses would vanish and lead to immediate savings. However, the total cost of ownership should be taken into account when thinking about open source products. Secondly, a lock in effect is avoided, i.e. a decision in favour of software from one vendor does not necessarily influence future decisions on enterprise IT systems. Hence, the company is not bound to a vendor because of prohibitively high swapping costs. In the past, proprietary standards often led to a high vendor dependency [37]. Thirdly, web services can flexibly be used and increase the agility of business processes. These web services can be purchased ad hoc in the marketplace without long-lasting contracts. Every time there is a need to buy a service the cheapest provider at that time can be chosen to deliver it.

Security aspects can be more important for some services than for others. Generally, an SOA is not more or less secure than a monolithic application provided by one vendor. Availability and authorisation mechanisms for example can be guaranteed by service level agreements. However, ensuring these features will prove much more complex when many services from different providers are purchased on an ad hoc basis without permanent contracts.

Thus, if availability is critical to a single service the flexibility of web services must be weighed against a holistic application bound to one vendor. Still, decreasing dependency on software suppliers can be an important strategic goal and can minimize costs considerably. If a service is classified as commodity service that is not critical to the company's core competences and most distinguishable processes it should be purchased as cheap as possible on the web. This clearly reduces vendor dependency. Even if availability is crucial to a service it might be purchased in the marketplace. When the failure of one provider can easily be made up by others the risk of non-availability is neglectable.

Therefore, an analysis concerning attributes like authorisation, authentification, confidentiality and availability is necessary in the service identification phase. Availability was discussed exemplarily in the previous paragraph being as relevant as the others. Strategic decisions regarding vendor dependency are closely linked to risk management aspects derived from the IT governance focus areas in section 3 and deal with similar service attributes.

C. Demand-Oriented Quality of Service Level

Flexible orchestration of services enables a demand-oriented quality of service level for products. Customers receive exactly the quality they request. Thus, customer satisfaction is increased at the same time. This kind of orchestration allows for an individualization of products in the sense of mass customization. This individualization does not necessarily increase operation expenses because the final product is tailored to a customer's demand from standardized parts, i.e. services.

Since service candidates in their existing design are seldom as standardized as they could be, their standardization potential has to be examined in the identification phase. If these candidates have a huge number of potential outputs the necessity of this variety should be checked and reduced if possible.

D. Specialisation on Core Competences

Specialisation on core competences plays an ever bigger role in today's competitive environment [38]. The management has to decide which place a company should take within a distributed value chain [39]. Depending on that a service which is a commodity for one company can be a core competence of another. For investment banks for example the settlement of trades is a commodity which they take for granted. The settlement bank on the other hand is only concerned with settling trades from various institutions. They have to guarantee a high quality and fast clearing of settlements. At the same time they have to control their costs in order to be able to offer the settlement at a marketable price.

Consequently, identified service candidates can be classified on the basis of their strategic importance. Services that deliver an exceptional value to customers should be considered valuable to a company and thus not be outsourced. On one hand this could be services that are crucial for the quality of a product. On the other hand this could be true for unique services that cannot be imitated by competitors.

E. Time-to-Market

Product lifecycles are being reduced for many years now and companies struggle to keep up with the speed to develop new products. Deployment of services can significantly reduce the time-to-market of new products [40; 44]. Particularly companies of the service sector need to care about this because their products are imitated almost immediately by competitors. Due to its agility and flexibility SOAs can react quickly to changing customer requirements. This advantage can be crucial to position new products successfully in the marketplace.

Single services can be able to reduce the time-to-market if they are sufficiently standardized and can be fitted into new developed processes neatly. Thus, service candidates have to be checked for their degree of standardization during the identification process.

Fig. 4 summarises the dimensions of strategy and their operationalisation for services.

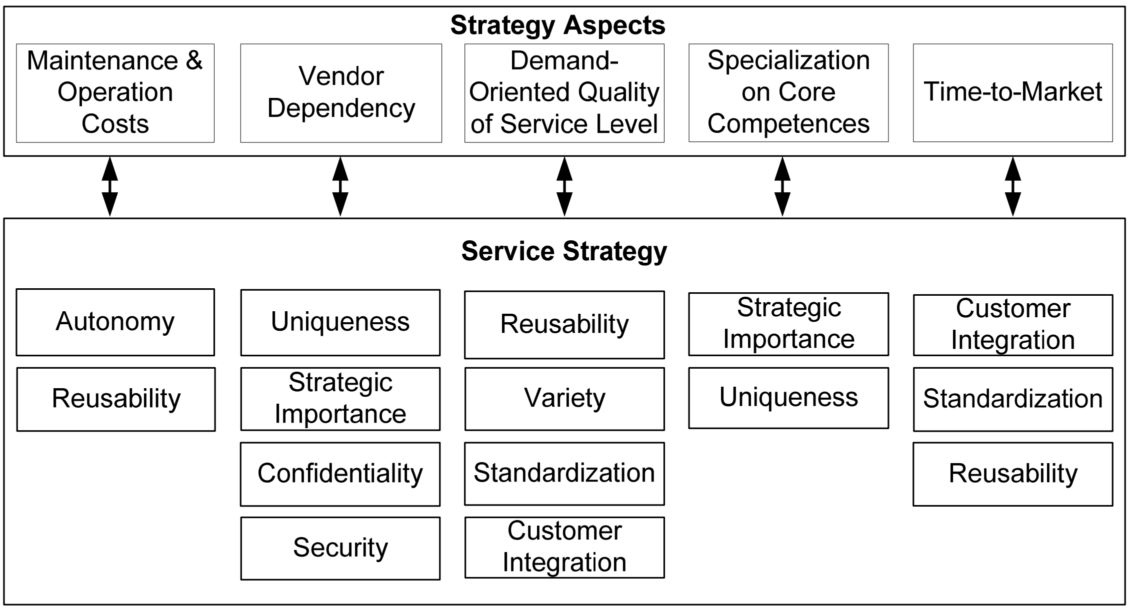

Figure 4. Dimensions of Service Strategy

V. ROLES AND TECHNIQUES TO SUPPORT SERVICE IDENTIFICATION

In order to operationalise the previously discussed strategic and governance aspects, this article proposes certain measures that should be included in a company's service identification method. For several years there have been efforts to guide the development of methods in order to guarantee a high quality. The task of method engineering is to give this guidance. Nowadays, the concept of method engineering is much discussed in the information systems community. The most popular approaches all identify activities, roles, results, techniques and the sequence of activities as important components of methods (for a detailed discussion see [41]). Existing approaches hardly address roles and techniques to support the process of service identification and service design [11]. Thus, this section presents two examples that show how roles and techniques support service identification and incorporate strategic and governance aspects on service level.

As described in section 2 IT governance aspects are widely ignored so far on service level. Strategic alignment for example should be further enhanced through alignment at the subsequent levels. For the purpose of designing services, this article recommends supporting an alignment organisationally via the appropriate composition of the "Design Team." A Service Design Unit (SDU), consisting in equal numbers of users from the functional departments and IT employees, should be established for this purpose (see Fig. 5). One team member assumes the role of Business Process Owner (BPO), who informs the SOA unit within the organisation, i.e., the Service Excellence Centre (SEC), about a potential service (the so called service candidate) that should be subjected to further analysis. Hence, the BPO and a member from the SEC, who will later on act as Service Owner (SO), jointly constitute the SDU. Since the BPO initiates the identification of the service, it is in principle assured that this service candidate meets business perspective requirements. The joint effort of the detailed configuration of the candidate into a service ensures that technological restrictions as well as the criterion of "strategic relevance“ are taken into consideration. Strategic alignment is therefore supported by an alignment at the service level.

In order to make sure that both strategic and governance aspects are already considered during the identification of a service, the dimensions depicted in Fig. 3&4 have to be operationalised even further. This entails as a first step the determination of which attributes meaningfully reflect the individual dimensions. In addition, the possible characteristics of the attributes have to be described.

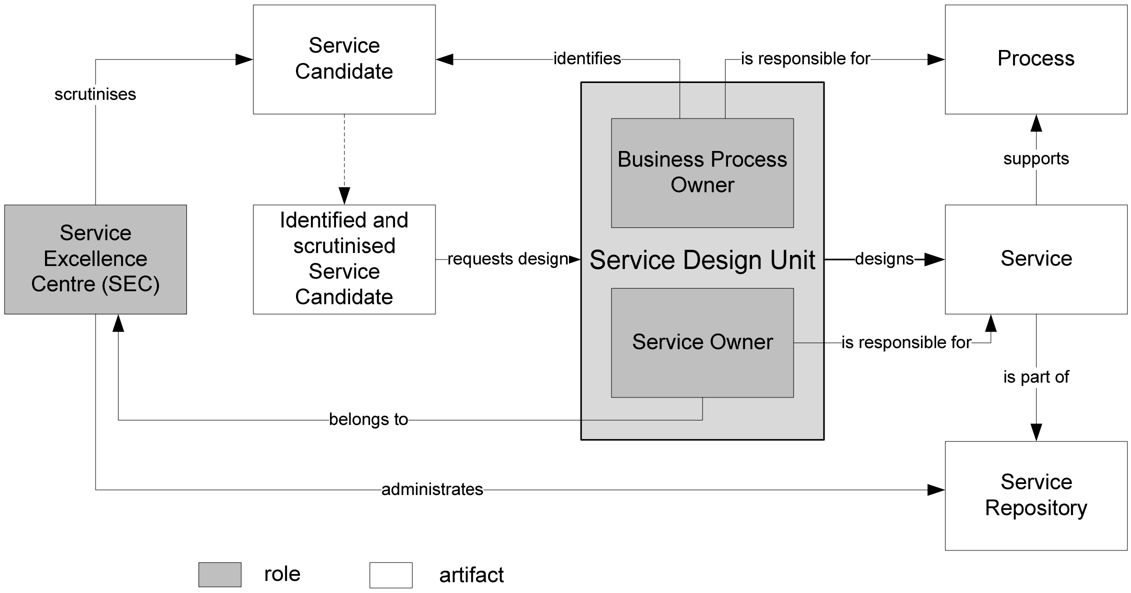

Figure 5. Integration of the Service Design Unit in the Organisational Context

For example, some attributes are binary and therefore have to be characterised as "present" and "not present," or as "yes" and "no." Other attributes occur in gradation on a continuum and can be classified, for example, on a three-grade Likert-scale. Free text complicates a standardised analysis and should therefore be avoided. However, it may be very worthwhile to assess quantities of observations (e.g. of interfaces). It is possible that some characteristics serve as "no go" criteria or as dominant criteria for the actual service design.

From an organisational perspective, the responsibility for the assessment of the dimensions showed in Fig. 3&4 rests with the SDU, i.e. with the BPO and the ultimate SO. In order to accomplish their task, they need certain techniques. A structured questionnaire that is described in the following can be such a technique. It facilitates an evaluation of whether or not the service candidate proposed by the process owner should indeed be implemented as a service. The two SDU members agree on the characteristics of the different attributes and are therefore in a position to develop conforming service governance.

Fig. 6 depicts an excerpt from a questionnaire that may assist the SDU within an organisation in the identification of services. The questionnaire is divided on the basis of the derived dimensions (for the methodology, see e.g. [42]), so that, for example, all questions pertaining to the autonomy of the service (dimension 4) are contained within one block of questions. The numbering scheme assigns the questions to the individual dimensions on the basis of the first digit. For example, questions 4.1 through 4.3 aim to analyse the dimension "autonomy."

Such a questionnaire serves as an integral element of a methodology for the process-oriented identification of services. When developing such a questionnaire, one should make an effort to not include questions that are too specific, so that the tool can be used in different departments or even organisation-wide. At the same time, however, questions that are too generic have limited use as well. The challenge is therefore to find a balance that permits an application of the questionnaire in many situations while delivering conclusive results.

VI. SUMMARY AND NEXT STEPS

The paradigm of service-oriented architectures undoubtedly represents a promising concept for the flexible and dynamic adaptation of organisation's IT landscape to new and changing business requirements. However, such architecture results in higher complexity and therefore increasingly calls for making the SOA manageable through the development of governance structures. Additionally, strategic goals of a company should explicitly be considered when an SOA is implemented. Whereas existing publications examined SOA governance for the entire infrastructure, this article focuses on governance of individual services. This involves operationalising the five key areas of IT governance and highlighting their relevance already during the phases of identifying and designing services. The same is true for the so far hardly covered field of strategy. High level goals are therefore operationalised for single services. Finally, this article promotes concepts of method engineering, i.e. roles and techniques, to support the incorporation of strategy and governance into the identification and design process of services.

Question 3.1	In your opinion, how important is this service to distinguish the own company from competitors?	very important ☐	average ☐	not important ☐
Question 3.2	Would you leave this service to an external provider without hesitation?	never ☐	possibly ☐	without hesitation ☐
Question 4.1	How many interfaces does the service have?	[number]		
Question 4.2	Is the input for the service always the same?	always ☐	sometimes ☐	never ☐
Question 4.3	Does the service always deliver the same output?	always ☐	sometimes ☐	never ☐
Question 5.1	Are customers being integrated in the service?	always ☐	sometimes ☐	never ☐
Question 5.2	Are staff members being integrated in the service?	always ☐	sometimes ☐	never ☐

Figure 6. Excerpt from a Questionnaire

As shown in Fig. 1, strategy and governance are not the only relevant topics. Covering the remaining fields is undoubtedly an important further step to improve service identification and design.

A questionnaire as depicted in Fig. 6 is certainly an important technique but has to be complemented by others to ensure that all method activities can be performed successfully by the respective roles. Moreover, the SDU members need guidance in how to interpret the results of this questionnaire. Thus, it is important to create guidelines for drawing conclusions from the received answers.

REFERENCES

[1] F.A. Rabhi, H. Yu, F.T. Dabous and S.Y. Wu, "A service-oriented architecture for financial business processes: A case study in trading strategy simulation," Information Systems and e-Business Management, vol. 5, no. 2, 2006, 185-200.

[2] R. Berbner, W. Johannsen, J. Eckert, M. Goeken and N. Repp, "SOA Governance - Management of Opportunities and Risk," efinance lab quarterly. no. 04, 2006, 4-5.

[3] J. Becker, M. Kugeler and M. Rosemann, Process Management. A Guide for the Design of Business Processes, Springer, Berlin: 2004.

[4] M.P. Papazoglou, "Service-Oriented Computing: Concepts, Characteristics and Directions," in Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003, 3-12.

[5] A. Becker, P. Buxmann and T. Widjaja, "Value Potential and Challenges of Service-Oriented Architectures - A User and Vendor Perspective," in Proceedings of the 17th European Conference on Information Systems (ECIS 2009), 2009.

[6] J.C. Henderson and N. Venkatraman, "Strategic alignment: leveraging information technology for transforming organizations", IBM Systems Journal, vol. 32, no. 1, 1993, 4-16.

[7] H. Krcmar, Informationsmanagement, Springer, Berlin: 2006.

[8] E.G. Nadhan, "Seven Steps to a Service-Oriented Evolution," Business Integration Journal, 2004, 41-44.

[9] K. Ivanov and D. Stähler, "Prozessmanagement als Voraussetzung für SOA," OBJEKTspektrum, vol. 12, no. 6, 2005, 60-64.

[10] R. Zacharias, "Serviceorientierung: Der OO-König ist tot, es lebe der SOA-König," OBJEKTspektrum, vol. 12, no. 2, 2005, 43-52.

[11] R. Börner and M. Goeken, "Methods for Service Identification: A Criteria-Based Literature Review," in 7th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS), 2009, 76-84.

[12] K. Klose, R. Knackstedt and D. Beverungen, "Identification of Services - A Stakeholder-Based Approach to SOA Development and Its Application in the Area of Production Planning," in ECIS'07: Proceedings of the 15th European Conference on Information Systems, 2007, 1802-1814.

[13] T. Böhmann and H. Krcmar, "Modularisierung: Grundlagen und Anwendung bei IT-Dienstleistungen", in Konzepte für das Service Engineering. Modularisierung, Prozessgestaltung und Produktivitätsmanagement, T. Herrmann, U. Kleinbeck, and H. Krcmar, Editors, Heidelberg: 2005, 45-83.

[14] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy and K. Holley, "SOMA: A method for developing service-oriented solutions," IBM Systems Journal, vol. 47, no. 3, 2008, 377-396.

[15] F. Kohlmann and R. Alt, "Business-Driven Service Modelling - A Methodological Approach from the Finance Industry," in Business Process & Service Computing: First International Working Conference on Business Process and Services Computing, 2007, 180-193.

[16] V. Winkler, "Identifikation und Gestaltung von Services. Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich," Wirtschaftsinformatik. vol. 49, no. 4, 2007, 257-266.

[17] T. Erl, Service-Oriented Architecture - A Field Guide to Integrating XML and Web Services, Prentice Hall, NJ: 2004.

[18] N. Josuttis, SOA in der Praxis - System-Design für verteilte Geschäftsprozesse, dpunkt.verlag, Heidelberg: 2008.

[19] N. Bieberstein, S. Bose, L. Walker and A. Lynch, "Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals," IBM Systems Journal. vol. 44, no. 4, 2005, 691-708.

[20] C. Legner and R. Heutschi, "SOA Adoption in Practice - Findings From Early SOA Implementations," in 15th European Conference on Information Systems (ECIS 2007), 2007, 1643-1654.

[21] W. Johannsen and M. Goeken, Referenzmodelle für IT-Governance, dpunkt.verlag, Heidelberg: 2007.

[22] IT Governance Institute, COBIT 4.1, 2007.

[23] V. Lotz, E. Pigout, P.M. Fischer, D. Kossmann, F. Massacci and A. Pretschner, "Towards Systematic Achievement of Compliance in Service-Oriented Architectures: The MASTER Approach," Wirtschaftsinformatik, vol. 50, no. 4, 2008, 383-391.

[24] M. Goeken and J.C. Pfeiffer, "Linking IT and Business Processes for alignment - A meta model based approach," in Conference on Enterprise Information Systems ICEIS, 2009.

[25] S. Alter and M. Goeken, "Konzeptionelle Metamodelle von IT-Governance-Referenzmodellen als Basis der Kombination und Integration in einer Multi-Model-Umgebung," in Business Services: Konzepte, Technologien, Anwendungen, H.R. Hansen, D. Karagiannis and H.G. Fill, Editors, 2009.

[26] B.H. Reich and I. Benbasat, "Factors that Influence the Social Dimension of Alignment between Business and Information Technology Objectives," in MIS Quarterly, Management Information Systems Research Center, 2000, vol. 24, no. 1, 2000, 81–113.

[27] S. Simmons, "Don't let the greatest benefits of SOA elude you," in IBM developerWorks, 2007.

[28] P.P. Tallon and K.L. Kraemer, "Investigating the relationship between strategic alignment and IT business value: the discovery of a paradox, " in Creating business value with information technology challenges and solutions Idea Group, N. Shin, Editor, Hershey: 2003, 1-22.

[29] W. v. Grembergen, Strategies for information technology governance, Idea Group Publishing, Hershey: 2004.

[30] G.L. Shostack, "How to Design a Service." in Marketing of Services, J.H. Donelly and W.K. George, Editors, 1981, 221-229.

[31] V., SOA-Security-Kompendium, Bonn: 2008.

[32] R.S. Kaplan and D.P. Norton, "Putting the Balances Scorecard to work," Harvard Business Review, September-October, 1993, 134-147

[33] P. Allen, Service Orientation: Winning Strategies and Best Practices, Cambridge University Press, Cambridge: 2006.

[34] A. Picot, R. Reichwald and R.T. Wigand, Die grenzenlose Unternehmung, Gabler, Wiesbaden: 2001.

[35] S. Reinheimer, F. Lang, J. Purucker and H. Brügmann, "10 Antworten zu SOA," HMD - Praxis der Wirtschaftsinformatik. vol. 44, no. 253, 2007, 7-17.

[36] M. Koch and M. Rill, "Erfolgsfaktoren zur Einführung serviceorientierter Architekturen bei Banken," in Innovationen im Retail Banking, D. Bartmann, Editor, Weinheim: 2005, 383-405.

[37] B. Mueller, G. Viering, F. Ahlemann and G. Riempp, "Towards Understanding The Sources Of The Economic Potential Of Service-Oriented Architecture: Findings From The Automotive And Banking Industry," in Proceedings of the 15th European Conference on Information Systems, 2007, 1608-1619.

[38] C.K. Prahalad and G. Hamel, "The Core Competence of the Corporation," Harvard Business Review, vol. 68, no. 3, 1990, 79-91.

[39] R. Baskerville, M. Cavallari, K. Hjort-Madsen, J. Pries-Heje, M. Sorrentino and F. Virili, "Extensible Architectures: The Strategic Value of Service-oriented Architecture in Banking," in European Conference on Information Systems, 2005.

[40] S. Schulte, N. Repp, J. Eckert, R. Steinmetz, R. Schaarschmidt and K. von Blanckenburg, "Service-oriented Architectures - Status Quo and Prospects for the German Banking Industry," International journal of interoperability in business information systems, vol. 3, no. 2, 2008, 9-31.

[41] M. Goeken, Entwicklung von Data-Warehouse-Systemen. Anforderungsmanagement, Modellierung, Implementierung, Deutscher Universitäts-Verlag, Wiesbaden: 2006.

[42] M. Bühner, Einführung in die Test- und Fragebogenkonstruktion, Pearson, München: 2006.

[43] T. Kohlborn, A. Korthaus, T. Chan, M. Rosemann, "Identification and Analysis of Business and Software Services - A Consolidated Approach," IEEE Transactions on Services Computing, vol. 2, no. 1, 2009, 50-64.

[44] F. Kohlmann, R. Alt, "Vernetzung durch Serviceorientierung," in Transformation von Banken, R. Alt, B. Bernet, T. Zerndt, Editors, Berlin Heidelberg, 2009, 99-128.

4.5 Zwischenfazit

Eine Meta-Methode zur Konstruktion von Serviceidentifikationsmethoden basiert in erster Linie auf den identifizierten Situationen, die zu einer Auswahl bestimmter Fragmente führen. Diese Situationen beschreiben die Kombination von Ausprägungen relevanter Kontextfaktoren und SOA-Implementierungszielen. Darüber hinaus können aber auch sehr viel konkretere Ziele im Rahmen einer Serviceidentifikation berücksichtigt werden. Die drei Artikel dieses Gebiets zeigen, wie konkrete Ergebnisse geliefert und bestimmte Aspekte berücksichtigt werden können.

Methodenelemente wie Techniken, Rollen und Ergebnisse werden in entsprechenden Fragmenten zusammengesetzt und gezielt für die Erreichung dieser Ziele genutzt. Spezielle Techniken wie der Einsatz eigens entworfener Fragebögen sowie neue Rollen wie die bereits erwähnte „Service Design Unit" sind Bestandteil dieser Methoden. Auch die durch ein Framework gestalteten Service Maps stellen als Ergebnisse eine Art von Methodenelement dar. Auf diese Weise können vielfältige fachliche Aspekte in den Prozess der Serviceidentifikation integriert werden.

5. Gebiet IV: Kontextfaktoren in Serviceidentifikationsprojekten

5.1 Einleitung

Kontextfaktoren stellen neben den SOA-Implementierungszielen eine wichtige Komponente bei der Bestimmung von Situationen im Sinne des SME dar. Die entwickelte Meta-Methode bietet einen Rahmen für die Entwicklung von Methoden im Bereich der Serviceidentifikation. Daher müssen auch für diesen Bereich relevante Kontextfaktoren genutzt werden. Die Bestimmung der gleichen erfolgt auf Basis von zwei Fallstudien.

Die erste Fallstudie analysiert die Implementierung einer SOA und betrachtet die Benutzung von Methoden in einem Softwareentwicklungsprojekt. Sie ist ausführlich in Artikel 7 beschrieben. Artikel 8 befasst sich mit der Generalisierbarkeit dieser Fallstudienergebnisse und baut auf dem Arbeitsbericht auf. Die zweite Fallstudie, die in einem „Service Analysis and Design"-Projekt durchgeführt wurde, wird zum Abgleich und zur Ergänzung der Ergebnisse in Artikel 9 verwendet. Dadurch soll die Eignung der in der Meta-Methode verwendeten Kontextfaktoren sichergestellt werden.

5.2 Artikel 7: SOA Development and Service Identification – A Case Study on Method Use, Context and Success Factors

Veröffentlicht: Börner, R., Goeken, M., Rabhi, F. (2012): SOA Development and Service Identification – A Case Study on Method Use, Context and Success Factors, in: Frankfurt School - Working Paper Series, Nr. 189.

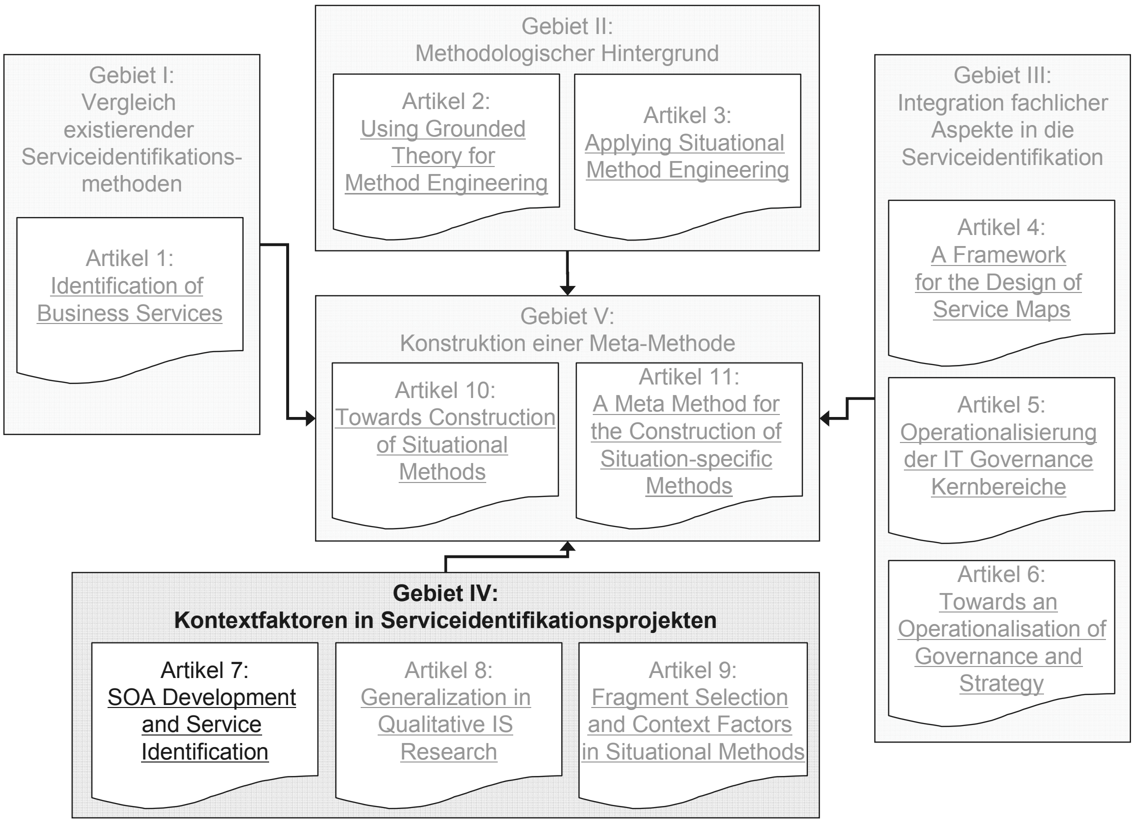

Abbildung 12: Artikel 7 im Kontext der kumulativen Dissertation

Frankfurt School – Working Paper Series

No. 189

SOA Development and Service Identification

A Case Study on Method Use, Context and Success Factors

by René Börner, Matthias Goeken and
Fethi Rabhi

April 2012

Abstract:

Although SOA development and service identification are widely discussed among academics and practitioners alike, little is known about how they are performed in practice. Thus, this paper investigates how SOA development and service identification work in real-life projects. An explorative single case study analysis is chosen as research methodology. It analyzes the ADAGE project, in which researchers implemented a service-oriented architecture in an Australian company. Furthermore, a situation-specific research process is developed. Elements of grounded theory and interpretative techniques are used to analyze interviews and documentations and to generalize the findings. Subsequently, 16 observations that describe SOA development and service identification in the ADAGE case are identified. Through generalization, these observations are transformed into hypotheses. In order to guide this generalization, both principles of interpretative field studies as well as abstraction mechanisms of conceptual modeling are utilized. The analysis of data and the generalization are accompanied by permanent comparison to discussions in related literature. A set of model fragments illustrating relationships of certain concepts and abstract categories are a major result of this case study. Aditionally, a number of opportunities for further research are outlined.

Keywords:

Service-oriented Architectures, Service Identification, Business Process Management, Web Services, Case Study Research

JEL classification: L86; M15; O32

ISSN: 14369753

Contact:

René Börner, Matthias Goeken
Management Department
Frankfurt School of Finance & Management,
60314 Frankfurt am Main, Germany

E-Mail: [r.boerner;m.goeken]@fs.de

Fethi Rabhi
School of Computer Science & Engineering
University of New South Wales
Sydney 2052, Australia

E-Mail: f.rabhi@unsw.edu.au

This article was published in Frankfurt am Main.

Content

1 Introduction..4

2 Methodology 5

2.1 The Appropriateness of Case Study Research..5
2.2 Research Process of Our Case Study...6

3 Case Description 11

3.1 Stakeholders 11
3.2 Scope and Goals 11
3.3 Proceedings...12

4 Case Analysis..17

4.1 Getting Started..17
4.2 Case Selection and data Collection 18
4.2.1 Documentation 18
4.2.2 Interviews with project participants 19
4.3 Analyzing Data..20
4.4 Generalization: Shaping Hypotheses...33
4.5 Integrating Hypotheses...43
4.5.1 Contingency Model (CM) 43
4.5.2 Model of Soft Factor Transition (MSFT)...44
4.5.3 Success Model (SM)...46
4.5.4 Emergent Method as a New Construct...47

5 Discussion and Further Research...48

5.1 Discussion of Findings and Further Research..48
5.2 Methodological Considerations – Discussion of the Research Process 49

6 Conclusion 51

References 52

Appendix 59

Appendix A: Interview Schedule 59
Appendix B: List of Websites 59
Appendix C: List of Services 59
Appendix D: Interview Guidelines for UNSW Researchers.....................................61

1 Introduction

Service-orientation is currently a dominating paradigm for enterprise and IT architectures, and a topical subject in research and practice alike. Service orientation and service-oriented architectures (SOA) promise a greater flexibility of IT and a faster adoption to changing business needs; furthermore it is expected that reusability of services reduces redundancies and subsequently saves costs for maintenance and operation of applications and software systems.

Despite the high expectations, empirical knowledge on SOA development as well as on service identification, which is a critical early phase in every SOA endeavour, is rare. This corresponds to the level of empirical knowledge in most areas of systems and software development (Jarke, 2009; Charters et al., 2009).

Although a number of approaches exist to support the development of SOA in general and service identification in particular (Birkmeier et al., 2009; Börner and Goeken, 2009; Thomas et al., 2010), little is known about the advantages and situational appropriateness of the different approaches. Similarly, little is known about how SOA projects work in practice, for example, whether they are guided by methods, whether they are business or IT driven etc. We assume that empirical knowledge on the way of working in real life projects is important to pursue both, exploratory, descriptive, or explanatory research goals on the one hand and design-oriented research on the other.

Theory and understanding of SOA development and service identification are currently not well developed. Therefore, an interpretative and qualitative research design was used, taking into account that our research goal was to explore and understand a single case in detail and to structure observations, to derive hypotheses, and to develop a small set of model fragments which can best be interpreted as middle range theories.

The paper is organized as follows: Following this introduction, in section 2 we present the underlying research approach by discussing the appropriateness of the case methodology for the study at hand and outlining the research process. After a general case description in section 3, observations, hypotheses and model fragments are discussed in detail to allow the reader to follow our argumentations and generalizations (section 4). Section 5 discusses the findings and limitations of our study, and outlines future research needs. Section 6 comprises a general conclusion with respect to the research questions.

2 Methodology

In this section, we discuss the appropriateness of the selected research method and our philosophical stance. We also present our research process and describe the activities in detail. The aim is to clarify the methodology to reveal our underlying assumptions and to be compliant with proposed guidelines for the conduct of case study research (Atkins & Sampson 2002, Klein & Myers 1999).

2.1 The Appropriateness of Case Study Research

Even though case studies are widely accepted research methods in information systems, there is no common definition (Myers, 2009). For instance, case study research cannot be easily distinguished from field study research. Darke et al. (1998) point out that “the case researcher has less prior knowledge of constructs and variables” (p. 275), whereas Eisenhardt (1989) argues that the specification of a priori constructs is valuable for building theory from case study research. Klein and Myers (1999) instead imply that field studies include in-depth case studies. Hence, we choose a broad definition which is frequently used: “A case study is an empirical inquiry that investigates a … phenomenon within its real-life context.” (Yin, 2003) (p. 41)

Despite the popularity of SOA, there is only little understanding of how to convey all advantages frequently mentioned in related literature. Moreover, little is known about methodological approaches in SOA development and how it takes place in practice. This corresponds to a low uptake of empirical research in systems and software development in general (Jarke, 2009).

Against this backdrop, we believe that qualitative case study research can make a useful contribution. Case studies are particularly relevant for research in its “early, formative stages” (Benbasat et al., 1987; Myers, 2009) which applies to the field of SOA (see also Luthria and Rabhi (2009) and Stebbins (2001)). As case studies can be descriptive and explorative in nature, they are supposed to give insights into how SOA development is performed. Furthermore, Benbasat et al. (1987) state that case study research is particularly appropriate for the study of information systems development, implementation, and use within organizations. Since we look at the development and implementation of an SOA, case study research is appropriate to investigate proceedings and reasons for the application of methods (see also Orlikowski (1993)).

The goal of our research is to understand and to explore how SOA development in its early stages and service identification in particular are performed in real-life projects. Due to this explorative objective, we consider case study research to be an appropriate method because a contemporary phenomenon is to be studied in its natural context (Yin, 2003), and we want to “provide descriptions of phenomena” (Darke et al., 1998). Descriptions and explanations of why a phenomenon occurs are provided by giving insight into the “generative mechanisms at work” (Walsham, 1995) (p. 79) observed within the case data.

With respect to our research goal, we derive hypotheses from the observations that have been made. Thus, we take the first steps in developing theory. The third research goal pointed out by Darke et al. (1998) – apart from providing descriptions and developing theory – is the testing of hypotheses and theory which is not envisaged in this research.

Many authors criticize single case studies for several reasons. For instance, Lee (1989) outlines four problems of single case studies, namely making controlled observations, making controlled deductions, allowing for replicability, and allowing for generalizability. However, according to Darke et al. (1998), a single case study may provide the basis for developing explanations of why a phenomenon occurs. Accordingly, Walsham (1995) indicates that single cases allow researchers to investigate phenomena in depth to provide rich descriptions and understanding. A single case study is appropriate when it represents i) a critical case (it meets all the necessary conditions for testing a theory), ii) an extreme or unique case, iii) a representative or typical case, or iv) a revelatory case (Yin, 2003).

Our case study, referred to as the ADAGE project, is certainly an extreme case since project members did not explicitly apply certain work methods and since documentation or requirements were not provided in written form. Most of the information was communicated verbally, and many process steps were chosen intuitively with methods being implicit in the best case. Furthermore, the project's principal was not aware of business processes, and the concept of services was virtually unknown at the beginning of the project. Even though we regard ADAGE as an extreme case, it is not unique. The absence of method use is frequently encountered in system development projects and represents one of the main reasons for failure (Standish Group International, 1994). Additionally, it is often the case that project participants not only develop a new system but that they are also confronted with a new technical and conceptual environment. Therefore, we also consider ADAGE an interesting case to improve the understanding of this phenomenon.

We chose an interpretative stance for the current research. By interpreting individual viewpoints of relevant stakeholders/actors and moving back and forth between the collected data and theoretical hypotheses, we tried to develop a picture of the whole. In doing so, we considered the details "in light of a larger sense of the whole." (Boland et al., 2010) (p. 4) This should improve the understanding of the subject matter and help to develop a consistent set of hypotheses (Walsham, 2006; Klein and Myers, 1999). We also acknowledge that the observations made in the field cannot be completely value-neutral. Instead, we assume the relativity of our own position. We therefore attach importance to the transparency and traceability of the process and to the justification of our observations and derivation of hypotheses.

2.2 Research Process of Our Case Study

Following Eisenhardt (1989), we designed a research process for the study at hand which is depicted in Figure 1. The process commenced with the explication of "a priori constructs" (see section 2). These a priori constructs also help to guide the case analysis. However, they must not be confused with observations or hypotheses which are the results of the analysis performed later on. Furthermore, the **first activity** included creating an initial research question and more specific questions for investigation (see section 4.1). This is an important first step to guide the case analysis and focus efforts.

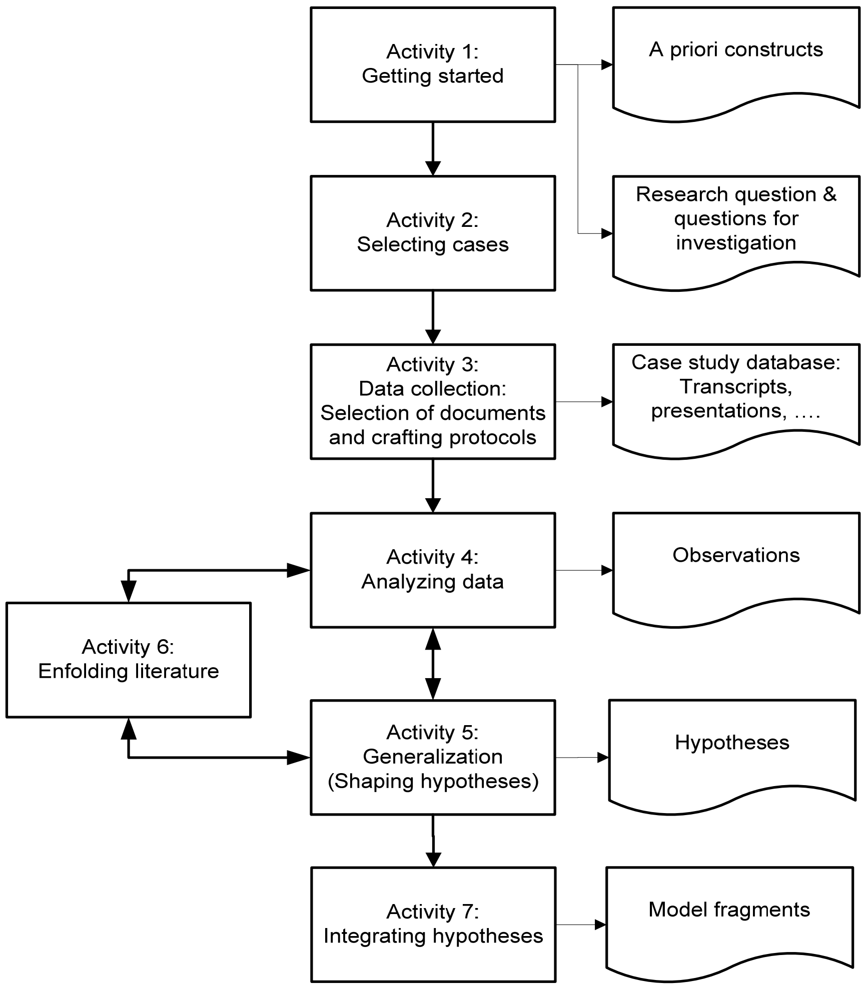

Figure 1: Activities and Deliverables of our Research Process

In a **second activity**, an appropriate case had to be selected. Since the objective of this case study was to enhance understanding and develop theory rather than testing it, the chosen case did not have to be representative. A theoretical sampling though is "particularly suitable for illuminating and extending relationships and logic among constructs" (Eisenhardt and Graebner, 2007) (p.27). Hence, we intentionally chose an extreme case as described earlier in this section.

Multiple data collection methods such as interviews and analysis of documentation were used in the **third activity** (see section 4.2). In order to have a solid and profound basis to build upon, collected data had to be gathered from multiple sources of evidence. Using different sources serves to underpin the completeness and correctness of data. Inconsistencies can be made obvious through the use of various sources. This triangulation of data is important for the reliability of the case study's outcomes (Yin, 2003). Flexible and opportunistic data collection methods allow for reacting to emergent themes by adjusting data collection when necessary (Eisenhardt, 1989). A case study database was created to make results and conclusions comprehensible for the reader. First, primary data such as project progress reports, presentations as well as published and unpublished papers were stored and listed. Second, reports of the investigation itself were provided. Interview schedules, a list of project members, annotated bibliographies, audio records of the interviews, and interview transcripts are examples for this second class of documents in the database.

The **fourth activity** – analyzing data – began with the description of the project which was subject of our case study (see section 3) and continued with the grounding of our observations, as described in section 4.3. This was followed by the shaping of hypotheses through generalization, which is the subject of section 4.4. In the presentation of our research, we distinguish between observations and hypotheses (analyzing data and generalization, respectively) in order to enhance transparency, traceability, and reliability of the procedure, even though they are closely interwoven.

Activity four was dedicated to properly grounding concepts and patterns by triangulation. Construct validity was supported by using multiple sources of evidence (Eisenhardt, 1989). The identification of concepts and patterns was conducted by employing techniques from grounded theory, for example, open and axial coding (Strauss and Corbin, 1990), and interpretative techniques (see Walsham (1995) and Boland et al. (2010) for hermeneutical exegesis in IS). Even though we did not use the coding techniques to their fullest extent, the general approach and respective tools supported the assignment of statements from the interviews and documents to concepts. The goal was to detect relevant particulars within the idiographic details and to identify relevant concepts as well as "the "why" of what is happening" (Eisenhardt, 1989) (p. 542). Examining the texts and interviews in their contexts – for example, by answering questions like 'who wrote the text, for what purpose, and what is the state of the project at the moment of writing' – allowed for deeper interpretations of the meaning of statements and comments (Myers, 2009). In this respect, interpretative techniques and coding helped to describe the generative mechanisms at work and to discover the underlying reasons for why a relationship or pattern exists.

This activity resulted in observations which are judgments or inferences from what we observed in the field and for what we found deeper grounding in the data and in related literature (adapted from Merriam-Webster). These judgments/inferences are a first abstraction from this data, the "first-order concepts" as Van Maanen (1979) labels them. Second-order concepts (analogous to the concepts forming our observations) are "notions used by the fieldworker to explain the patterning of the first-order data" (Van Maanen, 1979) (p. 451).

While concepts and observations are closely linked to the idiographic details, we generated hypotheses consisting of abstract categories in **activity five** (generalization). Extending Van Maanen's terminology (1979), we might label these hypotheses third-order concepts, which thus represent abstractions of abstractions of first-order data. We performed this by relating concepts described in the observations to categories applying interpretative techniques and principles of interpretative field studies (Klein and Myers, 1999) as well as abstraction mechanisms transferred from conceptual modeling. The categories and relationships between them should apply to multiple situations and hereby reach a certain level of generality.

Lee and Baskerville (2003) cast doubt on the generalizability of empirical descriptions in a case study beyond this given case. Nevertheless, they acknowledge that other researchers allow for generalization from empirical statements to theoretical statements (e.g., Walsham (1995)). Similarly, Klein and Myers (1999) recommend generalizations if they are “carefully related to the case study details … so readers can follow how the researcher arrived at his or her theoretical insights” (p. 75).

In order to prevent excessive overestimation of the generalizability, the resulting hypotheses formed by categories and relationships should be viewed as tendencies (Walsham, 1995). The introduction of abstraction mechanisms of conceptual modeling (section 4.4) extends existing research techniques. Our intention was to make “controlled deductions” (Lee, 1989) and thus improve reliability and make the development of hypotheses traceable and transparent to the reader, thus relating hypotheses and observations carefully.

The **sixth activity** in the research process illustrated in Figure 1 consisted of a comparison with related – conflicting or similar – work, so called “enfolding literature.” It aims at comparing opinions and positions found in related literature with the observations we made and improving the hypotheses. This results in hypotheses “with stronger internal validity, wider generalizability, and higher conceptual level.” (Eisenhardt, 1989) (p. 544). Analyzing data (4) and generalization (5) are closely interwoven and both supported by the analysis of literature. According to Gadamer (1976), “the harmony of all the details with the whole is the criterion of correct understanding” (p.117), and in a number of iterations “a complex whole of shared meaning emerges” (Klein and Myers, 1999) (p. 71). This is represented by the iterative layout of activities four to six. The focus is on constantly comparing hypotheses and data, iterating towards a set of hypotheses which closely fits the data (Eisenhardt, 1989).

Finally, **activity seven** integrated several hypotheses to build model fragments (section 4.5). Through another generalization, these fragments have an even higher level of abstraction and are thus more remote from the first-order data. Figure 2 illustrates the increase of abstraction with each activity performed moving from left to right. Considering semantic levels, model fragments encompass both the level of instances and the level of types. They incorporate categories used in our hypotheses but also include concepts – including those we were not able to observe in our case study.

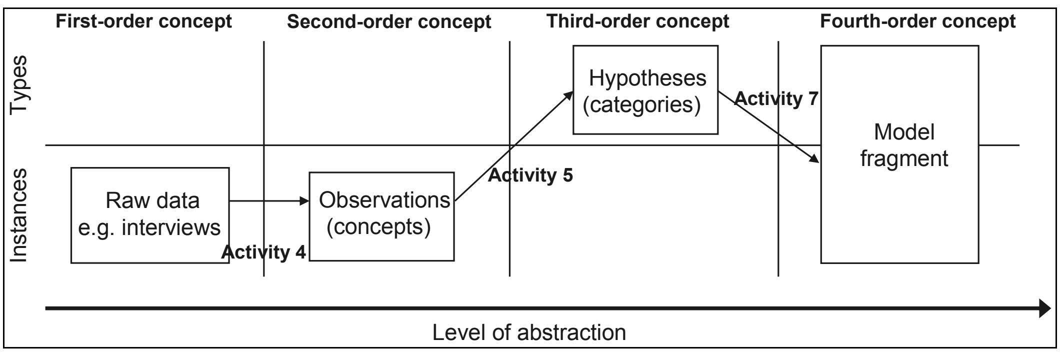

Figure 2: Abstraction and Semantic Level

We consider the resulting model fragments to be middle range theories as they are of limited conceptual range and to be intermediate to the hypotheses derived from the observations and the total conceptual structures of “grand theories“. At this point our notion of theory is adopted from Gregor's classification (Gregor, 2006). While the set of interconnected categories - forming the hypotheses - are "statements providing a lens for viewing or explaining the world", the middle range model fragments are "statements of relationships among constructs" (p. 613) in terms of generative mechanisms.

3 Case Description

3.1 Stakeholders

"Ad-hoc DAta Grids Environments" (ADAGE) is a collaborative research initiative between a university (UNSW) and an industry partner (SIRCA). It can be regarded as an umbrella for three sources of funding, so it aims to address differing motivations of its various stakeholders.

a) The first and most important part is the project "Efficient Management of Information Resources Over Ad-Hoc Data Grids," which is funded by the Australian Department of Education, Science and Training (DEST) and its successor the Department of Innovation, Industry, Science and Research, respectively. Funds were granted for a three year period from January 2007 until March 2010.

b) The second part of ADAGE was SIRCA's interest to use its huge data repository to better suit their customers' requirements. Due to the successful prototyping of services, SIRCA's support for ADAGE has been extended until June 2010.

c) Finally, the UNSW contributed to the project, for instance by providing a PhD student position to support research in the field of service-oriented architectures.

3.2 Scope and Goals

In the context of the ADAGE project, researchers at the UNSW implemented a service-oriented architecture for SIRCA. "The project aims to investigate how to efficiently gather, store, query, and manage ad-hoc data grids from both a manager and an end-user perspective, benefiting various applications including in e-science and e-research." (Department of Innovation, 2010) Its objective is to provide researchers a platform for easier retrieval and analysis of heterogeneous data from different sources (grid environment) spontaneously in an unforeseeable fashion (ad-hoc). Neither business processes nor SOAs were the selling point of the project. The former were hardly considered at all, whereas the latter was chosen as the preferred architectural paradigm of this project. Services (and their identification) were used as a means to meet SIRCA's requirements rather than being the subject of analysis themselves.

SIRCA provides a huge data repository containing historical news and financial market data, such as quotes, trades, and market depth data. It normalizes data and makes it accessible. Their aim is to supply this data to researchers worldwide, but so far their academic customers are mostly Australian and New Zealand universities. SIRCA also makes the same data available to industry customers through Thomson Reuters. Thus, their business model is quite simple and is covered by one business process only.

SIRCA was not familiar with the idea of business processes management, so that no model was delivered that could have been analyzed in the course of service identification. However, SIRCA's management had some requirements in mind that had to be fulfilled by services. Unfortunately, the former were not documented which makes traceability difficult. Requirements were communicated to the project team in scheduled weekly meetings and workshops. Service candidates were identified on the basis of these meetings and subsequently prototyped. In an iterative approach, the functionality of these candidates was adjusted to finally meet SIRCA's requirements. In some cases, they were completely dismissed and new ones had to be created. A close collaboration between SIRCA's research and development department and UNSW's project team was critical to ensuring the successful identification of services.

SIRCA's management did not aim at the implementation of an SOA in particular. The idea of services was basically advocated by UNSW's project team. SIRCA had been using web services in specific areas, such as enabling remote access to its data. However, know-how as far as SOAs are concerned was limited to the service technology level. At the outset of the project, funding for three years was provided. Towards the end of the project, funds for further six months were approved to implement the prototype and make it accessible to SIRCA's customers.

In ADAGE, services were created based on the availability of data. This led to a technical understanding of services which is, for instance, reflected by the synonymous use of the terms "service" and "web service" by project team members. Hence, the scope of service identification in this project was limited to software services.

First and foremost, the search for services was driven by the idea to retrieve and integrate data from different sources. In a second step, project members developed ideas about the types of services that could support researchers in analyzing data. This included, for example, building time series of financial data, merging data from different sources, and visualizing events. The identified and implemented services were offered to third parties, thus supporting their business processes, only after the completion of these steps. Clearly, this was a requirements-driven bottom-up approach. Goals included the provision on a graphical user interface (GUI) to customers, enabling the latter to directly invoke services in an ad-hoc fashion to analyze news and financial market data. This implies a distinct degree of customer interaction which influences the identification of services significantly.

3.3 Proceedings

The official project start was in January 2007, after funds were secured in September 2006. As described previously, there were manifold motivations for all involved parties to cooperate with one another. On the one hand, there was SIRCA's huge Thomson Reuters archive and its need for innovation. On the other hand, more complex demands from a wide range of user groups (e.g., Thomson Reuters, academics, etc.) had to be satisfied. In order to achieve this, the expertise offered by UNSW researchers, working in different groups across faculties, was to be utilized. The three following research groups contributed to the project:

a) The Finance IT Research Group of the School of Information Systems, Technology and Management provided know-how concerning financial information systems. The group had already conducted a number of capital markets projects between 2001 and 2004.

b) The Service-Oriented Computing (SOC) Group of the School of Computer Science and Engineering had plenty of experience in databases, information processing, and web technologies. It had been involved in major international SOC initiatives linked to business partners, such as IBM, HP, and SAP.

c) The Bibliometric and Infometric Research Group (BIRG) of the School of Information Systems, Technology and Management contributed through their experience in developing metrics related to document management and information services.

Before the project's official start, the following collection of initiatives (see Table 1) with respective responsibilities was presented in December 2006. A major objective of these initiatives was to capture domain knowledge in the different fields of cooperation in order to use existing data in a meaningful way.

Initiatives	**Responsibilities**
Requirements analysis of news systems	Finance IT Group
Technical overview of Reuters data	Finance IT Group
Conceptual modelling of financial data	Finance IT Group
Story reconstruction and news retrieval	BIRG Group
Metadata analysis	BIRG/Finance IT Group
Multidimensional navigation of news data	SOC Group
Summarization	SOC Group

Table 1: Initiatives as of December 2006

After the project start in January 2007, some of the initiatives (pilot projects) shown in Table 1 were mapped to project milestones that are illustrated in Figure 3. Each milestone had a manager, project teams, and objectives, respectively.

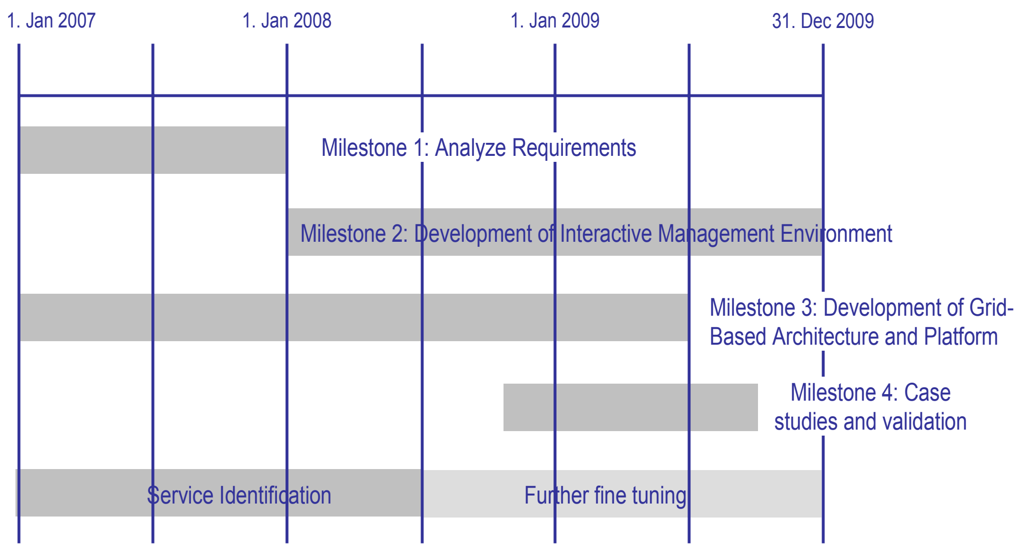

Figure 3: Milestones of the ADAGE Project

For the purpose of this paper, the focus will be on milestones one and three. The first goal of milestone one was the definition of data grid requirements in the context of ad-hoc data currently managed by SIRCA. Therefore, a review and assessment of existing ad-hoc data grid processing approaches was planned. Existing data formats had to be investigated and sample data repositories were created. Existing grid architectural models, tools, and platforms had to be analysed in milestone three. An architecture for processing queries across a large number of ad-hoc sources had to be defined and a respective prototype had to be developed. Regular monthly meetings with SIRCA were complemented by technical meetings based on the needs occurring in the project. Milestone four consisted of the news summarization case and the financial market data analysis case to prove applicability of the developed services.

Modeling and understanding of news and market data was a central activity in the first half of 2007. The identification and definition of possible services for the architecture to be developed was a major task and part of the Finance IT Research Group's responsibilities. By July 2007, project members had gained an insight into SIRCA's datasets and identified key shared user requirements. Since SIRCA could not provide documentation such as entity relationship models (ERM), it was basically left to the project team to use their full access to SIRCA's data to explore its structure.[1]

In October 2007, an event-based data model had been developed (Rabhi et al., 2009a). In the context of the SOA for processing e-market data, a couple of query and analysis services had been implemented. Services for news and market data as well as for data mining and statistical analysis had yet to be developed. Business processes and their corresponding workflows and end-user composition belonged to the work in process. By the end of 2007, communities of users that might benefit from the project were identified for use in cases one and two (finance and business intelligence, respectively).

Several financial data analysis processes had been studied exemplarily by September 2008 in order to capture specialist (domain) knowledge. Thus, requirements for tools and components could be defined and fulfilled. Besides implementing these requirements, the focus was on reuse of components and facilitation of a user's ability to design processes. In order to achieve both objectives, different levels of granularity and complexity were addressed. Easy to use tools allow end users to invoke elementary services and perform simple operations on datasets. More complex processes can be composed by users themselves. This is done through a drag and drop GUI (see Figure 4).

[1] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

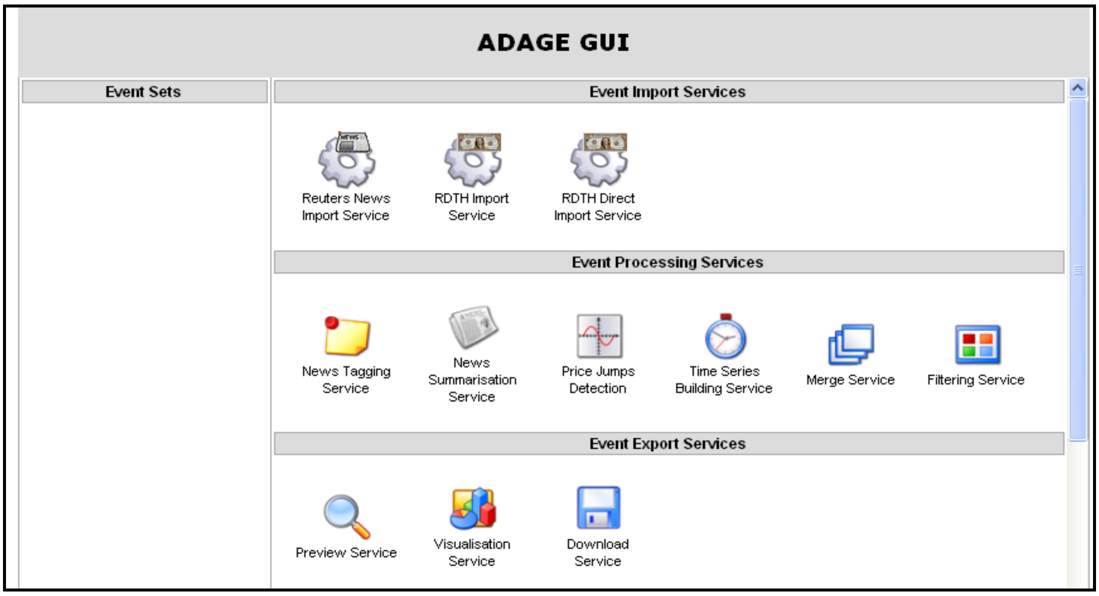

Figure 4: Graphical User Interface of ADAGE

The GUI empowers users to manually compose services to support a process at hand. Instead of letting users do this job, an automated composition of services is also possible. Workflow management systems facilitate enacting business processes as compositions of existing elementary services. Although such a tool (Triana) was trialed by Prof. Omer Rana's group at Cardiff University (Rabhi et al., 2009b), it was not used any further in the ADAGE project. Instead, it was deemed much more effective if users were allowed to orchestrate services themselves.

By the end of 2008, the project had yielded several outcomes. A core element and basis for all other results was the data model developed by Rabhi et al. (2009a). Moreover, a service-oriented architecture and a composition framework were designed. A new technique for time-series analysis which operates on financial market data had been developed (Guabtni et al., 2010; Rabhi et al., 2010). An additional outcome was the enhancement of existing clustering and ontology-driven summarization techniques for the analysis of time-sequenced unstructured data such as news (Pham et al., 2009).

It was not until January 2009 that process design approaches, such as Business Process Modeling, were taken into consideration. Although specialist knowledge had been captured earlier, a systematic process modeling exercise occurred for the first time. The service identification then concentrated on reusability and the facilitation of a user's ability to design services. Due to the way users analyzed data, business modeling (and thus service composition) could not be static but had to be rather dynamic. Furthermore, it is assumed that the service development process and service identification cannot be centralized. These activities should be organically driven, i.e., there cannot be one central instance (like the ADAGE project) which can, for example, provide services for data extraction from all conceivable data sources. Thus, the ADAGE SOA can only provide a framework which hopes to trigger other developers to publish their own services that enable an extraction of data from their respective sources (for meta design, see Fischer et al. (2009)). The data model is used as a solution to match business requirements with technical web services.

The ADAGE SOA was designed according to user-driven requirements, allowing execution of processes to be piloted by users at any stage of their execution. Services are therefore categorized into:

a) Event Import Services: These services provide market and news events. They are used for querying, filtering, and mapping of data. Extraction of data from databases (e.g., from Reuters Datascope Tick History) is done by Event Import Services.

b) Event Processing Services: Services in this category are used to process financial market and news events. For instance, the sampling period and financial measures that the user wants incorporated in an analysis can be selected.

c) Event Export Services: These services allow for the extraction of information from financial market and news events analyses. Visualization is one option that enables the user to explore patterns within the data. Another possibility (Download Service) is to save (export) the information in various file formats.

Milestone four in the project utilized a case study to ensure the usefulness and applicability of this categorization. A comprehensive overview of all services implemented in the course of the ADAGE project can be found in Appendix C. Although specialist knowledge was captured and example processes were analyzed, a technical viewpoint was taken in the first place. A major challenge was whether services could be easily composed by users on an ad-hoc basis. The DEST project officially ended on 31st March 2010.

4 Case Analysis

4.1 Getting Started

As illustrated in Figure 1, outlining the theoretical background in the sense of "a priori constructs" was the first step of our research process. Since the research question is the foundation for the analysis of every case study (Eisenhardt, 1989), our research process continued with the formulation of a research question. As outlined in section 2, the identification of services is recognized as one of the most important steps while implementing an SOA. Shortly after the beginning of the case study, it became obvious that a single focus on service identification might not deliver satisfactory results in the ADAGE case. Thus, we extended the scope of our analysis to the early stages of the SOA lifecycle, namely the design and development of services and the service-oriented architecture in ADAGE (see Figure 5). Additionally, the framework presented by Börner and Goeken (2009) provides a number of criteria that will be used in the following.

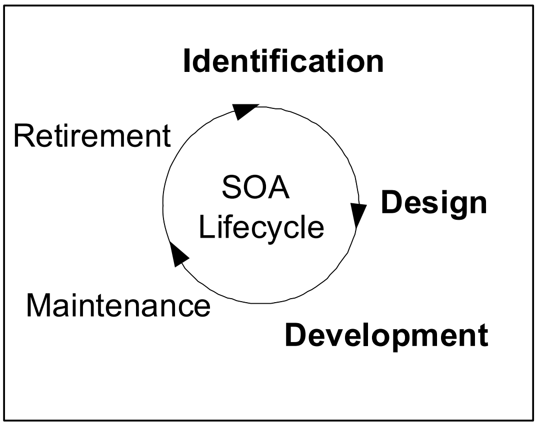

Figure 5: SOA Lifecycle

Some authors have already developed methods to support identification, design, and development of services. However, little is known about service identification and the early stages of the SOA lifecycle in real-life projects. Thus, in order to guide us through the analysis, we formulated the following initial research question:

How does SOA development and service identification work in real-life projects?

The rather generic nature of our research question necessitated the formulation of further, more concrete questions, the *questions of investigation*. These helped to focus our analysis and support, for example, the hermeneutic analysis of the interview transcripts. The questions of investigation were reflected by the interview guidelines. To provide an example, we attached the interview guidelines for the UNSW researchers in Appendix D.

It is widely accepted that method use improves the outcome of system development efforts (Avison and Fitzgerald, 2008; Sommerville, 2006; Lehman and Ramil, 2002). We therefore rate method use as an important aspect of SOA development and service identification in real-life projects. Hence, our first question of investigation was: *Are methods used at all and if yes, which ones?* This leads directly to another interesting aspect: *Which consequences does method use (or its absence) have for an SOA?*

Our investigation into the ADAGE project aimed at capturing the setting, i.e., all preconditions and circumstances, as completely as possible. Such external factors, so called “context factors,” are commonly considered to play an important role in software development projects. Thus, we tried to find out: *Which circumstances influence the proceedings of the service identification process?*

To examine the social context in more detail, we explored the roles of the different people within the project, their skills and knowledge as well as the evolution of skills and knowledge. One focus of our investigation was therefore: *What is the stakeholder’s understanding of services? Did this understanding change/develop in the course of the project? Was there a change of the skill level of the project participants?*

A fourth area of investigation dealt with technical aspects which might be related to the aforementioned methodological, contextual, and skill-/knowledge-related aspects. It comprised design decisions made during the project and design principles the project team applied. *How was the SOA implemented technically? Which software development techniques had been used?*

Furthermore, we considered the outcome of the project as important for investigation. We did not apply a catalogue of predefined success measures but wanted to capture how stakeholders assessed and evaluated the outcome of ADAGE. *Was the project a success? What criteria/measures were (implicitly) used to evaluate the outcome?*

4.2 Case Selection and Data Collection

Having these questions in mind, the relevant sources of data used in this study are now described in more detail.

4.2.1 Documentation

Major sources of documentation about project proceedings are 19 presentations that were held between December 2006 and November 2009. With only one exception, these presentations were compiled by a professor of the UNSW who acted as principal investigator and were presented in meetings with SIRCA. All of them provide a good overview of the general proceeding, achieved objectives, and next steps required at the time of the presentation. The only presentation made for a different purpose used the project as a case for a user-driven SOA for financial market analysis and took a slightly different point of view.

Furthermore, official progress reports give a structured overview of how a project is proceeding. Three progress reports were delivered after the first, second, and third year of the project, respectively. They addressed the Australian Department of Education Science and Training (DEST) and its successor, the Department of Innovation, Industry, Science and Research, which were responsible for funding the ADAGE project. Due to their much more formal character (compared to the previously described presentations), these reports offer less insights into actual work practices, such as the identification of services. As far as funding is concerned, a Statement of Income and Expenditure is also part of the yearly progress reports.

A considerable number of published and unpublished papers were authored by project team members in the course of the project. These papers encompass many intermediate results of the project and highlight different aspects of ADAGE. The variety of publications in which the papers have appeared shows the scientific and practical value of the project’s outcomes.

Additionally, various websites about the project or involved parties were used to retrieve background information. Of course, the graphical user interface (GUI), one of the important results of the ADAGE project, was itself accessible on the internet. Appendix B contains a list of the most important websites used for the case study.

Some newspaper articles about the project could be found as well and were added to the case study database for a matter of completeness. Information given in these articles was found to contribute only little to the questions of the case study.

4.2.2 Interviews with Project Participants

In order to retrieve additional information, particularly about the identification of services in the ADAGE project, four interviews were conducted. Three of the interviewees were project team members at the UNSW. In the following, they will be referred to according to their roles as principal investigator, second investigator and postdoctoral fellow. All of them were involved in identifying and designing services in the project. The interviews were conducted on 4th and 5th March, 2010 at the UNSW (see Appendix A for detailed schedule). Another interviewee was a representative of SIRCA who was interviewed on 9th March at SIRCA's premises on the 9th floor of 80 Clarence Street in Sydney's Central Business District. He will be referred to as industry coordinator in the following.

Before the interviews, a short introduction about the interviewer himself, his field of research, the purpose of the interview, and usage of the gathered data was given to the interviewees. All interviewees consented to the recording of the interviews. Thus, the case study database includes digital audio files of all interviews as well as complete transcripts for further analysis. The interviews were open-ended. However, guidelines with questions were used in each case. For the three project team members at the UNSW, the same question guidelines applied (Appendix D), whereas a different guideline was used in the interview with SIRCA's representative. The interview transcripts served as an important basis for the analysis of the service identification process.

The data collection procedure allowed for the accumulation of evidence from diverse sources. In the presentation of the results, we are going to combine details from different sources in order to triangulate them, to investigate the subject matter from different perspectives, and, in doing so, to provide a deeper grounding of our observations.

4.3 Analyzing Data

The aforementioned sources of evidence were analyzed thoroughly and led to a number of observations that will be outlined in this section. Through this analysis, we arrived at several concepts that we consider to be important for the success of the ADAGE project. Grounded in document studies and interviews, 16 observations – in terms of judgments from what was perceived in the data – describe these concepts and their relationships. According to the contextualization principle presented by Klein & Myers (1999), we try to present the subject matter in its "context so that the intended audience can see how the current situation under investigation emerged" (p. 73). All observations directly refer to the ADAGE case study. The first column of Table 2 gives an overview of all identified concepts.

As mentioned before, an examination of "enfolding literature" supports the derivation of hypotheses. Therefore, the presentation and justification of our observations in this section will be accompanied by references to related literature. Some observations are new in the sense that they have not been discussed in related literature yet. Others confirm and reinforce assumptions that have already been made in this field of research, whereas a third type is contrary to common belief.

Observation 1: The generous funding significantly influenced the way of service identification.

If you have to deliver defined outcomes within a tight time frame and under pressure from a paying customer, the focus will usually be on the result, i.e., a service delivering exactly what was asked for rather than on the elaborate identification and design that adheres to some proper methodological approach. Thus, one might generally expect that commercial projects sometimes tend to seek "quick and dirty"-solutions just to make things work as fast as possible. On the other hand, in academic projects like ADAGE with a generous funding secured for three years, one might rather expect an extensive use of methods that satisfy academic standards. Interestingly, for many reasons, it was different in the ADAGE case. Although it was an academic project with limited economic pressure on participants, service identification was far from being a rigorous process. The following excerpt argues that the latter was rather intuitive *because* the project was academic in nature and was intended to deliver innovation.

"Very intuitively, yes. And also if we were a commercial project, then we (would) have a contract. You have to make exactly what the client wants, otherwise they can sue you. But because this was a research project, the main requirement wasn't to deliver a precise thing, it was to deliver innovation. And the innovation was to take this domain and do services. So, the services we built [were] more like example services rather than we-had-to-dos. That's probably the difference between what we have done and maybe if services were developed in a more sort of commercial environment where you have this contract, your obligations."[2]

Nevertheless, this statement shows that the academic environment in which ADAGE took place clearly influenced the way team members identified services – however, in a different way than originally assumed. Even in this kind of non-commercial project, the budget affects the identification and design of services concerning the effort that is put into the development.

"Somehow the cost it takes to develop a particular service should be factored into the methodology."[3]

The budget encompasses such resources as time and manpower which are directly linked to the costs of a project. Not only ADAGE project members consider the budget to be an important contextual factor for the choice of methods. Other authors such as Becker, Knackstedt, Pfeiffer, and Janiesch (2007) state that "appropriate methods for problem solving must be chosen, adapted, or designed depending on the specific characteristics of a situation, such as qualification, number of employees, or available time." (p. 1) Hence, they confirm this relationship.

---

[2] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

[3] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

Observation 2: SIRCA's small company size significantly influenced the way of service identification.

SIRCA itself can be categorized as a small or medium-sized enterprise (SME). In the words of one staff member:

"We are a very simple outfit, we are a data repository. We collect lots of data, we do fairly substandard processing to it to normalize it, and make it easily accessible."[4]

Asked whether analyzing financial data was their main process, he responded: "This is our reason to be."[5] Hence, there is only one business process that can be identified for SIRCA. Since there is no need for process landscapes or industry reference models that are commonly used by larger companies, there are fewer inputs to be incorporated in the service identification process. This confirms the generally held belief that company size is a considerable contextual factor in many kinds of software projects (see also Sedera (2008)).

Observation 3: People's skills significantly influenced the way of service identification.

People's skills had a direct influence on applied methods in the ADAGE case as the following explanation shows:

"Another thing which is important in our context is people's skills. Sometimes we had to do certain services because we had the people who had the skills to. (…) We didn't do the asset identification because we didn't have anyone who knew about that but if we had then, we would have done it. So you are constrained by the cost it takes to develop, but also you were constrained by people's skills and in our case that's a big big problem because with service-oriented architectures and technologies there are not many people who understand them well enough to develop them and that is a big constraint."[6]

Employees' skills are commonly accepted to play a major role in software development projects (Faraj and Sproull, 2000) and the choice of respective methods (Becker et al., 2007). On the one hand, available skills result in the application while a lack of such skills results in the omission of certain techniques, as described in the above statement. On the other hand, this leads to communication structures that can be bound to and depend upon single persons. This will be elaborated in more detail under Observation 6.

The significance of budget, company size, and employees' skills is visible in the ADAGE project and is echoed in related literature. Besides these contextual factors, Börner (2010) mentions the geographic scope of operations and the existence of a designated IT department within an organization (Anderson et al., 2005) as significant contextual factors that determine situations in the sense of situational method engineering. The latter could neither be confirmed nor denied by the case study at hand.

---

[4] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

[5] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

[6] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

Observation 4: The implemented SOA satisfied users although no formal method for the identification of services was used.

Throughout the presentations held by researchers of the UNSW during the project, the successful progress of the project can be traced. Tasks that are work-in-progress in one presentation can usually be found under achieved goals in one of the following presentations. If they do not reoccur explicitly, it can be assumed in most cases that they were successfully implemented. Additionally, the interviews conducted come to the same conclusion and thus support the assumption expressed in Observation 4. According to the interview guidelines (see also Appendix D), all interviewees were asked if the implementation of the SOA and respective services in the ADAGE project was successful. In a number of the following statements, interviewees often use the term "case study" for what is commonly referred to as a "use case." It must not be confused with the research methodology we applied.

"Based on the (…) three case studies, I think it's quite successful because for the three we achieved what we wanted."[7]

This first statement clearly stresses the researchers' point of view. The project team analyzed user requirements to describe use cases. Subsequently, they identified services to meet these requirements. By implementing these services, the project was a success from their perspective. Asked if SIRCA was satisfied with what they got, the answers from the project team were as follows:

"The only feedback we got from them is actually, they were very happy with the ad-hoc way of doing this processing. Because for them, it's like quite new to have these services and you just use them"[8]

So far, the statements have shown how researchers of the UNSW perceived the success of ADAGE. Of course, their answers are partially based on feedback they received from SIRCA's representatives, staff, or clients. However, it is important to consider a primary source, in this case the manager of SIRCA's research and development unit.

"It's been demonstrated to us on a number of occasions that various styles of development and the evolution has been satisfactory for us, and the services that we have seen demonstrated what can be done and enable us to think about how we might incorporate things into our standard offerings which we give to people in a non-service-based environment at this point."[9]

The progress shown in several documentations accounts for the project's success. In the excerpts above, interviewees often state that customers were "happy" with what they got, so we can assume that customers were satisfied. Apart from the business intelligence use case of ADAGE where "the system has not been used by anybody"[10], the interviews support the assumption that the project can be considered successful.

---

[7] Personal Communication, Interview with the postdoctoral fellow (UNSW), 05.03.2010

[8] Personal Communication, Interview with the postdoctoral fellow (UNSW), 05.03.2010

[9] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

[10] Personal Communication, Interview with the second investigator (UNSW), 04.03.2010

As far as method use is concerned, the presentations and progress reports that were analyzed provide only little information about the actual service identification process. These documents describe primarily results rather than procedures. However, many insights were gained from the interviews that were conducted. The first obvious result is that actually "there was no methodology"[11] used for the identification of services, i.e., at least no explicit method was used. The approach used by the project team was indeed rather unstructured. Basically, team members analyzed data structures, observed users' behavior where possible ("I think the method should have been a kind of observing, how finance people use tools."[12]), and talked to SIRCA's staff. Of course, there must have been some way of identifying services, so we can assume that some kind of method was implicitly used at one or the other point of the project. Statements like the following show this implicit application of methods:

"So, it's actually case study based. This is not the best method but it's one of the methods."[13]

Other interviewees said the identification "was based on experience we [the project team] applied,"[14] and admitted that they had "done it a bit in an archaic way. It was a step-by-step approach where we could identify services."[15]

Why was ADAGE successful although no service identification methods were used? On the one hand, the limited complexity of SIRCA's business processes made the identification of services quite simple.

"Our processes are fairly, you know, atomistic. (…) then people access the data with some fairly straight-forward enterprise in that regard."[16]

On the other hand, interaction and communication between SIRCA and UNSW's project team were intense. The latter aspect will be considered in more detail under Observation 6. Overall, not only researchers at the UNSW but also SIRCA seemed to be convinced of the project's success. Actually, SIRCA extended the funding of the project for further three months to implement the ADAGE services in their IT infrastructure and offer them to their customers.

Related literature generally recommends the application of methods. Vessey and Glass (1998), for example, argue that the use of formal software methods leads to significant gains in software productivity under some circumstances. At least for our case, we can conclude that a method for service identification was not necessary for the success of the ADAGE project which contradicts the assumption made in literature.

---

[11] Personal Communication, Interview with the second investigator (UNSW), 04.03.2010

[12] Personal Communication, Interview with the postdoctoral fellow (UNSW), 05.03.2010

[13] Personal Communication, Interview with the postdoctoral fellow (UNSW), 05.03.2010

[14] Personal Communication, Interview with the second investigator (UNSW), 04.03.2010

[15] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

[16] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

Observation 5: Little SOA experience led to a technical understanding of services.

The project primarily dealt with the ad-hoc usage of data from multiple sources to provide users with easier access to data and support them in the analysis of this data. At the outset of the project, it was not exactly clear how user requirements were supposed to be implemented. The implementation of services in the context of an SOA thus presented one possibility for a target infrastructure.

"SIRCA wasn't aware of the concept of services, so we brought the concept of services. (…) In a way we succeeded because now SIRCA thinks in terms of services and looks at their offers in terms of services."[17]

On SIRCA's side, no one had thought about services before. Although a deeper knowledge of the actual implementation of an IT system should not be the principal's concern in the first place, a common understanding of how services can contribute to fulfill user requirements certainly helped to make the project a success. The idea of services was used to capture user requirements in a structured way.

"So we haven't been given descriptions of services but we suggested to SIRCA that the use of services could be to gather the most common functions that people do."[18]

Even on the UNSW's side, not many people were familiar with the SOA concept. This in turn focused large parts of the communication on the project leader. In later stages of the project, more team members became familiar with services and the latter became a central point of ADAGE. In a presentation dated 24th September, 2008, "Learning/understanding/modeling" associated with services was explicitly part of the project agenda. The structure of the graphical user interface[19] clearly shows the service-oriented mindset behind the implemented software. Nonetheless, both statements show that thinking in terms of services is not intuitive and has to be learned by all involved stakeholders. This assumption is largely confirmed by literature. Many companies are still function-oriented (Vergidis et al., 2008), and many employees only consider individual tasks with a limited scope in their daily work. In order to implement successful service-oriented architectures, they will have to get used to the concept of services.

Observation 6: Personal communication was vital to meet user requirements.

According to presentation slides dated February 2007, team members of the UNSW and representatives of SIRCA arranged monthly meetings. Additionally, they agreed on technical meetings whenever a need would arise. These frequent meetings with SIRCA were necessary to retrieve information that was not documented anywhere, to understand the systems they used, to access their technical infrastructure, and for psychological reasons, as the following statement shows.

"This project could never have been done the way it's been done if I wasn't going there on a weekly basis. So I had to go there, I was talking to them. Simply because there

[17] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

[18] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

[19] http://129.94.172.120/adagegui7/ , last accessed on 07 June 2010

[are] many reasons. One is domain knowledge (…) there [are] a lot of things that are not written anywhere. (…) So there is a lot of knowledge that we can only acquire by these frequent interactions. Secondly, SIRCA actually asked us to do other things for them that helped us (…) for example, they had developed a system, they needed documentation and we did documentation for them. By doing the documentation we got to understand the systems better. And thirdly, when we build our services we had to tap into SIRCA's technical infrastructure and that could only be done by being there. So it required frequent interactions for the domain knowledge, for access to the technical infrastructure and also for sort of general, I think for psychological reasons. You know, they see you, they help you better. I think that's a lesson I've learned, yes."[20]

In contrast to the presentation slides that refer to regular, planned monthly meetings, the statement provides evidence that weekly meetings took place to ensure a high level of personal communication. The mostly verbal nature of communication was also confirmed by SIRCA's representative who said:

"The bulk of our communication was in the form of meetings where we would discuss what we would like to support."[21]

On top of this, unscheduled meetings were held whenever there was a need for more information or clarification. From the very beginning, there had been frequent feedback loops between SIRCA and the ADAGE team.[22] This means meetings were not merely part of the project organization or even an annoying weekly obligation. Both sides appreciated the intense exchange and the direct way of communicating. Another aspect of that communication seems striking. Within both parties, there was basically one contact person who was responsible for project-related communication. According to SIRCA's representative, he "primarily (…) dealt with Fethi directly,"[23] and communication was "mainly discussion through me, rather than formal documentation."[24]

Asked if SIRCA provided any kind of documentation like data models, entity relationship models, or information about the data structure, the answer was:

"Largely we gave the group pretty much unfettered access to the data and stood back and let them do the analysis."[25]

This statement confirms the project leader's previous proposition that UNSW researchers wrote some documentation for SIRCA's software system that was unavailable at the beginning of the project. Since no one at SIRCA had been thinking in terms of business processes, two researchers of the UNSW explicitly confirmed that there was no documentation available. Even in the concrete matter of service identification, instead of

[20] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

[21] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

[22] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

[23] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

[24] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

[25] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

formal methods, direct communication was the first choice: “We have had discussions and we agreed upon the service would do a thing.”[26]

Most of the existing literature argues that in order to design services, methodologies and representations of service models, such as unified modeling language (UML) sequence diagrams, are valuable (Glushko, 2008). Contrary to common belief, our case leads to the assumption that these documentation techniques are not always crucial for the success. Documentation is important to provide a common understanding among project members. It might become less significant when – as in the ADAGE case – communication is focused on only one person. Eventually, it might even be more effective to do without documentation in the situation at hand.

Besides other documents, a presentation to SIRCA in September 2009 finally stated that requirements for tools and components were successfully implemented. This evidences that the direct communication actually led to a fulfillment of user requirements.

Observation 7: A technical understanding of services led to very fine-grained services.

The ADAGE project ended up with very fine-grained or so-called elementary services. Starting with SIRCA’s huge data repository, the project team analyzed the given data structures and looked for feasible services from this technical perspective.

“The idea is that you have elementary services and then you can compose these services to make bigger services. Unfortunately, our whole work has been on elementary [services].”[27]

After identifying services this way, the ADAGE team investigated how users could be supported with these elementary services. Depending on the users’ requirements, services can be composed to support (parts of) a business process. Team members agree that the project team approached the identification from a technical point of view[28] and they talk of a layer of elementary services[29], too.

Many authors state that – in contrast to top down modeling – (pure) bottom up techniques typically lead to fine-grained services or functional modules (Erradi et al., 2007; Kohlborn et al., 2009b). Hence, given the technical perspective taken in the ADAGE case, fine-grained services conform to what could have been expected based on existing literature.

Observation 8: A technical understanding of services led to a bottom up direction of analysis.

Since there were no business processes to base a service identification upon, data structures and the existing system served as the starting point. An analysis of these data structures, software, and interfaces is necessarily very technical in nature. The first pro-

[26] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

[27] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

[28] Personal Communication, Interview with the second investigator (UNSW), 04.03.2010

[29] Personal Communication, Interview with the postdoctoral fellow (UNSW), 05.03.2010

ject presentation in February 2007 stated that existing data formats had to be investigated. A later presentation dated 3rd July 2007 showed that project members had analyzed SIRCA's datasets and enabling technologies. The following statement also supports the assumption that identifying services was actually done in a bottom up direction. Business processes that are a precondition for any top down analysis were rather an outcome of composing technical services in a meaningful way.

"We had to build some services and put them together in some kind of business process to show that it is possible to do that kind of things."[30]

An asset analysis – a typical technique in bottom up approaches – was used in ADAGE. Asked for this kind of analysis, one team member answered:

"Yes, we look at what we have. (…) It should be like a tree and basically you need to place it somewhere. If you cannot place it within an existing bunch of services (…) it's something new."[31]

Again, this shows a very technical understanding of services. Even the categories, such as "import services" or "visualization services," to which services belong, were designed in a bottom up fashion as the following statement indicates.

"Every time we have a new case study, we try to re-engineer the categories that we have. (…) When we add a service we need to put it somewhere in a category, in a bunch of services."[32]

A more process-oriented approach would have determined categories from a business process perspective. One example for a fine-grained service is "Import RDTH data" since it merely represents a database query. However, its high reusability (see also Observation 10) shows that bottom up approaches and fine-grained services do not have to be a disadvantage. According to Inaganti and Behara (2007), a technical perspective – focusing on the data level – leads to the application of bottom up techniques. However, to avoid problems stemming from pure bottom up approaches, many authors advocate hybrid approaches that cover top down techniques, such as business process analysis, as well as bottom up methods, such as asset analysis (Klose et al., 2007; Kohlmann and Alt, 2007). Due to the limited complexity of our case, the utilized bottom up approach seems to be sufficient.

Observation 9: The implemented fine-grained services provided flexibility for users and developers at the same time.

On the one hand, the service-oriented architecture implemented in ADAGE provides an enormous flexibility for users because they can simply use the web browser based GUI to compose services according to the analysis at hand.

"The graphical user interface enables you to invoke and compose services in different

[30] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

[31] Personal Communication, Interview with the postdoctoral fellow (UNSW), 05.03.2010

[32] Personal Communication, Interview with the postdoctoral fellow (UNSW), 05.03.2010

ways."[33]

On the other hand, it provides flexibility for developers as well. The architecture can be easily extended and complemented by other services.

"That gives us the flexibility to develop more services and plug them in, or allow others to plug into the environment in the future."[34]

As stated above, ADAGE is also meant to provide a framework for other developers to contribute by creating their own services. For instance, one developer team (such as UNSW's project team) cannot provide a data extraction service for every existing database. Thus, if a user needs a data import from a specific database he can create a service himself and add it to the ADAGE framework. "The idea is to make it open, and basically people can develop services themselves."[35] That way, users of the ADAGE GUI (and the services provided by it) can actively participate in the development and build their own services.

"The flexibility to plug in a service, which could be developed quite quickly to do another step in the analysis process [is] important for various researchers."[36]

Providing a high degree of flexibility for users, developers and those users who become developers themselves by adding further services, the ADAGE case confirms existing literature. Flexibility is frequently said to be one of the most important features of SOAs (Alonso et al., 2003; Papazoglou and Georgakopoulos, 2003). It enables the agility that finally leads to a competitive advantage of organizations (Erl, 2007). According to Levy (2009), many companies plan to use SOA to drive increased flexibility in application development.

Observation 10: Fine-grained services supported reusability.

Reusability of services is a frequently mentioned feature of service-oriented architectures. In essence, the objective is to avoid redundancies of provided software functionality. In ADAGE, an event-based data model for news and financial data was developed (Rabhi et al., 2009a) in October 2007. With this data model, both types of data can be processed at a high level of abstraction reusing the same services.

"Some of the things started as different services, and then you realize there is a common functionality that can be extracted out and offered separately. And this way you get the better SOA."[37]

One conclusion we can draw from the statement above is that the reusability of services was not pursued in a structured way in the ADAGE project. Instead, it was an iterative process. After services had been initially created, someone had to identify common functionalities. This implies that the reusability aspect has to be on the agenda of the

---

[33] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

[34] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

[35] Personal Communication, Interview with the postdoctoral fellow (UNSW), 05.03.2010

[36] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

[37] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

person overlooking an SOA implementation. This rather iterative approach is also confirmed by the following statement:

"Because it always links to what the user wants and, in some cases, if it is not something reusable we make it part of a bigger service. If it is something reusable a lot, then it becomes a service, but that is still linked to what the user wants."[38]

On the customer's side, the reusability aspect was seen from a slightly different perspective. Avoiding redundancies or saving development costs were not SIRCA's primary concern. However, due to the graphical user interface, they had a quite precise idea of what the services were doing. Thus, even on SIRCA's side, the advantages of an SOA in terms of reusability were visible. The fine-grained services provided by the ADAGE GUI guarantee a high reusability and provide for the implementation of user requirements at the same time. According to all interviewees, services in ADAGE are frequently reused. A number of authors have praised the reusability of services as a core aspect of service-oriented architectures (Papazoglou, 2003; Erl, 2005; Josuttis, 2008). Krafzig, Banke and Slama (2004) argue that the use of a service should not be restricted to the project for which it was originally developed. Advantages such as the reduction of maintenance and development costs, reduction of testing effort for new functionalities, and a shorter time to market are often mentioned. Unfortunately, there is only little empirical evidence so far. On the contrary, a case study at Credit Suisse, one of Europe's largest banks, attests only poor reusability of services (Hagen, 2003). Our case study confirms the non-empirically grounded claim of SOA literature that such an architecture indeed supports reusability of services.

Observation 11: Orchestration of services significantly contributed to flexibility.

There are two ways of composing services in order to support business processes. One of them is orchestration, i.e., a central instance (either a user or a workflow system) invokes services to support business processes. The second approach is the so called choreography, in which one service invokes one or several others. According to business rules implemented in these services, the composition of services happens without a central instance controlling it.

"The graphical user interface is allowing ad-hoc processes. So you don't define the process like this, but basically you do it in ad-hoc ways. (…) People don't know exactly what they want to find. So they take data, they apply some tools or services and then, depending on the results, they will decide 'Oh, it looks interesting. Let's try this tool now.'"[39]

In the ADAGE case, the composition of services is supposed to be possible on an ad-hoc basis. Therefore, orchestration is the only feasible option. There is no business logic that could be known and implemented in advance to reflect the necessity or sequence in which services are invoked. Thus, orchestration provides for the demanded flexibility of financial data analysis. The end user represents the central instance that invokes services

[38] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

[39] Personal Communication, Interview with the postdoctoral fellow (UNSW), 05.03.2010

on demand and this provides a high degree of flexibility. This is also evidenced by several presentations, for example, on 21st October, 2008 and 17th February, 2009. In contrast, Janssen and Kuk (2007, 1721) argue that decentralized coordination, i.e., a choreography, is associated with high flexibility because this coordination mechanism can do without a central instance.

Observation 12: Fulfilled user requirements led to user satisfaction.

The incorporation of user requirements in ADAGE was done in a rather ambiguous way. On the one hand, there were significant efforts to acquire domain knowledge from experts in order determine these requirements. This is well documented in the presentations in which, for example, "capturing specialist knowledge" is a frequently revolving theme (beginning in early presentations in February 2007 and still ongoing in September 2008). From the interviews, statements like the following confirm this effort.

"We have the analysis of the needs and the communities of users. What tools do they require?"[40]

On the other hand, the analysis was strongly data-driven, i.e., many services were developed on the basis of existing data structures. Only afterwards the users' requirements were considered.

"So the way we came, we said 'Okay, let's not look at what your user wants, but what your user could want.'"[41]

However, the GUI that is presented to users aims explicitly at usability. Users are enabled to compose services easily according to their needs. Hence, user orientation in ADAGE consists of two aspects that have to be considered separately. Firstly, the identification of services was data-driven rather than user-driven. Only after building services, those were matched to user requirements gained from experts. The importance of acquiring domain knowledge from either customers themselves or employees who know the needs of customers (so called "boundary spanners" (Bowen and Schneider, 1985)) is acknowledged, for example, by Shostack (1987). Secondly, in contrast to the first aspect, the presentation of services to and composition by users is focused on usability and indeed very user-oriented. Several other cases that can be found in literature also focus on usability, which is sometimes used as a benchmark for the success of services (Zimmermann et al., 2004).

Observation 13: Identification and development of services were incremental and iterative processes that supported fulfillment of user requirements.

In the process of identifying services, researchers of the UNSW chose a step-by-step approach to address user requirements with an adequate number and scope of services. First, certain functionalities were implemented as a service. Second, this service was published in the ADAGE GUI. Third, users tested the service and its functionalities and gave feedback. Fourth, this feedback was used to improve existing services or design

---

[40] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

[41] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

new ones. After this, the cycle was started again. This incremental approach is clearly outlined in a presentation dated 10th August, 2009 and supported by the following statement:

"So, the method is basically incremental, that case study after case study you are getting more and more services. And at some stage, having the users contributing as well. The idea is to make it open, and basically people can develop services themselves."[42]

In the near future, users will be able to develop services themselves. The latter can complement existing services after being released by the project team. The iterative character in which services were designed is shown in the following statement.

"We generated the output and then we sent it to Dennis, and Dennis was checking it manually, check if it works fine. (…) And then for weeks and weeks he was sending back defects."[43]

Once existing services were sufficiently improved, new ones were added one after the other in an incremental approach. Thus, starting from a small basis with only few services, the offer of services was broadened step by step.

Looking at methods for service identification in literature, the sequence of activities to identify services constitutes an important element of such methods. Frequently, a linear sequence of activities is proposed, leaving only little room for iterative steps or feedback loops (Börner and Goeken, 2009). The observations in the ADAGE case contradict these approaches. The method that emerged in the course of the project (emergent method – see Observation 4) was feedback-driven and based on "builds" in terms of increments. The feedback-driven fashion also entailed a high degree of user participation. These elements (feedback-driven processes, builds/increments, user participation, and iterations) are typical core elements of iterative and incremental development approaches (Larman and Basili, 2003; Gilb and Finzi, 1988). Based on a field study, Haines and Rothenberger (2010) argue that "building the service infrastructure requires a rigorous – and perhaps more waterfall-style – approach focusing on long-term planning to achieve well-designed and reusable services. On the other hand, composite application development may in many circumstances benefit from using a more agile, yet rigorous, software development approach." (p. 137)

The realization of an iterative and incremental development approach – even if it was emergent – allowed for flexibility in reacting to user requirements as well as for the integration of new ideas and unforeseen demands concerning functionality. We regard this as an important factor to achieve the user satisfaction we encountered in the case study.

Observation 14: The flexibility provided by services led to user satisfaction.

Generally, flexibility is one of the most commonly cited properties of service-oriented architectures. Oftentimes, it is associated with the flexible composition of services by a

---

[42] Personal Communication, Interview with the postdoctoral fellow (UNSW), 05.03.2010

[43] Personal Communication, Interview with the postdoctoral fellow (UNSW), 05.03.2010

company in order to support business processes that need to be adapted at an ever increasing speed. Flexibility for users can be considered just as important.

"The flexibility to plug in a service, which could be developed quite quickly to do another step in the analysis process would be important."[44]

The following statement shows that this flexibility leads to increased customer satisfaction.

"So [users] like that flexibility that the system offers in composing services in different ways."[45]

By satisfying customers, the flexibility of the ADAGE services contributes to the overall success of ADAGE. Again, in many presentations throughout the whole project, flexibility through end-user composition and customization are revolving topics on a number of slides. Related literature acknowledges that flexibility represents a major value potential of SOAs (for an overview see (Becker et al., 2009)).

Observation 15: User satisfaction was a success measure.

Satisfied customers were clearly a goal of SIRCA and thus of the ADAGE project. Although customer satisfaction was not explicitly defined as a success measure, we consider it to be elementary for the success of ADAGE.

"Yes, they see the service at the graphical user interface level, so they like the graphical user interface level. (…) So, they like that flexibility that the system offers in composing services in different ways. And this is why they sort of continue supporting us."[46]

If customers "like" the GUI, they are satisfied with it, and the continuing support from SIRCA's side clearly indicates the project's success. In many presentations (e.g., one dated 24th September, 2008), the objective *Implement the requirements* is outlined. This implies that reaching this goal was considered necessary for users' satisfaction and, hence, a successful SOA implementation. This aspect strongly supports Observation 15.

According to the DeLone and McLean IS success model (DeLone and McLean, 2003), "customer satisfaction" is a result of system quality, information quality, and service quality alike. It is thus one important success measure to evaluate the success of information systems. Hence, our observations from the ADAGE case conform to DeLone and McLean's view.

Observation 16: Reusability was a success measure.

The objective *Reusing existing components* was part of the overall strategy, as can be gathered, for example, from a presentation dated 21st October, 2008 . It reoccurs in many presentations throughout the whole project. Whereas success was seen from a business perspective so far, this is a rather technical aspect that can be used to evaluate success. From this point of view, an SOA can be seen as a success if it provides for re-

---

[44] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

[45] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

[46] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

usability and thus reduces redundancies. A significant amount of money can be saved by reducing costs for development and maintenance of functionalities which are represented by services (Bieberstein et al., 2005; Legner and Heutschi, 2007). Hence, not only literature considers reusability as a success measure but also the ADAGE project team. Furthermore, SIRCA sees the benefit of reusable services as well. Although services are used for only one business process at the moment, SIRCA can at least imagine deploying the former in other processes, as the following statement shows.

“Could we puzzle them in different ways to do different things? - Absolutely!”[47]

Table 2 gives an overview of our observations in the first column. In the following section, we will use these observations to derive hypotheses. Abstraction mechanisms are utilized for a generalization that transforms observed concepts to abstract categories. This will result in a model of hypotheses and their relationships (Figure 6).

4.4 Generalization: Shaping Hypotheses

The generalizations to derive hypotheses are based upon the observations described above. According to Walsham (1995), generalization aims at giving explanations of phenomena “which may be valuable in the future in other organizations and contexts” (p. 79). He outlines four types of generalizations: the development of concepts, the generation of theory, the drawing of specific implications, and the contribution of rich insight. Our main goal is to generalize our observations and, in doing so, create a set of hypotheses.

As mentioned earlier, we repeatedly apply principles and techniques of interpretative case study research as well as abstraction mechanisms of conceptual modeling in order to guide this activity. While the principles are well known in case study research, to the best of our knowledge, the latter have not been used in the realm of qualitative research. We believe that these mechanisms provide a stable set of well described (and in some way formalized) ways to abstract from instances (resp. parts) to types (resp. sets). This will promote transparency and traceability of the generalization procedure in this activity.

In the following, we will briefly introduce two prominent abstraction mechanisms which are later applied to several observations (Goeken, 2006; Olivé, 2007; Mattos, 1989):

- Classification relates instances with a type, or to be more precise, it consists of determining the types which an object is an instance of (Olivé, 2007) (pp. 12 and 383). Instances have common properties and are assembled into a new entity for which uniform conditions hold.
- Aggregation (Composition) defines part-whole structures of instances.

The concepts we identified in our observations can be interpreted as instances, and thus we can use abstraction mechanisms to generalize from these observations. The resulting

[47] Personal Communication, Interview with the industry coordinator (SIRCA), 09.03.2010

types or sets form the components of our hypotheses, and we have labeled them categories. Hence, by applying abstraction mechanisms, we create abstract categories (e.g., we can classify the concept “fine-grained services” as an instance of the category “granularity”). Of course, the categories induced will have more concepts than those which we have observed in our case study. For instance, the category “granularity” also has the instance “coarse-grained services.” Using these alternative instances of categories, we go beyond the details observed and further generalize the findings.

In applying these abstraction mechanisms, we clearly move beyond the observations. We acknowledge that we leave the firm empirical foundation when deducing types and adding instances not observed in the case study. However, by making the generalization as transparent as possible, we argue that the generation of reasonably grounded hypotheses can be justified. Furthermore, while formulating our hypotheses, we bear in mind the criteria of good hypotheses like falsifiability, consistency, testability, or disconfirmability. (Lee, 1989)

In the following, we present our hypotheses, support them and describe how the generalization and the merging of the observations was performed, applying the aforementioned abstraction mechanisms. The mapping of our observations to the hypotheses is illustrated in Table 2. As in activity four, related literature is given and compared to our findings.

Observation	**Hypothesis**	**Related Literature**	**Compared to literature**
1. The generous funding significantly influenced the way of service identification.	1. Context factors significantly influence the way of SOA development and service identification.	Becker et al. (2007), Sedera (2008), Faraj and Sproull (2000), Börner (2010), Anderson et al. (2005), Bucher et al. (2007)	+
2. SIRCA's small company size significantly influenced the way of service identification.			+
3. People's skills significantly influenced the way of service identification.			+
4. The implemented SOA satisfied users, although no formal method for the identification of services was used.	2. Not using a formal method for service identification does not necessarily lead to a failure of SOA projects.	Vessey and Glass (1998), Balzert (2001), Sommerville (2001), Kohlmann and Alt (2007), Klose, et al. (2007), Winkler (2007), Arsanjani et al. (2008), Kohlborn et al. (2009a)	-
5. Little SOA experience led to a technical understanding of services.	3. SOA project experience leads to a different understanding of services and thus affects success measures.	Vergidis et al. (2008), Börner et al. (2009), Anderson et al. (2005), Baskerville et al. (2005), Becker et al. (2009)	+
6. Personal communication was vital to meet user requirements.	4. Personal communication can substitute utilization of formal methods.	Glushko (2008), Faraj and Sproull (2000), Ancona and Caldwell (1992)	-
7. A technical understanding of services led to very fine-grained services.	5. The understanding of services affects both the granularity of services and the direction of the service identification approach.	Erradi et al. (2007), Kohlborn et al. (2009b), Inaganti and Behara (2007), Klose et al. (2007), Kohlmann and Alt (2007), Kohlborn et al. (2009a)	+
8. A technical understanding of services led to a bottom up direction of analysis.			+
9. The implemented fine-grained services provided flexibility for users and developers at the same time.	6. The right granularity of services can affect multiple success measures and depends strongly on the project at hand.	Alonso et al. (2003), Papazoglou and Georgakopoulos (2003), Erl (2007), Levy (2009), Rabhi et al. (2009a), Papazoglou (2003), Erl (2005), Josuttis (2008), Krafzig et al. (2004), Hagen (2003), Elfatatry (2007)	+
10. Fine-grained services supported reusability.			o
11. Orchestration of services significantly contributed to flexibility.		Janssen and Kuk (2007)	-
12. Fulfilled user requirements led to user satisfaction.		Bowen and Schneider (1985), Shostack (1987), Zimmermann et al. (2004)	+
13. Identification and development of services were incremental and iterative processes that supported fulfillment of user requirements.		Börner and Goeken (2009a), Larman and Basili (2003), Gilb and Finzi (1988), Haines and Rothenberger (2010)	o
14. The flexibility provided by services led to user satisfaction.		Becker et al. (2009)	+
15. User satisfaction was a success measure.	7. The success of SOA projects is expressed through technical and business-oriented success measures.	DeLone and McLean (2003), Bieberstein et al. (2005), Legner and Heutschi (2007), Luthria et al. (2007), Papazoglou (2003)	+
16. Reusability was a success measure.			+
Legend: Our findings "+" largely confirm literature, "o" confirm some but refute other literature or "-" refute literature, respectively.			

Table 2: Observations, Hypotheses and Enfolding Literature

Figure 6 shows how the relation between concrete concepts (observations) is mapped to an abstract view that generalizes from the ADAGE case to other settings of SOA implementations. The upper part shows types and respective hypotheses.

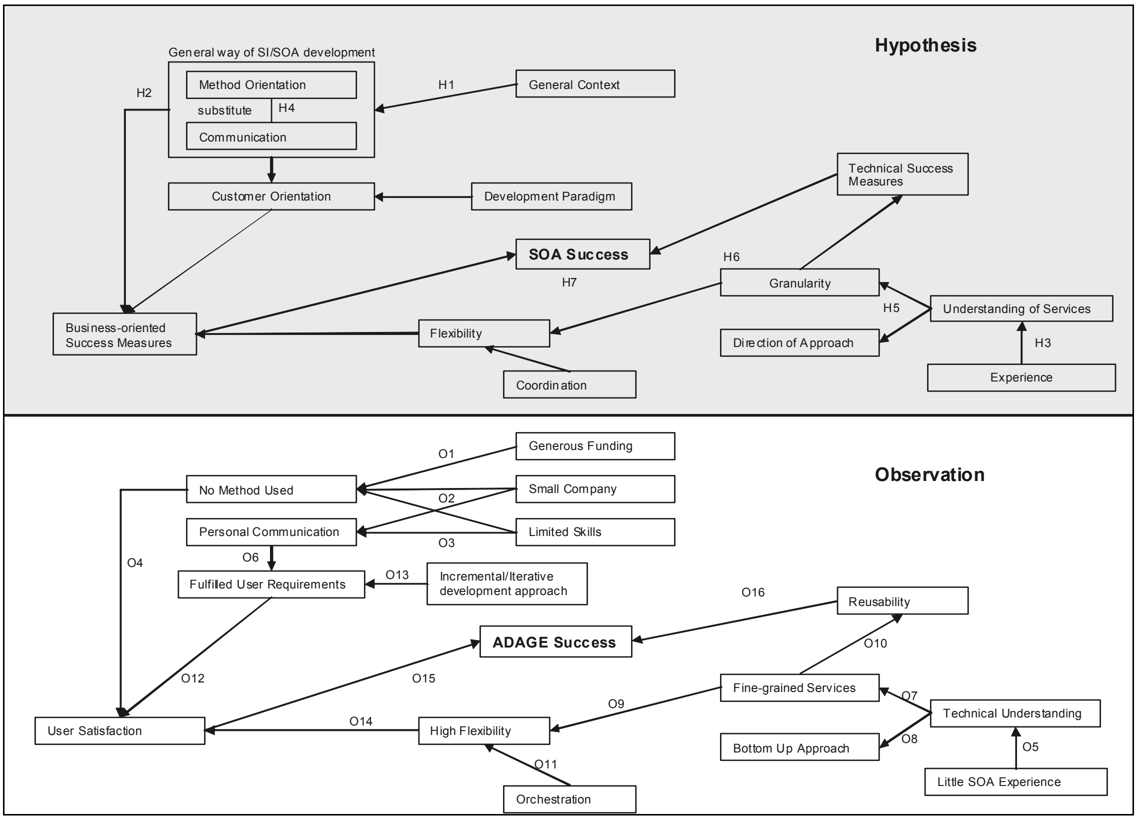

Figure 6: Interrelationships of Observations and Hypotheses

Hypothesis 1: The general context significantly influences the method of SOA development and service identification.

Certainly, there are many factors that affect the approaches to service identification in general and the choice of certain methods in particular. Some of them cannot be influenced by the project team and have to be taken for granted (Bucher et al., 2007). In the following, such conditions and circumstances shall be referred to as *context factors*.

We observed generous funding, small company size, and limited skills in ADAGE as conditions and circumstances in this respect (observations 1 to 3). Applying the abstraction mechanism classification, we can generalize these instances by determining their types and, furthermore, define part-whole structures of instances and types (aggregation): Figure 7 exemplarily shows the generalization from observations 1, 2, and 3 to Hypothesis 1. For instance, the category "company size" is the result of classification from the observed concept "small company." Thus, there is a has-instance-(instance-of-) relationship between these two. Furthermore, an aggregation of both concepts and categories leads to "ADAGE context" and "general context," respectively. They are connected to the original concepts and categories through is-part-of-relationships. Budget, company size, and people's skills are the resulting categories that are related to the concepts described in observations 1 to 3. These categories are generic "context factors."

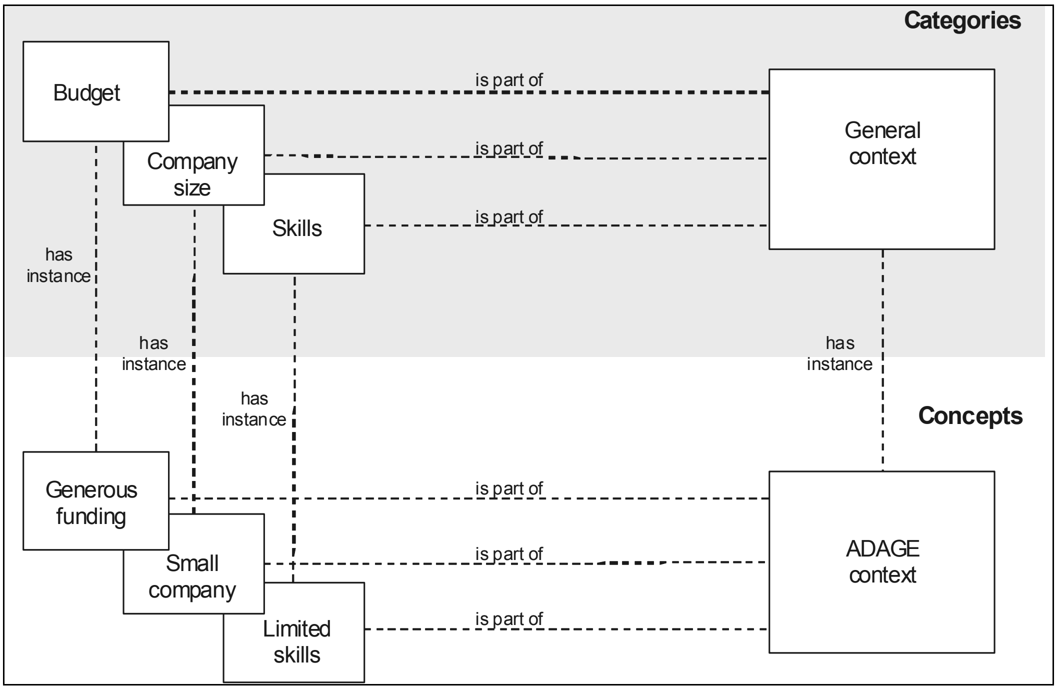

Figure 7: Generalization and Aggregation at the Example of Hypothesis 1

As shown in the observations, the combination of a small company and employees with limited skills (trumping the opportunities of a generous budget and an academic environment) led to an absence of formal methods in service identification. This is certainly an extreme case. However, context factors significantly influence the way of service identification. Thus, we hypothesize that large companies with highly skilled employees will use different methods for SOA development and the identification of services.

Hypothesis 2: Not using a formal method for service identification does not necessarily lead to a failure of SOA projects.

A method orientation, i.e., using or not using one or more methods, depends on context factors. Certain situations (as combinations of context factors) can make a method obsolete. The right choice of methods (or the decision to use no method at all) influences the success of software projects in general and SOA projects in particular (Balzert, 2001; Sommerville, 2001). This influence is expressed indirectly through customer satisfaction, which is one possible success measure for SOA projects.

However, the following statement gives reason to assume that – for simple settings like in the SIRCA case – a method is not absolutely necessary.

“It was very simple. You use a method when it’s very complex. (…) In business processes something like that would make sense.”[48]

Although no particular method was used, the ADAGE project can be considered successful after all, as Observation 4 showed in detail. Thus, Hypothesis 2 is supported by the evidence given in the interviews.

[48] Personal Communication, Interview with the second investigator (UNSW), 04.03.2010

Most literature on service identification argues that an application of methods is essential (Kohlmann and Alt, 2007; Klose et al., 2007; Winkler, 2007; Kohlborn et al., 2009a; Arsanjani et al., 2008). However, none of the authors discusses if the adoption of a method depends on the complexity of a project or its context factors, respectively.

At least in one case, the absence of a method did not lead to a failure of the project. In other situations, however, there might be a need for methods to support service identification. We assume that certain situations, i.e., the combination of context factors, make the application of methods either obsolete or necessary. This respective situation further influences the concrete configuration of a method that should be used in the situation at hand.

Hypothesis 3: SOA project experience leads to a different understanding of services and thus affects success measures.

In ADAGE, we observed that little SOA experience accompanied with the dominance of IT specialists resulted in a technical understanding of services (Observation 5). Classifying the concepts observed, we assume that the level of experience influences the understanding of services. We therefore hypothesize that companies with a track record of service implementation tend to involve business departments early in the process of service identification. Therefore, the understanding of services is much more process-oriented. Through multiple cause-and-effect relationships shown in Figure 6, success measures such as reusability are affected.

The task of planning a service-oriented architecture should be tackled by both business departments and an organization’s IT division (Börner et al., 2009). Implementing an SOA is primarily a technical challenge. We assume that – like in the ADAGE case – many SOA projects underestimate the importance of properly planning an implementation. Hence, consideration of business processes is often poor, and employees of the IT division take the lead in these projects. Especially in companies that have little or no experience with the nature of services, the dominance of IT specialists might lead to a technical understanding of services. More experience in this field might lead to different outcomes. A project team might decide to use a process-oriented approach, a user-oriented approach, or even choose a technical approach if suitable. The most significant difference is that an experienced project team with a broader understanding of SOA is free to choose an adequate approach from a range of options. Unfortunately, this makes a prediction almost impossible. The only cause-and-effect-relationship that can be established here is that more experience allows for different opportunities whereas little experience most certainly leads to a technical understanding.

Anderson et al. (2005) argue that the “extent to which the enterprise architect, service developer, and operations project staff in the internal IT department of the firm are skilled” (p.68), i.e., their experience and subsequently their understanding of services, is a critical factor for web service implementation. Looking at our observations 5, 7, and 10, we can further hypothesize that SOA experience indirectly influences reusability of services. This is confirmed by Baskerville, Cavallari, Hjort-Madsen, Pries-Heje, Sorrentino, and Virili (2005), who show that “very few web services could be reused exactly as originally implemented” (p. 7) and thus, experience is essential for SOA implementations. Becker et al. (2009) state that "especially in the early phases of SOA maturity [i.e., with little experience] there is no experience about [what] a reusable service would look like" (pp. 7-8). This is another hint that SOA project experience tends to change the understanding of services and enhances reusability.

Hypothesis 4: Personal communication can substitute utilization of formal methods.

A combination of the small company size and the lack of skilled staff made direct communication crucial for the success of ADAGE. Thus, this direct communication was vital for the service identification as Observation 6 highlights. It finally delivered services that cover user requirements and satisfy customers. Figure 8 illustrates that different forms of communication and method orientation influence concrete concepts of how services are identified and SOAs are developed. Generally, we hypothesize that in certain situations communication can substitute the utilization of methods.

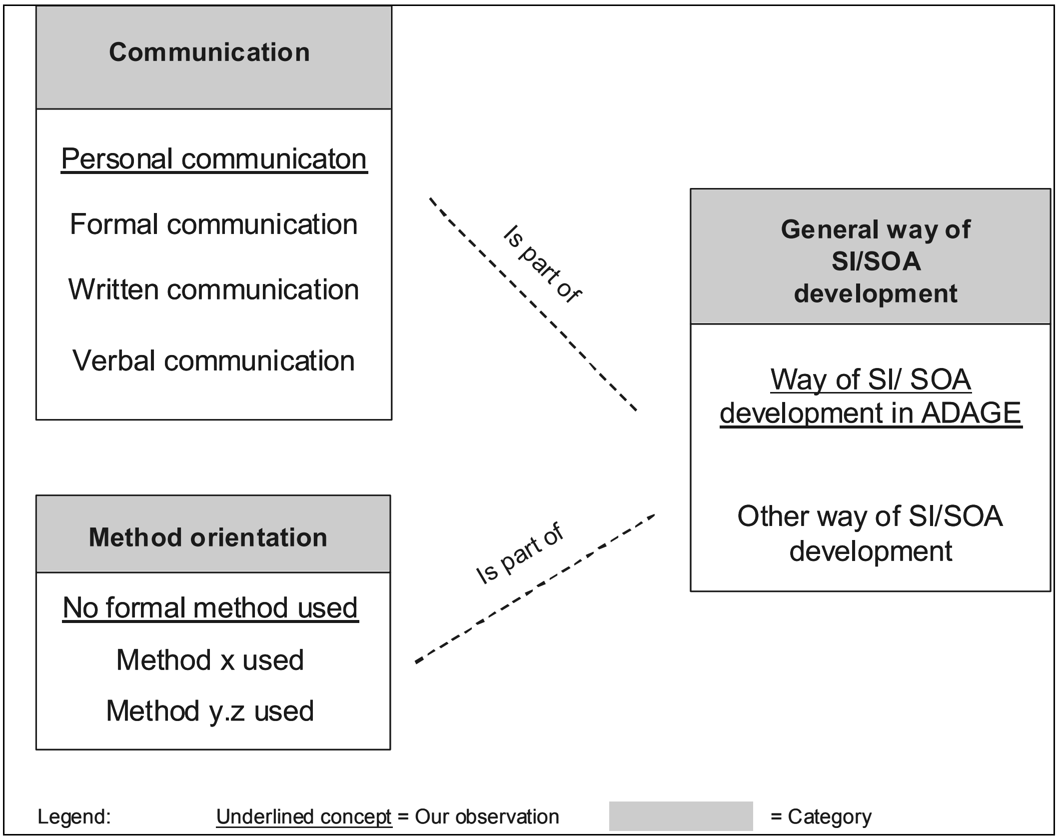

Figure 8: Communication and Method Orientation as Part of a Way of SI/SOA Development

These situations are characterized by the combination of context factors. In a case in which people's skills are very limited, the company size is rather small, and the budget is tight, communication is likely to be intense and to focus on one person with respective knowledge. The same can be true even if the budget (and consequently the time frame) is more generous, as in the ADAGE case (see observations 1 to 3). Certainly, in larger projects with a substantial number of staff involved on both sides, a structured approach using appropriate methods for identifying services and implementing an SOA can hardly be replaced by communication alone. Administrative measures for the coordination of development teams are discussed, for example, in Faraj and Sproull (2000) and Ancona and Caldwell (1992). This is evidence that larger teams (in larger projects) necessitate a higher degree of formal organization, and personal communication is unlikely to meet demands.

Hypothesis 5: The understanding of services affects both the granularity of services and the direction of the service identification approach.

In the ADAGE project, a technical understanding of services led to very fine-grained services and a bottom up approach stressing technical aspects of the underlying (data) infrastructure (see observations 7 and 8). Firstly, we can hypothesize that a different understanding of services led to different granularities. Moving on a scale from technical to business-oriented understanding, the identified services might become coarser-grained along this line. From a business perspective, services should support processes or at least sub processes. Hence, they encompass more functionalities than an elementary service that, for example, reads an address from a database. The latter will more likely be the outcome if service identification is conducted from a technical point of view.

Secondly, the direction of the approach is influenced by this understanding. Again, moving along the above scale will result in the application of bottom up approaches on one side and top down approaches on the other side. It is unlikely to end up with a pure instance of any of the approaches. Even in the ADAGE case, a very technical understanding did not result in a pure bottom up approach. However, the share of techniques that are typically assigned to top down approaches (such as strategic analyses) will increase with a business-oriented understanding of services. Accordingly, in their comparison of service analysis approaches, Kohlborn et al. (2009a) differentiate two types of SOA concepts, i.e., understandings of SOA. One of them is rather technical and, thus, delivers so-called software services while the other (business-oriented) one results in business services.

Hypothesis 6: The right granularity of services affects multiple success measures and depends strongly on the project at hand.

In service-oriented architectures, granularity of services is a widely discussed issue among researchers and practitioners alike. Our observations 9 and 10 indicate that fine-grained services positively contributed to both reusability and (indirectly) customer satisfaction in the ADAGE case. Thus, we hypothesize that granularity indeed plays a major role for the success of SOAs in general since it potentially affects more than one success measure. Granularity, flexibility, and success measures are therefore categories which constitute Hypothesis 6.

Many authors have argued that – for a successful SOA implementation – a business point of view is indispensable. Accordingly, only coarse-grained services that support a business functionality or a business process provide a value for an organization and make an SOA a success. From a technical point of view, reusability, for example, suffers from too coarse-grained services. The finer-grained a service is the higher its reusability.

Whereas the ADAGE case showed clear advantages of fine-grained services, coarse-grained services might be the better choice in other settings. Some customers might not be interested in the flexible composition of services every time they use them. If they simply want to outsource part of a business process, they might prefer a coarse-grained service that encompasses all necessary functionalities and delivers a comprehensive result. Accordingly, Elfatatry (2007) argues that "the appropriate level of granularity for a service and its methods is relatively coarse. A service generally supports a single distinct business concept or process." (p. 38) In the ADGAE case, fine-grained services enhanced both reusability and user satisfaction, i.e., technical and business-oriented success measures, respectively. In other cases, the granularity of services might be a trade off because effects on these two kinds of measures could be converse.

We argue that, due to different preconditions, the right granularity of a service has to be elaborated depending on the situation at hand. There is no silver bullet for right-sizing a service without considering the context of service implementation. Thus, a situation-specific approach to the choice of methods is important to provide for an adequate granularity.

Hypothesis 7: The success of SOA projects is expressed through technical and business-oriented success measures.

In our observations 15 and 16, we found user satisfaction and reusability to be success measures for the ADAGE project. Both contribute to the overall success of the project but are quite different in nature. Reusability is a rather technical aspect, whereas user satisfaction is a clearly business-oriented measure. Thus, we assume that the overall success of an SOA consists of both business-oriented and technical success measures.

From the business side, customer satisfaction (as the category corresponding to user satisfaction with ADAGE services) is a crucial issue to generate new business and maximize turnover and profit. According to DeLone and McLean (2003), the "use" of a system and the "intention to use" are just as important since they indicate user acceptance. Whereas customer satisfaction was observed to be a success measure in our case study, the other two factors were not observable. Since DeLone and McLean argue that these three factors are closely interrelated, we aggregate them under business-oriented success measures.

Due to the immature state of the service implementation, SIRCA's customers had not been able to use services directly.

"So at the moment if somebody needs something, I use the service to do what they want, simply because they are not mature, but part of the follow-up project is to release the toolset to the right public."[49]

Nonetheless, the project team did care about the use of services and implicitly considered this use to be a success measure. In this context, a researcher of the UNSW men-

[49] Personal Communication, Interview with the principal investigator (UNSW), 04.03.2010

tioned that in the business intelligence use case of ADAGE “the system has not been used by anybody.”[50]

IT departments tend to use technical measures for success. Observation 16 identified reusability as being such a success measure. IT departments pursue the objective to cut costs and improve the overall reliability and manageability of the IT infrastructure, for example, by avoiding redundancies. Typically, standardization is another frequently mentioned technical success measure (Luthria et al., 2007; Papazoglou, 2003).

Factors contributing to these success measures usually stem from different perspectives regarding an SOA. However, there are concepts that might influence both types of measures. Observations 9 and 10 show that granularity (indirectly) affects both business-oriented and technical measures. A reduced time-to-market (or development time) is another example for a concept having an impact on both kinds of measures but could not be observed in our case.

4.5 Integrating Hypotheses

Through generalization of our observations, we created hypotheses as illustrated in Figure 6. Due to the fact that the concepts observed as well as the categories in the different hypotheses overlap, we can combine the hypotheses in a model or in a small set of model fragments. Our observations and hypotheses are closely linked to the primary data we retrieved from documentation and interviews. In the following, we will describe phenomena that move beyond this data. By performing another generalization and looking at causal links between concepts, we arrive at middle-range-theories that were not necessarily obvious beforehand.

The resulting models (model fragments) can best be interpreted as a “middle range theory” because they are sufficiently abstract to be applied to different contexts but do not offer a set of general laws. Middle range theories allow contextual explanations and “lie between the minor but necessary working hypotheses that evolve in abundance during day-to-day research and the all-inclusive systematic efforts to develop a unified theory that will explain all the observed uniformities of social behaviour, social organization and social change.“ (Merton, 1967) (p. 39) The model fragments we present in this chapter have limited scope and are applicable to limited conceptual ranges as they only grasp phenomena from the systems development / SOA domain. Furthermore they refer to selected aspects in the realm of SOA development because we focused our analysis on the questions of investigation presented in 4.1.

Partitioning our model, we can identify chains of interconnected hypotheses (sections 4.5.1 to 4.5.3). These “partial models” allow for further detailing and variations and, hence, the formulation and discussion of alternative theories. Even though this discussion moves beyond our case data, it is still linked to the hypotheses and indirectly grounded in them. Since these theories are based on chains of several hypotheses, they

[50] Personal Communication, Interview with the second investigator (UNSW), 04.03.2010

provide new insights. In the following subsections, three models which present an external, internal, and a success perspective are elaborated.

Furthermore, by analyzing the interplay of different observations and hypotheses, we identify a phenomenon worth discussing in more detail. We develop a construct called “emergent method” by applying a different “type of generalization” suggested by Walsham (1995).

4.5.1 Contingency Model (CM)

Context factors such as budget, company size, and people’s skills are consolidated in Hypothesis 1. All of them (“general context”) influence the application of methods, communication, and the general approach to implementing SOA projects. We assume that alternative instances of the mentioned categories will result in different ways of SI/SOA development, as depicted in Figure 9. Furthermore, the list of categories constituting our contingency model is limited to those we were able to observe in our case study. For instance, in a more business-driven SOA implementation project, the choice of a BPM software tool could influence the way of SI/SOA development. Since there was no BPM tool used in our case, we were not able to observe such an impact. Accordingly, the way of SI/SOA implementation will most likely include more concepts than method orientation and communication which we were not able to observe.

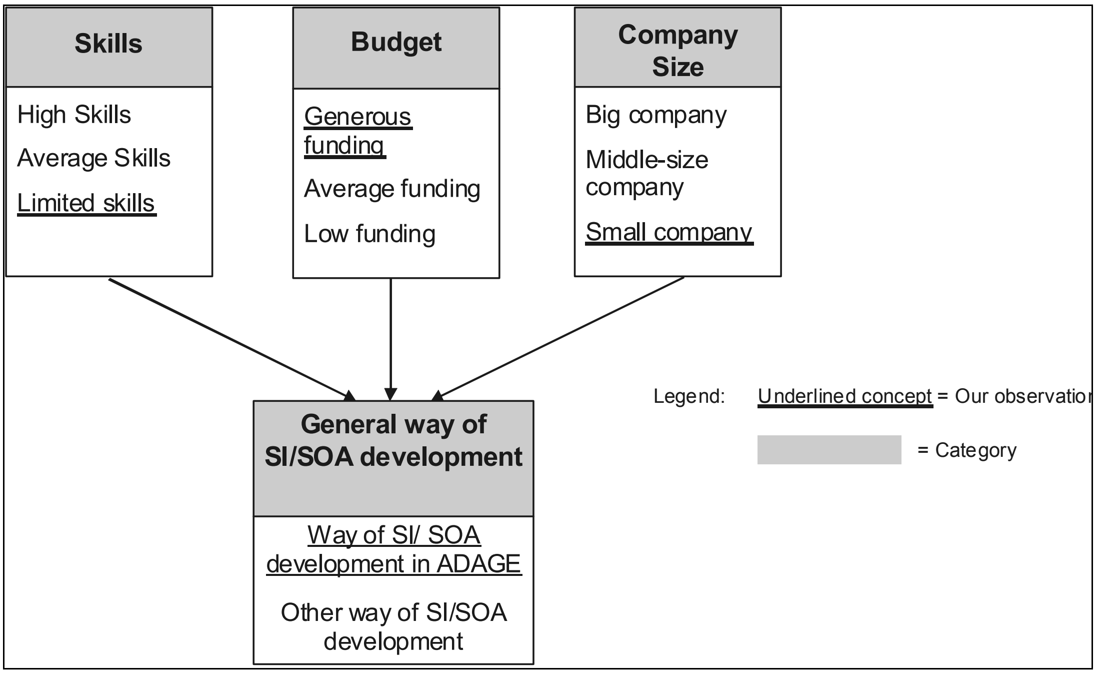

Figure 9: Influence of the General Context on the Way of SI/SOA Implementation (Contingency Model – CM)

To the best of our knowledge, there are neither frameworks/models nor empirical studies in academic literature which analyze situational factors in the realm of SOA. A first step towards a comprehensive list using more case studies has been made by Börner, Goeken, Kohlborn and Korthaus (2011).

4.5.2 Model of Soft Factor Transition (MSFT)

The model fragment which ties together hypotheses 3, 5, and 6 is depicted in Figure 10. Our observations (underlined concepts) and the resulting cause-and-effect chain show that a team with only little SOA experience produced a highly flexible service-oriented architecture and highly reusable services. In our hypotheses, experience leads to a distinct understanding of services (H3). This results in different granularities and directions of approaches (H5) and subsequently in a certain degree of flexibility and reusability (H6). Hence, the experience of project teams influences flexibility and reusability as well as the direction of the identification approach through a chain of cause-and-effect relationships as depicted in Figure 10. Teams with little SOA experience and limited knowledge regarding business functions and processes are forced to rely on the experience they have, which is basically technical. We assume that developing an SOA based on the knowledge stemming from experience made with traditional, more technical driven software engineering paradigms, for example, object-oriented programming or modules, results in fine-grained services. The understanding of services and granularity are the generative mechanisms in this model fragment.

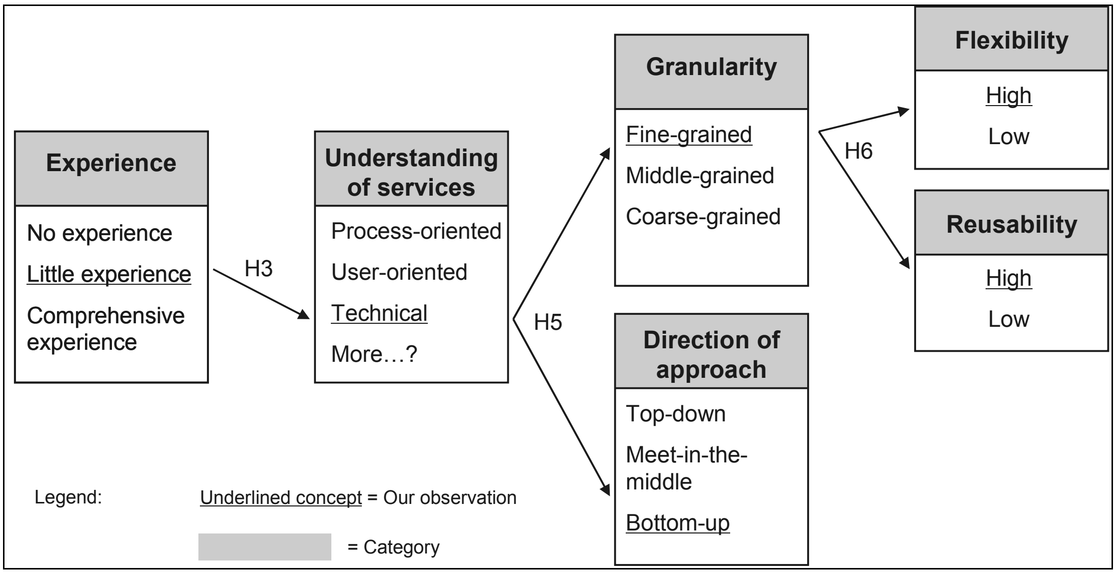

Figure 10: Causal Link of Experience and Flexibility/Reusability (Model of Soft Factor Transition – MSFT)

We could assume that a very experienced team might have a process-oriented understanding which leads to rather coarse-grained services that provide less flexibility. Since flexibility is regarded to be a big advantage of SOAs, this causal link is counterintuitive. We would expect an experienced team to be able to reap the benefits of SOAs, including an enhanced flexibility. Figure 11 illustrates our observed pattern (dotted line) and the assumed possible pattern described above (dashed line).

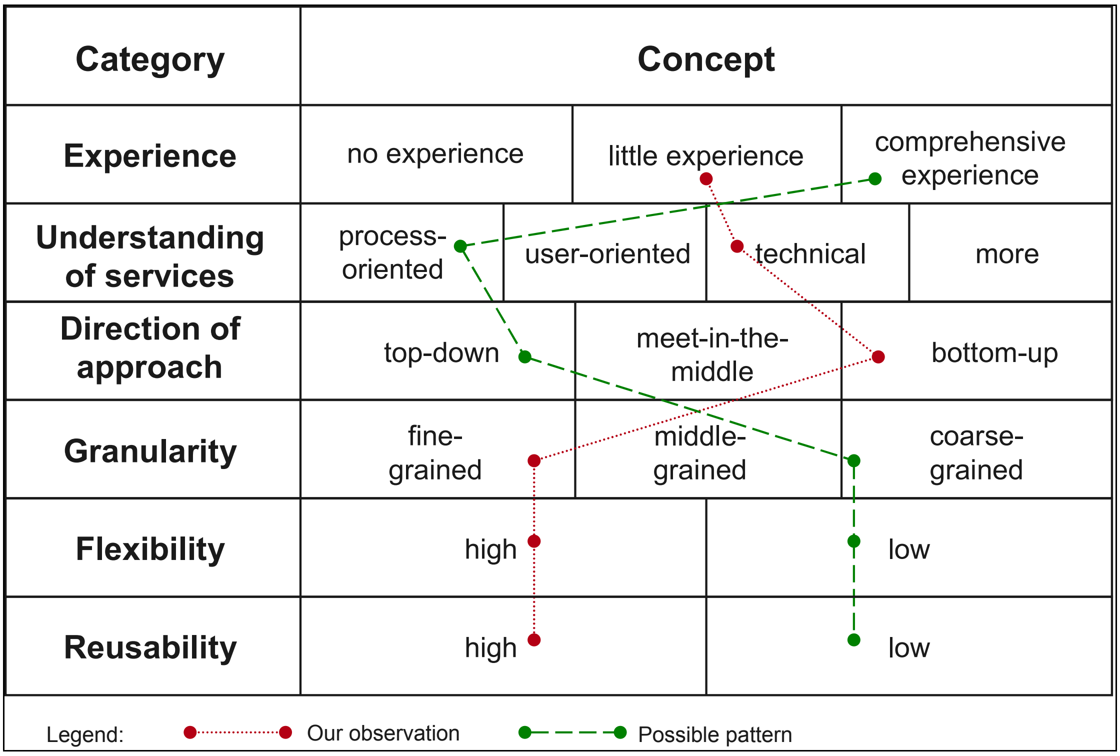

Figure 11: Project Patterns

However, we have to keep in mind that more experience does not necessarily lead to one specific understanding of services but enables an adequate choice of approach and understanding, as argued in Hypothesis 3. A comprehensive experience, for instance, is not closely knit to a process-oriented understanding and could as well produce middle-grained services. A large number of patterns is thus conceivable. We argue that – depending on multiple context factors in projects – the resulting patterns represent certain types of projects. An identification of such project types could lead to a targeted application of methods supporting SOA development and the identification of services. Project types can define situations in the sense of situational method engineering and would subsequently contribute to a more sophisticated service identification through the use of situational methods (Bucher et al., 2007).

Observations 7 and 8 showed that the technical understanding of services in the ADAGE project led to fine-grained services and an almost pure bottom up approach for identifying services. In this regard, the lack of experience once again led to a rather object-oriented thinking which reveals another generative mechanism at work in our case study. In Hypothesis 5, these observations are generalized. Undoubtedly, there is an effect of this understanding on granularity and the direction of service identification approaches. Actually, the question of granularity continues to be an unresolved issue among scientists and practitioners alike and constitutes an active research area (e.g., Artus (2006)). The right size of a service is always a trade-off between its potential for reuse and better management of quality issues.

4.5.3 Success Model (SM)

Many observations as well as hypotheses are bound to the notion of *success*. Naturally, the primary objective of every project is to be completed successfully. This is also true for SOA implementation projects. But how can you evaluate the success of such a project? Our observations and hypotheses utilized several success measures that are influenced by a number of different concepts in our model illustrated in Figure 6. In our observations, we found success factors such as *reusability* and *user satisfaction*. These success factors reflect the success of an SOA. However, they take different view points and usually represent interests of different stakeholders. On the one hand, reusability is a rather technical aspect. Subsequently, in most cases, it is pursued by the IT department that implements a service-oriented architecture. The number of business processes that invoke a service and the frequency of invocation can serve as a unit of measurement. If the IT department can prove that a service is reused frequently in many processes, it is easy to argue that maintenance and development of this single service is much cheaper than having the same functionality provided by multiple applications scattered across the organization. On the other hand, user satisfaction is clearly a business goal. Users have little interest in cost savings or technical demands on the company's side. Instead, they appreciate flexibility and a timely satisfaction of their demands in a good quality. Hence, business processes must be flexible and agile to result in a certain degree of user satisfaction. As argued before, a successful SOA can cater to these demands.

A number of success measures can be employed (see, e.g., the framework by Börner and Goeken (2009)). Business departments can thus consider flexibility and time-to-market as important factors. From the technical side, autonomous services or a high degree of standardization might be important. In both cases, the measurement of success factors is far from trivial, and appropriate measures will have to be defined to operationalize the measurement of SOA success. To which extent technical and business-oriented success factors are aligned or not is an interesting question that is left to further research. Based on a literature review and expert interviews, Lee, Shim, and Kim (2010) identify 20 critical success factors in SOA implementation. However, their interrelationships and contribution to success are not clear.

4.5.4 Emergent Method as a New Construct

Observation 4 shows clearly that no formal method was applied in the ADAGE case. User requirements were analyzed, as can be seen in Observation 11. According to Observation 13, an iterative and incremental approach was established during the course of the project.

Hence, our observation that no method was used, which was originally based on interview statements (see Hypothesis 2), may be questioned with good reason. Of course, no explicit method was used. However, certain rules developed during the course of the project and these were applied repeatedly, as Observation 13 evidences. Following Mintzberg's ideas of "emergent strategies" (Mintzberg, 1994), we argue that within the ADAGE project a method emerged. We consider a method as an "emergent method" if a pattern develops in the absence of intentions or despite them (Mintzberg, 1987). In ADAGE, the pattern emerged in the absence of a pre-defined and formal method.

Most likely, this observation can be made in other IT development projects as well. Under which circumstances emergent methods appear is another intriguing research question. An analysis of their stability and adaptability is of interest as well. Maybe those methods have a higher degree of acceptance among team members and are more reliable due to their emergence within the context of the project.

5 Discussion and Further Research

In the previsous section we developed three model fragments and created one new concept called emergent method. The model fragments cannot be treated as being independent. On the one hand, there are evident relationships that can be traced in Figure 6. Obviously, the notion of success which is operationalized by certain measures integrates the contingency model (CM) and the model of soft factor transition (MSFT). On the other hand, there are most likely more links that have not been observed in our case. As discussed earlier, the concept of skills in the CM is most likely related to what is called experience in the MSFT. Moreover, the way of SI/SOA development (CM) is probably influenced by the direction of the identification approach, which is currently not part of the contingency model but of the MSFT. Hence, we assume that there are more causal links on several levels of our proposed model, so that the latter will have to be adjusted after more elaborate work on this topic is performed.

5.1 Discussion of Findings and Further Research

In our contingency model, we explained the influence of a number of context factors on the way of SI/SOA development. An extension of the list of factors, which form part of the general context, will be left to future research and can be achieved for instance through further case studies or expert interviews. Moreover, based on only one case study, it is impossible to determine the extent to which a single factor contributes to certain outcomes. After conducting more case studies, a factor analysis could help to understand more reliably how the combination of identified factors affects the use of methods. Since there is already a large body of literature that provides a comprehensive list of context factors for domains different from SOA, future research should concentrate on identifying those that have a significant impact on SOA implementation projects and, in doing so, understanding the generative mechanisms more deeply.

The model of soft factor transition links experience with SOA design and implementation aspects (flexibility and reusability) and provides a description of a generative mechanism which is likely to be at work. One question for future research that arises from this model is if a technical (resp. business-oriented) understanding of services necessarily leads to fine-grained (resp. coarse-grained) services and a bottom up (resp. top down) approach. Most likely, more experience would enable the project team to flexibly adapt granularity, create services on different levels of granularity, or apply a top down (resp. a hybrid) approach. Further case studies would help to shed light on these cause-and-effect relationships.

Furthermore, there might be other context factors that have strong effects on granularity and the direction of the approach. For instance, a high degree of top management commitment might support top down approaches because strategic aspects on an organizational (or even inter-organizational) level have to be incorporated. These aspects might be neglected if only middle management or the IT department were involved in the project.

Another intriguing question is whether one of these mindsets yields more success, and if this is the case, by which measure? According to most literature, business orientation is crucial for successful SOA implementations because only this focus ensures that services can support business processes. However, the ADAGE case showed that an almost pure bottom up approach and very fine-grained services stemming from a technical understanding still can result in a successful SOA implementation. Of course, the ADAGE architecture is far from being business process-oriented. Nonetheless, it can be considered a success. Since case studies can be used for generating hypotheses (like in this paper) and testing hypotheses (De Vries, 2005), a multiple case study could be useful to investigate correlations of service understanding and success in SOA projects.

In general, concepts and categories in our model fragments are designed based on the evidence retrieved from the ADAGE project. However, relationships and interdependencies which we were not able to observe may exist. For instance, we identified skills of employees to be an important context factor influencing service identification methods. Furthermore, SOA experience had an impact on concepts such as understanding of services and, subsequently, flexibility and reusability. Most likely, concepts such as skills and experience are somehow related, even though there is no explicit hint to be found within the collected data. Literature argues that experience is one possible source to improve skills (see, e.g., Adelson and Soloway (1985), Dokko, Wilk and Rothbard (2009), Guile (2002)). Thus, further empirical evidence might support interdependencies found in literature and make an adjustment of our model necessary.

Our success model shows that multiple concepts more or less directly influence success of an SOA. This success can be viewed from different perspectives, i.e., technical or business-oriented. A model illustrating how these success factors contribute to the overall success of an SOA implementation could be a result of further empirical research. Using multiple case studies is one possibility to find supporting or contradicting evidence for the generated hypotheses. This would enable a cross-case pattern search and underpin the validity of the hypotheses (Eisenhardt, 1989). Further interviews, surveys, and document examination could be a basis for qualitative or (in large numbers) even quantitative cross-sectional analyses that are common and wide-spread methodologies in ISR (Wilde and Hess, 2007).

Currently, the success measures in our model are not well operationalized. Future research should improve the conceptualization of success measures as well as their measurement. It is necessary to describe how to measure these factors and which units to use. Finally, an evaluation of the importance of a single concept (e.g., the direction of approach) and to what degree it influences the success of an SOA would be desirable. Such a quantification could be achieved by a factor analysis of context factors.

5.2 Methodological Considerations – Discussion of the Research Process

Although single case studies are an appropriate and accepted method for our purpose (see section 3), there are limitations to this approach. According to Darke et al. (1998), "where explanatory research is undertaken, a single case may provide the basis for developing explanations of why a phenomenon occurs, and these may then be further investigated by applying them to additional cases in other settings." (p. 281) Moreover, a search for cross-case patterns using divergent techniques as proposed by Eisenhardt (1989) is only possible with multiple case studies. Although statistical, sampling-based generalizability cannot be provided by single case studies, Lee & Baskerville (2003) provide a framework that distinguishes four types of generalizability and allows for the use of single case studies. However, more cases would be desirable to support the hypotheses generated in this paper on the basis of a single case study.

Since our case study is explorative in nature, it is well suited to show starting points for further research in this field. Our observations are grounded in facts retrieved from various sources, such as documentation and interviews. The hypotheses, however, have been generated in an interpretative manner and, thus, demand additional empirical evidence and testing. It has to be noted that in a single case study there is a certain probability that important concepts and categories were missed. Therefore, further research should not only focus on testing the hypotheses provided but should maintain openness to new phenomena arising.

Regarding the research process, we modified the established framework of Eisenhardt (1989) and extended methodological work in the realm of qualitative research (Klein and Myers, 1999; Myers, 2009) by applying abstraction mechanisms of conceptual modeling as guidelines for generalization. The intention was to enhance transparency and traceability of the generalization and to better justify the derivation of hypotheses. In this respect, it could be argued that these mechanisms might reduce creativity and limit abduction in the interpretative analysis. On the other hand, there are more abstraction mechanisms (e.g., aggregation, contextualization, kind-of …) which could be applied in a similar way. A more detailed elaboration, for example, the development of an appropriate framework for the use of abstraction mechanisms in qualitative research, requires further intensive methodological work.

6 Conclusion

Based on a case study of an SOA implementation project in an Australian company, we have made 16 observations and subsequently developed seven hypotheses, three model fragments, and one additional concept. In our research process, which is based on Eisenhardt (1989) and adapted as explained in section 3.2, we used techniques such as abstraction and generalization to interpret first-order data retrieved in the case study. We tried to provide evidence for the generation of hypotheses and the exploration of areas where existing knowledge is limited.

All our findings help to answer our initial research question because each unveils facets of how SOA development and service identification work in real-life projects. Moreover, our questions of research (section 5.1) have been answered satisfyingly:

- Whereas literature recommends the *use of methods*, there was no method used in our case. This must not necessarily lead to a failure of a project. Under certain circumstances, i.e., in some contexts, the absence of formal methods can be compensated for by intense communication.
- Several observations, hypotheses, and model fragments deal extensively with the *context factors* that influence the proceedings of the service identification process. We were able to observe some of these factors and to investigate their influence on SOA implementation projects.
- The *understanding of services* and related skills significantly influence the way of SOA development. Several hypotheses and model fragments show that concepts such as granularity, the direction of the identification approach, and finally the success of an SOA implementation are more or less directly influenced by the understanding of services (which is usually subject to change with increasing experience).
- Due to the absence of a method for service identification, we were not able to observe a stringent application of *software development techniques*. Due to the limited scope of the project, software development was mostly intuitive and bound to few developers.
- Our findings identified user satisfaction and reusability as *success measures*. More generally speaking, they suggest that it might be meaningful to differentiate between business-oriented and technical success measures, an issue which should be studied in future research.

The ambiguous results concerning the effects of method use, service understanding, granularity, etc. show that a method for service identification has to be tailored to a project at hand. The general context – including numerous context factors – has to be considered in order to appropriately support service identification. Especially the interplay of these factors has to be elaborated in order to develop situation-specific methods for service identification.

References

Adelson, B. and Soloway, E. (1985) The Role of Domain Experience in Software Design, *IEEE Transactions on Software Engineering*, 11, 11, 1351-1360.

Alonso, G., Casati, F., Kuno, H. and Machiraju, V. (2003) Web Services: Concepts, Architectures and Applications, Springer, Berlin, Heidelberg.

Ancona, D. G. and Caldwell, D. F. (1992) Demography and Design: Predictors of New Product Team Performance, Organization Science, 3, 3, 321-341.

Anderson, D., Howell-Barber, H., Hill, J., Javed, N., Lawler, J. and Li, Z. (2005) A Study of Web Services Projects in the Financial Services Industry, Information Systems Management, 22, 1, 66-76.

Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S. and Holley, K. (2008) SOMA: A method for developing service-oriented solutions, IBM Systems Journal, 47, 3, 377-396.

Artus, D. J. N. (2006) SOA Realization: service design principles, IBM, [http://www.ibm.com/developerworks/webservices/library/ws-soa-design/], Accessed: 25-08-2010.

Atkins, C. and Sampson, J. (2002) Critical Appraisal Guidelines for SingleCase Study Research, European Conference on Information Systems, Gdansk, Poland, 100-109.

Avison, D. E. and Fitzgerald, G. (2008) Information Systems Development: Methodologies, Techniques and Tools, McGraw-Hill, New York, NY.

Balzert, H. (2001) Lehrbuch der Software-Technik: Software-Entwicklung, Spektrum Akademischer Verlag, Heidelberg.

Baskerville, R., Cavallari, M., Hjort-Madsen, K., Pries-Heje, J., Sorrentino, M. and Virili, F. (2005) Extensible Architectures: The Strategic Value of Service-oriented Architecture in Banking, European Conference on Information Systems, Regensburg, Germany.

Becker, A., Buxmann, P. and Widjaja, T. (2009) Value Potential and Challenges of Service-Oriented Architectures - A User and Vendor Perspective, 17th European Conference on Information Systems (ECIS), Verona, Italy.

Becker, J., Knackstedt, R., Pfeiffer, D. and Janiesch, C. (2007) Configurative Method Engineering, 13th Americas Conference on Information Systems (AMCIS), Keystone, Colorado, Paper 56.

Benbasat, I., Goldstein, D. K. and Mead, M. (1987) The case research strategy in studies of information systems, MIS Quarterly, 11, 269-386.

Bieberstein, N., Bose, S., Walker, L. and Lynch, A. (2005) Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals, IBM Systems Journal, 44, 4, 691-708.

Birkmeier, D., Glöckner, S. and Overhage, S. (2009) A Survey of Service Identification Approaches, Enterprise Modelling and Information Systems Architectures, 4, 2, 20-36.

Boland, R. J., Newman, M. and Pentland, B. T. (2010) Hermeneutical exegesis in information systems design and use, Information & Organization, 20, 1-20.

Börner, R. (2010) Applying Situational Method Engineering to the Development of Service Identification Methods, 16th Americas Conference on Information Systems, Lima, Peru, Paper 18.

Börner, R. and Goeken, M. (2009) Identification of Business Services - Literature Review and Lessons Learned, 15th Americas Conference on Information Systems (AMCIS), San Francisco, California, Paper 162.

Börner, R., Goeken, M., Kohlborn, T. and Korthaus, A. (2011) Context Factors for Situational Service Identification Methods, 6th International Conference on Internet and Web Applications and Services, Xpert Publishing Services, St. Maarten, The Netherlands Antilles, 35-42.

Börner, R., Looso, S. and Goeken, M. (2009) Towards an Operationalisation of Governance and Strategy for Service Identification and Design, 13th IEEE International EDOC Conference, Auckland, New Zealand, 180-188.

Bowen, D. E. and Schneider, B. (1985), Boundary-Spanning-Role Employees and the Service Encounter, in John A. Czepiel, Michael R. Solomon and Carol F. Surprenant (Eds.) The Service Encounter: Managing Employee/Customer Interaction in Service Businesses. Lexington, Lexington, MA, 124-147.

Bucher, T., Klesse, M., Kurpjuweit, S. and Winter, R. (2007), Situational Method Engineering - On the Differentiation of "Context" and "Project Type", in Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.) Situational Method Engineering: Fundamentals and Experiences. Springer, Boston, 33-48.

Charters, S. M., Budgen, D., Turner, M., Kitchenham, B., Brereton, P. and Linkman, S. G. (2009) Objectivity in Research: Challenges from the Evidence-Based Paradigm, Australian Software Engineering Conference, Gold Coast, Australia, 73-80.

Darke, P., Shanks, G. and Broadbent, M. (1998) Successfully completing case study research: combining rigour, relevance and pragmatism, Information Systems Journal, 8, 4, 273-289.

De Vries, E. J. (2005) Epistemology and Methodology in Case Research: A Comparison between European and American IS Journals, 13th European Conference on Information Systems, Regensburg, Germany.

DeLone, W. H. and McLean, E. R. (2003) The DeLone and McLean model of information systems success: a ten-year update, Journal of Management Information Systems, 19, 4, 9-30.

Department of Innovation, I., Science and Research (2010) International Science Linkages - Selection Outcomes, Commonwealth of Australia.

Dokko, G., Wilk, S. L. and Rothbard, N. P. (2009) Unpacking Prior Experience: How Career History Affects Job Performance, Organization Science, 20, 1, 51-68.

Eisenhardt, K. M. (1989) Building Theory from Case Study Research, The Academy of Management Review, 14, 4, 532-550.

Eisenhardt, K. M. and Graebner, M. E. (2007) Theory building from cases Opportunities and challenges, Academy of Management Journal, 50, 1, 25-32.

Elfatatry, A. (2007) Dealing with Change: Components versus Services, Communications of the ACM, 50, 8, 35-39.

Erl, T. (2005) Service-Oriented Architecture - Concepts, Technology, and Design, Prentice Hall, Upper Saddle River, NJ.

Erl, T. (2007) SOA - Principles of Service Design, Prentice Hall, Upper Saddle River, NJ.

Erradi, A., Kulkarni, N. and Maheshwari, P. (2007) Service Design Process for Reusable Services, Lecture Notes In Computer Science, 4749, 606-617

Faraj, S. and Sproull, L. (2000) Coordinating Expertise in Software Development Teams, Management Science, 46, 12, 1554-1568.

Fischer, G., Nakakoji, K. and Ye, Y. (2009) Metadesign: Guidelines for Supporting Domain Experts in Software Development, IEEE Software, 26, 5, 37-44.

Gadamer, H.-G. (1976), The Historicity of Understanding, in P. Connerton (Eds) Critical Sociology, Selected Readings. Penguin Books, Harmondsworth, UK, 117-133.

Gilb, T. and Finzi, S. (1988) Principles of Software Engineering Management Addison Wesley, Reading, MA.

Glushko, R. J. (2008) Designing a service science discipline, IBM Systems Journal, 47, 1, 15-27.

Goeken, M. (2006) Entwicklung von Data-Warehouse-Systemen. Anforderungsmanagement, Modellierung, Implementierung, Deutscher Universitäts-Verlag, Wiesbaden.

Gregor, S. (2006) The nature of theory in information systems, MIS Quarterly, 3, 30, 611-642.

Guabtni, A., Kundisch, D. and Rabhi, F. A. (2010) A user-driven SOA for Financial Market Data Analysis, Enterprise Modelling and Information Systems Architectures, forthcoming.

Guile, D. (2002) Skill and Work Experience in the European Knowledge Economy, Journal of Education & Work, 15, 3, 251-276.

Hagen, C. (2003), Integrationsarchitektur der Credit Suisse, in Stephan Aier and Marten Schoenherr (Eds.) Enterprise Application Integration - Flexibilisierung komplexer Unternehmensarchitekturen, 1. edn. Gito, Berlin, 61-83.

Haines, M. N. and Rothenberger, M. A. (2010) How a Service-Oriented Architecture May Change the Software Development Process, Communications of the ACM, 53, 8, 135-140.

Inaganti, S. and Behara, G. K. (2007) Service Identification - BPM and SOA Handshake, BPTrends, 3, 1-12.

Janssen, M. and Kuk, G. (2007) Comparing Coordination Arrangements Enabled by Web Services and Web Service Orchestration Technology, The 15th European Conference on Information Systems, 1718-1728.

Jarke, M. (2009) Perspectives in the Interplay Between Business and Information Systems Engineering and Computer Science, Business & Information Systems Engineering, 1, 1, 70-74.

Josuttis, N. (2008) SOA in der Praxis - System-Design für verteilte Geschäftsprozesse, dpunkt.verlag, Heidelberg.

Klein, H. H. and Myers, M. D. (1999) A set of principles for conducting and evaluating interpretive field studies in information systems, MIS Quarterly, 23, 1, 67-94.

Klose, K., Knackstedt, R. and Beverungen, D. (2007) Identification of Services - A Stakeholder-Based Approach to SOA Development and Its Application in the Area of Production Planning, 15th European Conference on Information Systems (ECIS), St. Gallen, Switzerland, 1802-1814.

Kohlborn, T., Korthaus, A., Chan, T. and Rosemann, M. (2009a) Identification and Analysis of Business and Software Services - A Consolidated Approach, IEEE Transactions on Services Computing, 2, 1, 50-64.

Kohlborn, T., Korthaus, A., Chan, T. and Rosemann, M. (2009b) Service Analysis - A Critical Assessment of the State of the Art, 17th European Conference on Information Systems, Verona, Italy.

Kohlmann, F. and Alt, R. (2007) Business-Driven Service Modeling - A Methodological Approach from the Finance Industry, in Witold Abramowicz and Leszek Maciaszek (Eds.) Business Process & Service Computing: First International Working Conference on Business Process and Services Computing, Gesellschaft für Informatik e.V., Leipzig, Germany, 180-193.

Krafzig, D., Banke, K. and Slama, D. (2004) Enterprise SOA: Service Oriented Architecture Best Practices, Prentice Hall International, Upper Saddle River, NJ.

Larman, C. and Basili, V. R. (2003) Iterative and Incremental Development: A Brief History, Computer, 36, 6, 47-56.

Lee, A. S. (1989) A scientific methodology for MIS case studies, MIS Quarterly, 13, 1, 33-50.

Lee, A. S. and Baskerville, R. L. (2003) Generalizing Generalizability in Information Systems Research, Information Systems Research, 14, 3, 221-243.

Lee, J. H., Shim, H.-J. and Kim, K. K. (2010) Critical Success Factors in SOA Implementation: An Exploratory Study, Information Systems Management, 27, 2, 123-145.

Legner, C. and Heutschi, R. (2007) SOA Adoption in Practice - Findings From Early SOA Implementations, in Hubert Österle, Joachim Schelp and Robert Winter (Eds.) 15th European Conference on Information Systems (ECIS), St. Gallen, Schweiz, 1643-1654.

Lehman, M. M. and Ramil, J. F. (2002) Software Evolution and Software Evolution Processes, Annals of Software Engineering, 14, 1-4, 275-309.

Levy, E. (2009) Data as a service emerges, Information Management, 21-23.

Luthria, H. and Rabhi, F. A. (2009), Building the Business Case for SOA: A Study of the Business Drivers for Technology Infrastructure Supporting Financial Service Institutions, in Dennis Kundisch, Daniel J. Veit, Tim Weitzel and Christoph Weinhardt (Eds.) Enterprise Applications and Services in the Finance Industry. Springer, 94-107.

Luthria, H., Rabhi, F. A. and Briers, M. (2007) Investigating the Potential of Service Oriented Architectures to Realize Dynamic Capabilities, 2nd IEEE Asia-Pacific Conference on Services Computing, Tsukuba, Japan, 390-397.

Mattos, N. M. (1989) Abstraction Concepts: The Basis for Data Knowledge Modeling, in Carlo Batini (Eds) 7th International Conference on Entity-Relationship Approach: A Bridge to the User., North Holland, 473-492.

Merton, R. (1967), On sociological theories of the middle-range, in R. Merton (Eds) On Theoretical Sociology: Five essays old and new. Free Press, New York,

Mintzberg, H. (1987) The Strategy Concept I: Five Ps For Strategy, California Management Review, 30, 1, 11-24.

Mintzberg, H. (1994) Rethinking Strategic Planning Part I: Pitfalls and Fallacies, Long Range Planning, 27, 3, 12-21.

Myers, M. D. (2009) Qualitative Research in Business & Management, Sage Publications, London.

Olivé, A. (2007) Conceptual Modeling of Information Systems, Springer, Berlin Heidelberg.

Orlikowski, W. J. (1993) CASE tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development, MIS Quarterly, 17, 3, 309-340.

Papazoglou, M. P. (2003) Service-Oriented Computing: Concepts, Characteristics and Directions, 4th International Conference on Web Information Systems Engineering, IEEE Computer Society, Rome, Italy, 3-12.

Papazoglou, M. P. and Georgakopoulos, D. (2003) Service-Oriented Computing, Communications of the ACM, 46, 10, 25-28.

Pham, Q.-K., Benatallah, B., Saint-Paul, R., Mouaddib, N. and Raschia, G. (2009) Time-Aware Content Summarization, 15th ACM Conference on Knowledge Discovery and Data Mining SIGKDD'2009, Paris, France.

Rabhi, F., Baradarannia, M. and Yao, L. (2010) A Case study using the Ad-hoc DAta Grid Environment (ADAGE) for Financial Time Series Building, FinanceCom 2010, Frankfurt/Main, Germany.

Rabhi, F., Guabtni, A. and Yao, L. (2009a) A data model for processing financial market and news data, International Journal of Electronic Finance, 3, 4, 387-403.

Rabhi, F., Rana, O. F., Guabtni, A. and Benatallah, B. (2009b), A User-Driven Environment for Financial Market Data Analysis, in Dennis Kundisch, Daniel J. Veit, Tim Weitzel and Christoph Weinhardt (Eds.) Enterprise Applications and Services in the Finance Industry. Springer, Berlin Heidelberg, 64-77.

Sedera, D. (2008) Does Size Matter? Enterprise System Performance in Small, Medium and Large Organizations, 2nd Workshop on 3rd Generation Enterprise Resource Planning Systems, Copenhagen, Denmark.

Shostack, G. L. (1987) Service Positioning through Structural Change, Journal of Marketing, 51, January 1987, 34-43.

Sommerville, I. (2001) Software Engineering, Pearson Studium, München.

Sommerville, I. (2006) Software Engineering 8, Addison Wesley, Reading, MA.

Standish Group International (1994) The CHAOS Report, [http://www.projectsmart.co.uk/docs/chaos-report.pdf], Accessed: 08-09-2010.

Stebbins, R. A. (2001) Exploratory Research in the Social Sciences, Sage Publications, Thousand Oaks.

Strauss, A. and Corbin, J. M. (1990) Basics of qualitative research: Grounded theory procedures and techniques, Sage Publications, Thousand Oaks, CA.

Thomas, O., Leyking, K. and Scheid, M. (2010) Serviceorientierte Vorgehensmodelle: Überblick, Klassifikation und Vergleich, Informatik-Spektrum, 33, forthcoming.

van Maanen, J. (1979) The Fact of Fiction in Organizational Ethnography, Administrative Science Quarterly, 24, 4, 539-550.

Vergidis, K., Turner, C. J. and Tiwari, A. (2008) Business process perspectives. Theoretical developments vs. real-world practice, International Journal of Production Economics, 114, 1, 91-104.

Vessey, I. and Glass, R. (1998) Strong vs. weak approaches to systems development, Communications of the ACM, 41, 4, 99-102.

Walsham, G. (1995) Interpretive case studies in IS research. Nature and Method, European Journal of Information Systems, 4, 74-81.

Walsham, G. (2006) Doing interpretive research, European Journal of Information Systems, 15, 320-330.

Wilde, T. and Hess, T. (2007) Forschungsmethoden der Wirtschaftsinformatik - Eine empirische Untersuchung, Wirtschaftsinformatik, 49, 4, 280-287.

Winkler, V. (2007) Identifikation und Gestaltung von Services. Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich, Wirtschaftsinformatik, 49, 4, 257-266.

Yin, R. K. (2003) Case Study Research - Design and Methods, SAGE Publications, Thousand Oaks London New Delhi.

Zimmermann, O., Milinski, S., Craes, M. and Oellermann, F. (2004) Second Generation Web Services-Oriented Architecture in Production in the Finance Industry, 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2004), ACM, Vancouver, Canada, 283-289.

Appendix

Appendix A: Interview Schedule

Date	Time	Room	Interviewee Role
04.03.2010	14.15-14.45	CSE Building, UNSW, Sydney	Second investigator
04.03.2010	15.15-15.45	CSE Building, UNSW, Sydney	Principal investigator
05.03.2010	11.00-12.00	CSE Building, UNSW, Sydney	Postdoctoral fellow
09.03.2010	16.00-16.30	80 Clarence St., 9th floor, Sydney	Industry coordinator

Appendix B: List of Websites

Website	URL
ADAGE GUI	http://129.94.172.120/adagegui7/
ADAGE Project Website	http://cgi.cse.unsw.edu.au/~soc/adage/index.php?n=Main.HomePage
Competitive Grants Round 10 Outcomes	https://grants.innovation.gov.au/ISL/Pages/Doc.aspx?name=CGR10.htm
SIRCA Homepage	http://www.sirca.org.au/
LibreSource – ADAGE project	http://soc-server.cse.unsw.edu.au/projects/adage

Appendix C: List of Services

Categories	Services	Description
Import	Reuters News Import Service	Each service imports news as event sets from a particular source. RNIS imports from the Reuters archive.
	RDTH Import Service (Market data from file)	Imports events from the Reuters Datascope Tick History (RDTH) system1 into the shared event repository. These events are mostly trades and quotes from every financial exchange worldwide. In addition, RDTH gives prices for every asset identified by the Reuters Identification Code (RIC), such as currency exchange rates, indices, and interest rates. This service allows events to be selected according to a wide range of criteria, such as period, RICs, exchange etc. This was a test service to avoid connecting to external systems
	RDTH Direct Import Service (Market data from external systems)	Each service imports market data as event sets from a particular source. RDTH Import Service imports from the Reuters Data Tick History (RDTH) service
Processing	Time series building	Processes trades and quotes to produce trade price time-series sampled at equal time intervals. This service is highly customizable as it allows the sampling period to be modified and a number of measures (such as the return, spread, and vwap) to be included in the time series. In addition, events from different event sources can be combined within a single time series. For example, the service can determine prices of multiple stocks and the index in which they belong. Another example is to combine a stock, its futures, and the interest rates in one time series. Each time series value is modeled as a Measure in our data model. Builds a time series of user-defined financial measures from market data event sets
	Merge	Used to merge different time series, for example, equity with index data
	Tagging	Used to tag events with “appropriate tags.” For example, news can be tagged with keywords inside the news story. Must be done prior to summarization.
	Price Jump	Used to detect abnormal returns by comparing two price time series with each other
	News Summarization	Used to group together news stories with the same tags.
Export	Download	Exports a time series from the shared event repository into a text format (e.g., CSV) suitable for processing using a statistical package like SPSS or Matlab. Used to download event sets on the local computer
	Visualization	Used to visualize event sets. Different graphs are shown depending on the type of events
	Preview	Gives some information on an event set

Appendix D: Interview Guidelines for UNSW Researchers

1) Can you tell me about your participation in the ADAGE project? (tasks, time frame, direct communication partners at SIRCA)
2) What kind of documentation on business processes did SIRCA provide?
3) Which other documents were provided?
4) How did SIRCA communicate user requirements?

5) Which primary business processes did you identify from the information you got from SIRCA?
6) How did you get from SIRCA's processes and requirements to services? What was the starting point? When was the identification completed?
7) How did you decide on cuts in services? Where does one service start/end?
8) Do you think some of your services have different granularities?
9) Are there different service layers?
10) Did you think about visualization techniques (like service landscapes)?
11) How did you model relations between services and inputs/outputs?
12) Did your services give the support that SIRCA expected? How do you know?
13) Was the service identification straight forward or more of an iterative approach?
14) Did you frequently use feedback loops to match your services with SIRCA's requirements?
15) Did you have to redesign or replace services?
16) How did you make sure that the implemented services met SIRCA's needs?
17) Were there any other stakeholders apart from SIRCA?
18) Do you think the SOA/services you implemented are a success? By which measure would you judge this?
19) Were there any requirements you were not able to implement as a service? Why?
20) What were the main difficulties in identifying services?
21) Looking at SOA principles, would you say your services are
    a. reusable
    b. loosely coupled
    c. autonomous
    d. interface-oriented
    e. flexibly composable
22) Did you use any method or guidelines to pursue the service identification? If not, would you feel more confident about the quality of implemented services if you had used a method (guidelines) when identifying them?
23) Is there room for an improvement in identifying and designing services so that user needs are better met?

5.3 Artikel 8: Generalization in Qualitative IS Research - Approaches and Their Application to a Case Study on SOA Development

Veröffentlicht: Goeken, M., Börner, R. (2012): Generalization in Qualitative IS Research - Approaches and Their Application to a Case Study on SOA Development, in: Australasian Journal of Information Systems, 17. Jg., Nr. 2, S. 79-108.

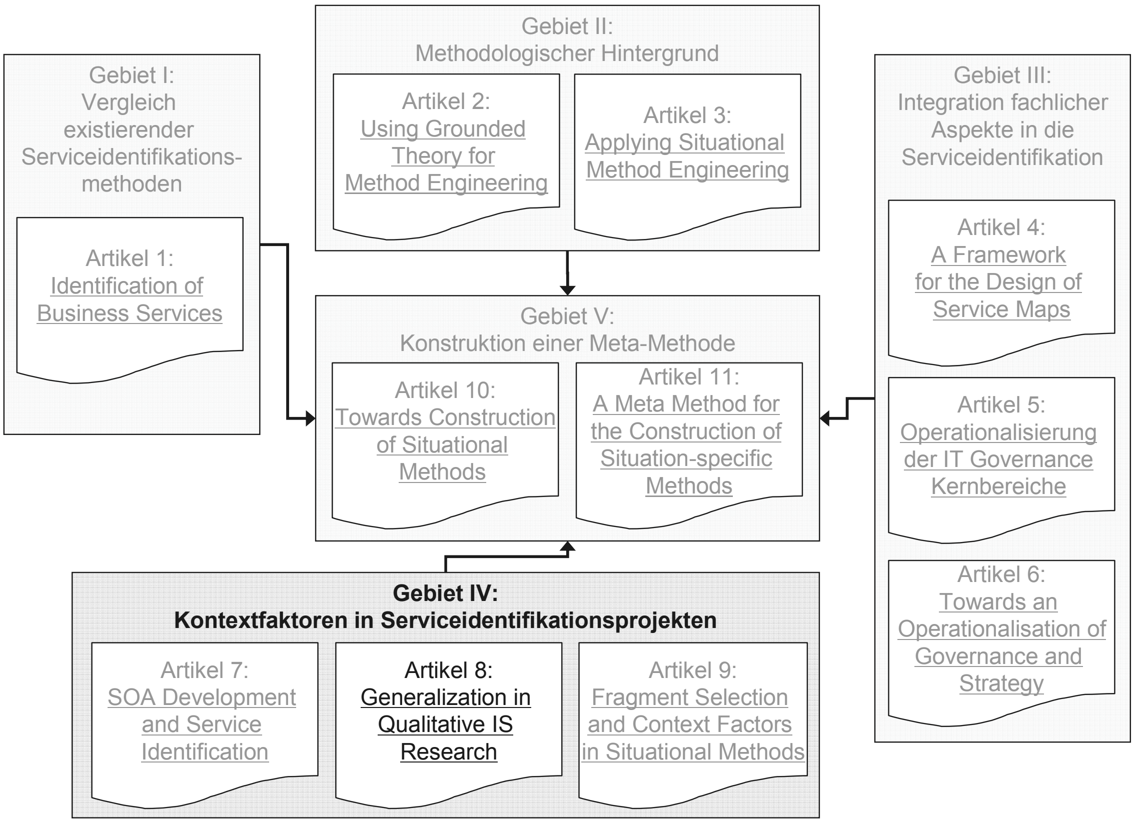

Abbildung 13: Artikel 8 im Kontext der kumulativen Dissertation

Generalization in Qualitative IS Research Approaches and Their Application to a Case Study on SOA Development

Matthias Goeken
Frankfurt School of Finance & Management
m.goeken@fs.de

René Börner
Frankfurt School of Finance & Management
r.boerner@fs.de

Abstract:

In qualitative information systems research, little methodological support has been provided so far for the generalization from empirical data. Generalizability, however, is a major concern in this field. It has been subject to a number of publications in recent years, but commonly accepted conceptualizations of generalizability and methodological guidance for the process of generalization are still missing. In order to address this problem, this paper investigates if and how generalization approaches from the literature and abstraction mechanisms from the field of conceptual modeling can be utilized to generalize from case study data. An explorative single case study on the development of service-oriented architectures provides the raw data for an exemplary application. The paper shows that abstraction mechanisms from the field of conceptual modeling can be used – in conjunction with generalization approaches from the literature – for the generalization of case study data and provides guidance on how to use these mechanisms. This increases transparency, traceability, and reliability of the generalization and might help to improve other qualitative research endeavors as well.

Keywords:

Generalization, Case Study, Qualitative Research, Abstraction Mechanisms, Service-Oriented Architectures, Service Identification

Introduction

Generalizability is widely discussed in different kinds of research. Therefore, it also constitutes a major challenge in qualitative information systems research, and quite a few frameworks and definitions from this and other fields of research exist (Smaling, 2003; Walsham, 1995; Lee and Baskerville, 2003). So far, a widely accepted conceptualization of what generalizability means is missing. As Lee and Baskerville (2003) point out, "qualitative IS researchers (…) have not yet broached conceptions of generalizability appropriate to their own research" (p. 221).

However, generalizability of research results is extremely important. Usually, scientific readers do not want to learn about specific subjects that researchers observed in one case (Babbie, 2010). The interest is on what one can learn from a case and how to generalize the observations of a specific subject in order to explain the findings from a theoretical point of view, to predict future occurrences, or to give guidance in terms of recommendations for similar situations that might occur in the future.

Due to accepted sampling procedures and data analysis methods, quantitative research is often considered more conducive to producing generalizable results (Stoddart, 2004). Babbie (2010) outlines three major problems qualitative researchers are faced with, i.e., the subjectivity of researchers, the small number of researched cases, and doubts about the representativity, and argues that they lead to an endless potential for biased sampling and abstractions.

However, prominent authors (e.g., Walsham, 1995; Lee and Baskerville, 2003; Yin, 2003) contend that generalization is desirable and possible, but that qualitative research must employ special kinds of generalization. In this respect, Williams points out that papers reporting on results of research using qualitative and interpretive methods frequently "will make generalizing statements about findings whilst not commenting upon the basis upon which such generalisations might be justified" (Williams, 2000).

Hence, to achieve status for their work, qualitative researchers should be more explicit about the way of generalization and adopt more formal methods.

In order to better argue for the generalization of case study data and to make traceable the basis upon which generalizations might be justified, this paper investigates various conceptualizations and approaches of generalization in qualitative research and their application in a research project. Furthermore, we propose and apply abstraction mechanisms from the research area of conceptual modeling to provide methodological support for generalization.

This paper will utilize a case study on the development of a service-oriented architecture (SOA) to demonstrate how generalization applying different approaches can be performed to develop several hypotheses and finally three model fragments.

The paper is organized as follows: Section 2 elaborates on case study research as part of qualitative research. Generalization is discussed regarding both qualitative research in general and our case study research in particular. The case analysis is presented in section 3. Section 4 discusses findings and limitations. Finally, section 5 presents the conclusions of this paper.

Case Study Research and Generalization

The Appropriateness of Case Study Research

We suppose that qualitative case study research can make a useful contribution to our research endeavor. Case studies are particularly relevant for research in its "early, formative stages" (Benbasat et al., 1987; Myers, 2009) which applies to the field of SOA (Luthria and Rabhi, 2009; Stebbins, 2001). Furthermore, Benbasat et al. (1987) state that case study research is especially appropriate for the study of information systems development, implementation, and use within organizations.

Our research goal is to understand and to explore how SOA development and service identification in particular are performed in real-life projects and to examine this in a real-life setting.

As a case study can be descriptive and explorative in nature, it is supposed to give insights into how SOA development is performed. Descriptions and explanations of why a phenomenon occurs are provided by giving insight into the "generative mechanisms at work" (Walsham, 1995) (p. 79) observed within the case data. With respect to our research goal, we derive hypotheses and models (model fragments) from the observations that have been made. Thus, this case study takes the first steps in developing theory.

Darke et al. (1998) distinguish single-case and multiple-case design (p. 277). A single case study might be appropriate for our purpose, but there are limitations to be considered. According to Darke et al. (1998), "where explanatory research is undertaken, a single case may provide the basis for developing explanations of why a phenomenon occurs, and these may then be further investigated by applying them to additional cases in other settings" (p. 281). Hence, in single case study research theoretical or analytical generalization is suitable, where case study results are used to develop theory (p. 278).

In our study the intention was not to generalize to another population but to a theoretical understanding and explanation. This resulted in models representing generic socio-technical processes. Since our case study is explorative in nature, it is well suited to show starting points for further research in this field.

It is intended to produce generalizable results. Hence, we will first outline different notions of generalization in qualitative research from literature (2.2). Thereafter, the approach to and conception of generalization applied in this case study is discussed (2.3) and the research process is described (2.4).

Generalization in Qualitative Research

In the discussion of generalization, we will first refer to the meaning of generalization in a colloquial sense. Thereafter, we are going to develop a framework describing the outcomes of generalization and different conceptualizations and ways to generalize that we identified in the literature. In subsequent sections, the discussion will refer to this framework.

According to Merriam-Webster's dictionary, "to generalize" means "to derive or induce (a general conception or principle) from particulars." Lee and Baskerville (2003) quote the Oxford English Dictionary which defines "to generalize" as "to form general notions by abstraction from particular instances." Hence, abstraction, derivation, and induction are closely linked to the phenomenon of generalization in a colloquial sense. They are typical activities or mechanisms within the generalization process, aiming at the development of general propositions which are “of a different kind from those developed on the first level of common-sense thinking which they have to supersede” (Schutz, 1954) (p. 270).

However, in science, there are different, partially conflicting notions and conceptualizations regarding the questions what generalization means and how to generalize (Byrne and Sahay, 2007). The most significant differences occur when qualitative and quantitative research is compared. Whereas the latter predominantly concentrates on statistical generalization, many authors deem this type of generalization inadequate for interpretative, qualitative research (Yin, 2003; Lee and Baskerville, 2003; Smaling, 2003). Even though Lee and Baskerville criticized researchers in the field of qualitative research (see introduction), sporadic attempts and a handful of approaches to conceptualize and apply generalization in research projects can be found in literature.

In the following, this section will provide a discussion of existing literature that deals with the outcomes and types of generalization (table 1). Throughout the paper, the authors will refer to table 1 where appropriate in order to show how different aspects of generalization are addressed and why some approaches have been chosen for our analysis.

Geertz (1973) promotes “thick descriptions” as the outcome of the research process. They provide details for phenomena in their specific context, are ‘thick’ because they embrace the meaning behind the mere observation, and thus might enable an assessment of similarities and differences between two or more cases. Yet, first and foremost, they do “not ... generalize across cases but … within them” (p. 26). This notion is akin to the “working hypotheses” introduced by Lincoln and Guba (1985). They argue that transferability depends on the suitability of working hypotheses that represent tentative propositions of situations and their similarities.

Source	Major concept	
Outcomes of generalization		
Geertz (1973)	“Thick descriptions”	
Lincoln & Guba (1985)	“Working hypotheses”	
Popay et al. (1998)	(Logical generalization to a) “Theoretical understanding”	
Walsham (1995)	Concepts	Specific implications
	Theory	Rich insight
Types of generalization		
Lee & Baskerville (2003)	Type EE: Generalizing from data to description	Type ET: Generalizing from description to theory
	Type TE: Generalizing from theory to description	Type TT: Generalizing from concepts to theory
Klein & Myers (1999)	Contextualization	Hermeneutics cycles
Stake (1995)	“Naturalistic generalization”	
Smaling (2003)	“Analogical generalization”	
Eisenhardt (1989)	Grounded Theory techniques for data analysis; enfolding literature	
Yin (2003)	“Analytic generalization” / “generalizing to theory”	
Stoddart (2004)	Generalization about “generic social processes”	
Hedström & Ylikoski (2010) and Woodward (2002)	Generalization by detecting “causal mechanisms” / “causal generalizations”	

Table 1: Approaches to Generalization

Whereas Geertz as well as Lincoln and Guba are rather critical towards the possibility of generalization from a specific situation to others, Williams (2000) presumes that working hypotheses and thick descriptions at least “take the form of speculative generalisations” (p. 212) and are – what he calls – “moderatum generalisations”. In this type of generalization, aspects of a situation “can be seen to be instances of a broader recognisable set of features” (p. 215). He distinguishes moderatum generalizations from statistical and total generalizations (total generalization means that one situation is an instance of a general deterministic law that governs another situation as well).

Popay et al. (1998) refer to the different outcomes of qualitative and quantitative generalization. In qualitative research, “the aim is to make logical generalizations to a theoretical understanding of a similar class of phenomena rather than probabilistic generalizations to a population” (p. 348). They point out that in order to reach a theoretical understanding through logical generalization, one needs to apply methods different from the ones in quantitative research.

Along these lines, Tsoukas (2009) refers to Znaniecki and points out that while both forms tend to reach general and abstract truths, quantitative/statistical induction abstracts by generalization, whereas qualitative/logical induction generalizes by abstracting.

Walsham (1995) presents four types of generalization from case studies which are also outcomes in the aforementioned sense: Development of concepts, generation of theory, drawing of specific implications, and contribution of rich insight. They encompass fairly different results that can be obtained from case study data. On the one hand, he is explicit about the characteristics of these outcomes when he points out that they “should be viewed as 'tendencies', which are valuable in explanations of past data but are not wholly predictive for future situations,” and they are “explanations of particular phenomena (…) which may be valuable in the future in other organizations and contexts” (p. 79). On the other hand, he does not elaborate on the process of generalization in his paper and an extension (Walsham, 2006).

Whereas literature provides different descriptions of different outcomes of the generalization process, it does not offer a clear picture of how to generalize. A methodological guidance for the way the mentioned outcomes can be obtained or how to abstract, derive or induce in different situations is missing. However, we can identify hints, and orientation in the related literature that helps to develop a frame of understanding.

In order to structure the ways generalization can be performed, Lee and Baskerville (2003) develop a framework of four different generic types of generalization. On the one hand, empirical statements are generalized to either other empirical statements (Type EE: Generalizing from data to description) or to theoretical statements (Type ET: Generalizing from description to theory). On the other hand, theoretical statements are generalized to either empirical statements (Type TE: Generalizing from theory to description) or to another theoretical statement (Type TT: Generalizing from concepts to theories), respectively.

Klein and Myers (1999) offer a generic description in their principle #4 “The Principle of Abstraction and Generalization:” They describe the way of generalizing as “relating the idiographic details (…) to theoretical, general concepts” (p. 72). To do so, they recommend “contextualization” and “hermeneutic cycles” (their principles #1 and #2) as two important means. Since Klein and Myers’s aim is to propose a useful set of principles, along with their philosophical rationale, there is no detailed description provided.

Even though the principles and generic types are described in some detail and various illustrative examples are given, Lee and Baskerville as well as Klein and Myers do not intend to give guidance in terms of an applicable research method.

Based on the relationship between the reader’s experience and the case study itself, Stake (1995) argues for an empirically-grounded generalization that he calls “naturalistic generalization.” Accordingly, the case data can be understood and interpreted by readers more comprehensively if it matches their experience. In this case, the generalization emerges when the reader recognizes similarities in the case study details and finds descriptions that resonate with his own experience.

Smaling (2003) deals with the problem if and how findings from one case study can be transferred to another. He argues that “analogical generalization” – in contrast to inductive generalization – is “plausible when there are solid arguments that, when a particular researched case has characteristics which are relevant for the research conclusions, another case that has not been researched also has these relevant characteristics” (p. 57). This includes case-to-case generalization as well as exemplary generalization. Smaling makes one important assumption, namely that generalization is more firmely based “for the more one knows about similarities and differences between a case that has been researched and one that has not” (p. 55). His approach is partially in line with the notion of Lincoln and Guba (1985) and their idea of transferability and fittingness.

Eisenhardt (1989) provides useful procedural knowledge (a “roadmap”). She describes a process of building theory from case study research. However, regarding the most important step analyzing data she points out: “Analyzing data is the heart of building theory from case studies, but it is both the most difficult and the least codified part of the process” (p. 539). She advocates the application of grounded theory techniques and approaches to qualitative data analysis in order to support the emergence of theoretical categories and concepts. In addition, she recommends “enfolding literature” to improve the generalizability of the research findings by “tying the emergent theory to existing literature” (p. 545).

Accordingly, Yin (2003) suggests an “analytic generalization” with the goal “to expand and generalize theories” (p. 10). He argues that single case studies “are generalizable to theoretical propositions and not to populations or universes” (p. 10). This corresponds with the “Type ET Generalizability” in Lee and Baskerville (2003) and the “logical generalization to a theoretical understanding” of Popay et al. (1998). Following his approach, theory also becomes a vehicle for generalization. This, of course, is only applicable where appropriate theories exist.

Stoddart (2004) criticizes “folk notions of science (…) entwined with the positivist tradition” and advocates the idea of “generic social processes.” Based on previous work by Becker (1990) and Prus (1994), he abandons claims to generalizability about populations. Instead, he focuses on generalizing about “generic social processes” in order to “see how they play out in potentially diverse social settings” (p. 308).

In newer social science, some authors pursue a similar idea (Hedström and Ylikoski, 2010; Tsoukas, 2009). This attempt to generalize is the delineation of causal mechanisms. Generalization occours by “opening up black boxes and making explicit the causal cogs and wheels through which effects are brought about” (Hedström and Ylikoski, 2010) (p. 54). In this approach, researchers attempt to identify possible mechanisms and to turn them into a plausible mechanism through the collection of empirical evidence about the assumed entities, activities, relationships, etc. “What separates proper mechanism-based explanations from mere mechanism-based storytelling is this kind of rigorous checking of the assumptions upon which the mechanism schemes rest" (Hedström and Ylikoski, 2010) (p. 53). In this domain, Woodward (2002) puts forward the idea of "causal generalizations" by providing an account of explanatory relevance.

These ideas seem to be very similar to Bhaskar's "generative mechanisms" which have been adopted by Walsham (1995) but, interestingly enough, attract considerable interest in modern sociology.

In the following sections, we apply different approaches to generalization presented in table 1. In addition, we introduce techniques from conceptual modeling for methodological guidance in qualitative research.

Research Paradigm and Generalization in Our Case Study

There is a commonly drawn distinction in case study methodology between positivist and interpretivist research (Doolin, 1996; Darke et al., 1998). Even though this distinction is commonly used, there seems to be a loss of clarity in what makes the difference. Weber (2004), for example, argues, that "many, if not all, of the alleged metatheoretical differences between positivism and interpretivism are spurious" and that the real differences "lie more in the choice of research methods" (p. x). Others contradict this claim by explicating the "functional outcomes of research," which is basically the distinction between "understanding" and "explanation" (Hovorka and Lee, 2010).

Understanding, which is associated with interpretivism, is conceptualized in a subjective fashion. Researchers should develop an understanding of the subjective understanding of the participants' own understanding (Lee, 1991). In doing so, he or she is closely connected to the thoughts and motivations of the human objects under study and, in addition, offers an interpretation of and for human conduct (Doolin, 1996), which is also called "interpretive understanding of the subjective understanding."

In contrast, explanation in positivism refers to an observing researcher's formal position and is guided by the criteria of the natural science model (Darke et al., 1998; Lee, 1989). Accordingly, the constructs and variables used belong to the researcher and are not part of the subjects' experience (Hovorka and Lee, 2010). This leads to a positivist understanding, providing explanation of the empirical reality.

Despite these basic differences, scholars argue that the two perspectives can be mutually supportive, rather than mutually exclusive (Lee, 1991) and that the choice of techniques and methods to improve scientific knowledge should be based on their specific strengths and weaknesses (Weber, 2004). For that reason, Hovorka and Lee (2010) and Lee (1991) demonstrate how the two approaches can be integrated and how the linkage between explanation and understanding can be used to improve knowledge creation. Basically, this integration is viable because the interpretative understanding may provide the basis on which to develop the positivist understanding. Thus, the positivist interpretation of the interpretive understanding will lead to a different theoretical explanation.

However, it also has to be mentioned that previously interpretative researchers have already argued in a similar manner: Insofar as research begins with statements of particulars and ends with a general statement (a theoretical explanation), this reasoning process is a form of generalization. “Hence, generalizability is an essential feature of interpretive research that endeavors to provide theory and not just description.” (Lee and Baskerville (2003); with reference to Geertz (1973) and Yin (2003)).

In our case study, we make use of the two perspectives, as they are mutually supportive, to analyze data and elaborate on patterns and mechanisms, and hereby to generalize by formulating hypotheses and designing model fragments. According to Lee (1991), “understanding has at least two specific meanings. In its first sense, understanding refers to “understanding” as the process by which people in everyday life come to interpret and, therefore, to understand and guide themselves in their world. However, the observing social scientist is also one such person, albeit with different cognitive motives” (p. 348). Thus, the subjective meanings that give rise to the behavior of the people that are studied play a significant role in our research process. They are reflected throughout the whole research process and can be traced in section 3. However, with every generalization, the researchers more or less move away from subjective meanings in order to structure the self-perception of the project members.

Our conceptualization of generalization for the case study at hand is as follows: Our goal is to reach a higher level of abstraction by identifying theoretical, general concepts in a first step. This is done by relating the idiographic details and the subjective understanding to concepts in order to get from data to observations and – in a second step – from observations to theory which is in line with Lee and Baskerville’s (2003) approach (table 1). Our notion of theory is adopted from Gregor (2006), who defines theory as “statements providing a lens for viewing or explaining the world” (p. 4).

A critical issue remains how data and concepts can be related. The approaches presented in table 1 propose that these relationships can be drawn by “naturalistic generalization” (Stake 1995) or “analogical generalization” (Smaling, 2003). The former is not used because we do not agree that the interpretation and generalization should be made by the reader alone. The latter approach refers to case-to-case generalization, which is not our primary goal. An “analytic generalization” as proposed by (Yin, 2003) is not chosen because there are no commonly accepted theories in this field of research.

Instead, we apply techniques of the grounded theory methodology and interpretative techniques; in later phases, we also compare our findings with conflicting and similar literature (see 2.4). This corresponds to logical generalization aiming at the development of a theoretical understanding and a mode of induction described by Znaniecki as “generalization by abstraction.”

Additionally, in search for guidance and methodological support, we identified abstraction mechanisms of conceptual modeling as a helpful means because they aim at identifying "useful abstractions of the similarities of classified phenomena" (Parsons and Wand, 2008), an aim very similar to the one of qualitative research. Abstraction mechanisms are generic relationship types with a defined semantic. This helps with linking data and concepts residing on different levels of abstraction.

We argue that in qualitative research these mechanisms will support the grouping and structuring of details and concepts. Hence, in terms of Eisenhardt (1989), abstraction mechanisms are "structured and diverse lenses" which help to identify concepts and categories and transparently structure "similarities and differences" (Geertz, 1973; Smaling, 2003).

Literature offers a range of abstraction mechanisms (Goeken, 2006; Olivé, 2007; Mattos, 1989; Analyti et al., 2007). Two of them will be applied to the case study data in the following sections:

- Classification relates instances with a type, or to be more precise, it consists of determining the types which an object is an instance of (semantic: "has type"/"type of"). Instances have common properties and are assembled into a new entity type for which uniform conditions hold. Hence, a type is an abstraction representing instances on a higher abstraction level.
- Aggregation (composition) defines part-whole structures by describing that the whole is a composite formed by parts. Hence, aggregation is an alternative way of forming an abstraction (on a higher abstraction level) with the semantic "part of" (also called meronymic relationship/holonymic relationship).

The application of other abstraction mechanisms that we do not utilize in this paper (e.g., specialization (semantic: "kind of"/"is a"); grouping (semantic: "member of"); roles (semantic: "role of"); materialization (semantic: "materializes") will be subject of future research.

In the following, we will use aggregation and classification as a means to abstract from idiographic details and concepts to categories. Furthermore, the induction of categories ("types" and "aggregations") supports inferences about non-observed properties. As they are abstract placeholders, they enable and guide the derivation of alternative concepts that could not be observed in the original data. In doing so, researchers leave the firm empirical foundation. However, by making this procedure as transparent as possible, the creation of abstract types, and thus the generalization by abstraction, becomes traceable and can be justified.

One further goal is to identify generic socio-technical processes - comparable to "generic social processes" and "causal generalizations" (see table 1) - which are abstracted formulations of social behavior and the interplay of social and technical aspects. These can be interpreted as "middle range theories" because they are sufficiently abstract to be applied to different contexts but do not offer a set of general laws (Hedström and

Ylikoski, 2010) (p. 61). Middle range theories allow contextual explanations and "lie between the minor but necessary working hypotheses that evolve in abundance during day-to-day research and the all-inclusive systematic efforts to develop a unified theory that will explain all the observed uniformities of social behaviour, social organization and social change" (Merton, 1967) (p. 39).

In order to prevent excessive overestimation of generalizability, the main findings of our research (e. g., hypothesis and model fragments which form the "generic social processes" and "causal generalizations") "should be viewed as 'tendencies', which are valuable in explanations of past data but are not wholly predictive for future situations" (Walsham, 1995) (p. 79). Subsequently, developed theories should not be seen as "proven statements" but rather as "well founded but as yet untested hypotheses" (Lee and Baskerville, 2003). This parallels the notion of "moderatum generalization" (Williams, 2000 and section 2.2).

Research Process of Our Case Study

Following Eisenhardt (1989), we designed a research process for the study at hand which is depicted in figure 1.

The process commences with the explication of "a priori constructs." Furthermore, the first activity includes creating an initial research question and more specific questions of investigation (see section 3.2). This is an important first step to guide the case analysis and focus efforts.

In a second activity, an appropriate case has to be selected. Since the objective of this case study is to enhance understanding and develop theory rather than testing it, the chosen case does not have to be representative. One goal of this study is thus the generation of theory and not its justification through testing. A theoretical sampling though is "particularly suitable for illuminating and extending relationships and logic among constructs" (Eisenhardt and Graebner, 2007). Hence, we intentionally choose an extreme case that will be described in section 3.1. As outlined in table 1, a detailed description and comprehensive background information is important to support the "empirically-grounded" and "naturalistic generalization" promoted by Stake (1995) and to support the principle of contextualization by Klein and Myers (1999).

Multiple data collection methods such as interviews and analysis of documentation are used in the third activity (see section 3.3). In order to have a solid basis to build upon, collected data has to be gathered from multiple sources of evidence to underpin the completeness and correctness of data. This triangulation of data is important for the reliability of the case study's outcomes (Yin, 2003). Flexible and opportunistic data collection methods allow for reacting to emergent themes by adjusting data collection when necessary (Eisenhardt, 1989).

The fourth activity – analyzing data – begins with the grounding of our observations, as described in section 3.4. This is followed by the shaping of hypotheses through generalization, which is the subject of section 3.5. In the presentation of our research, we distinguish between observations and hypotheses (analyzing data and generalization, respectively) in order to enhance transparency and traceability of the procedure, even though they are closely interwoven.

Activity four is dedicated to properly grounding concepts and relationships. Their identification is conducted by employing techniques from grounded theory (see Eisenhardt (1989) in table 1) and interpretative techniques (see Walsham (1995) and Boland et al. (2010) for hermeneutical exegesis in IS). Even though we do not use the coding techniques to their fullest extent, the general approach and respective tools support the assignment of statements from the interviews and documents to concepts. The goal is to detect relevant particulars within the idiographic details and to discover underlying reasons for why concepts, relationships or patterns exist.

This activity results in observations which are judgments or inferences from what we observed (adapted from Merriam-Webster). These judgments are a first abstraction from the raw data, labelled “first-order concepts” by van Maanen (1979). Second-order concepts (the concepts forming our observations) are “notions used by the fieldworker to explain the patterning of the first-order data” (van Maanen, 1979). Hence, they move away from the subjective understanding (Hovorka and Lee, 2010) and 2.3).

While concepts and observations are still closely linked to the idiographic details, in activity five (generalization), we generate hypotheses consisting of abstract categories. Extending van Maanen’s terminology (1979), we might label these categories, which form hypotheses, third-order concepts, which thus represent abstractions of abstractions of first-order data. We generate these hypotheses by relating concepts described in the observations to categories applying abstraction mechanisms adopted from conceptual modeling. The categories and relationships between them should apply to multiple situations and hereby reach a higher level of generality.

The sixth activity consists of a comparison with related work, so called “enfolding literature” (Eisenhardt, 1989).1 It aims at comparing opinions and positions found in related literature with the observations we made and improving the hypotheses. This results in hypotheses “with stronger internal validity, wider generalizability, and higher conceptual level” (Eisenhardt, 1989). In doing so, we try to turn possible mechanisms into plausible mechanisms (Hedström and Ylikoski, 2010) (p. 52).

Analyzing data (4) and generalization (5) are closely interwoven and both supported by the analysis of literature. According to Gadamer (1976), “the harmony of all the details with the whole is the criterion of correct understanding” (p.117), and in a number of it-

[1] Due to the focus on generalization, the extensive discussion of literature related to every hypotheses and model is omitted in this paper. It can be found in Börner et al. (2012).

erations “a complex whole of shared meaning emerges” (Klein and Myers, 1999). This is represented by the iterative layout of activities four to six which accords to the hermeneutic cycles promoted by Klein and Myers (1999). The focus is on constantly comparing hypotheses, data, and competing and similar literature iterating towards a set of hypotheses which closely fits the data (Eisenhardt, 1989).

Finally, activity seven integrates several hypotheses to build model fragments (section 3.6). Through another generalization, these fragments have an even higher level of abstraction and are thus more remote from the first-order data.

Figure 1 illustrates the increase of abstraction with each activity performed moving from left to right. Considering semantic levels, model fragments encompass both the level of instances and the level of types. They incorporate categories used in our hypotheses and also concepts – including those we were not able to observe in our case study.

Generalization in an SOA Case

Case Description

In the context of the project “Ad-hoc DAta Grids Environments” (ADAGE), researchers at the University of New South Wales (UNSW) implemented a service-oriented architecture for SIRCA, a data provider. The authors of this paper did not participate in the actual SOA implementation project but analyzed data about it ex post as described in the following sections. The project aimed at providing customers an easier retrieval and analysis of heterogeneous data from different sources (grid environment) spontaneously in an ad-hoc fashion. SOAs were not the focus of this software implementation project but service orientation was chosen as the preferred architectural paradigm. Thus, services (and their identification) were used as a means to meet the company’s requirements rather than being the subject of analysis themselves.

SIRCA provides a huge data repository containing historical financial market data such as news and trading data. Its aim is to supply this data to researchers especially at Australian and New Zealand universities.

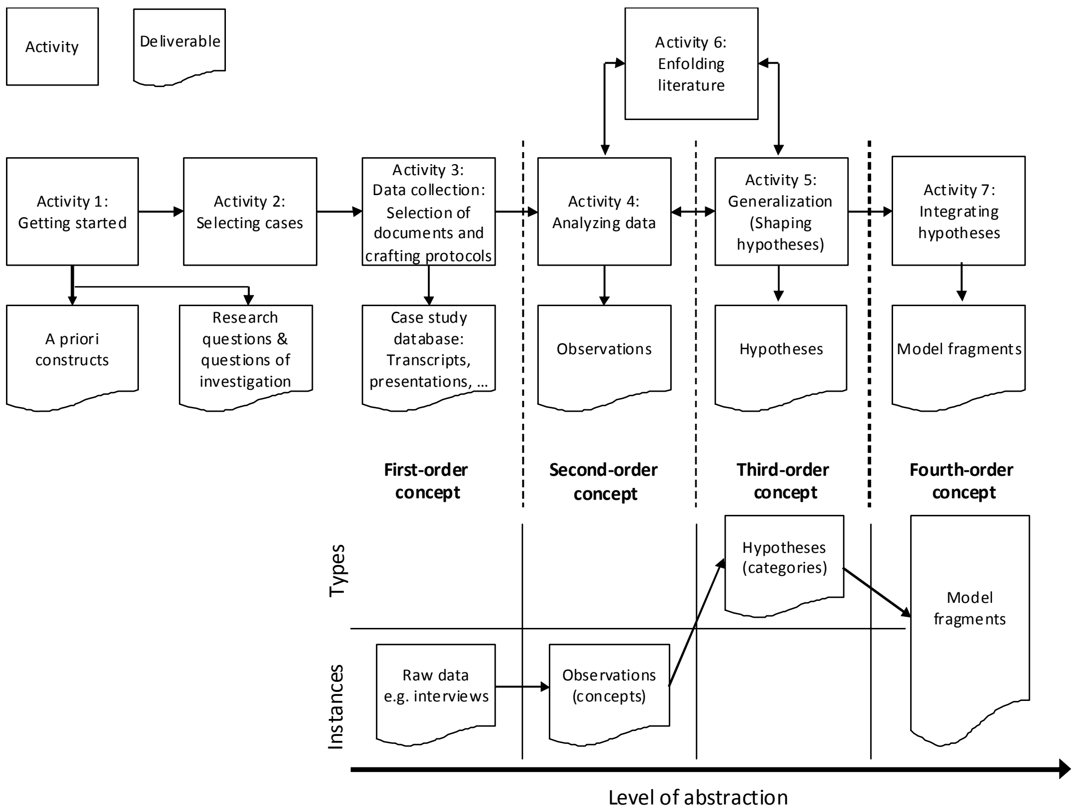

Figure 1: Research Process and Deliverables of the Respective Activities

In ADAGE, services were created based on the available data. This implied a technical understanding of services. The company’s employees were not thinking in terms of business processes, so that no model that could have been analyzed in the course of service identification was delivered. The company’s management, however, had some requirements in mind that should be fulfilled by services. Unfortunately, these were not documented, which makes traceability difficult. Requirements were communicated to the project team in scheduled weekly meetings and workshops. Service candidates were identified on the basis of these meetings and prototyped. In an iterative and incremental approach, the functionality of these candidates was adjusted to finally meet the company’s requirements. A close collaboration between the SIRCA’s research and development department and the university’s project team was a key to ensure the successful identification of services.

First and foremost, the search for services was driven by the idea to retrieve and integrate data from different sources. In a second step, project members identified which services could support researchers in analyzing data, e.g., building time series of financial data. Clearly, this was a requirements-driven bottom up approach. Goals included the provision of a graphical user interface (GUI) to customers, enabling them to directly invoke services in an ad-hoc fashion to analyze financial market data. Certainly, SIRCA’s case is not a typical example of service identification projects. Due to its rather extreme character, it helps to identify possible instances of situational factors that usually cannot be found in typical cases.

Since this paper focuses on methodological aspects, we partly omit more detailed descriptions of the case study data. A comprehensive discussion and description including interview guidelines, transcripts, the raw data leading to our observations, etc. can be found in Börner et al. (2012).

Getting Started

As illustrated in figure 1, outlining the theoretical background in the sense of "a priori constructs" is the first step of our research process. Since the research question is the foundation for the analysis of every case study (Eisenhardt, 1989), our research process continues with the formulation of a research question. Shortly after the beginning of the case study, it became obvious that a single focus on service identification might not deliver satisfactory results in the ADAGE case. Thus, we extended the scope of our analysis to the early stages of the SOA lifecycle, namely the design and development of services and the service-oriented architecture in ADAGE.

Some authors have already developed methods to support identification, design, and development of services (for an overview see Börner and Goeken (2009b), Kohlborn et al. (2009b) and Birkmeier et al. (2009)). However, little is known about service identification and the early stages of the SOA lifecycle in real-life projects. Thus, in order to guide us through the analysis, we formulate the following initial research question:

How does SOA development and service identification work in real-life projects?

The rather generic nature of our research question necessitates the formulation of further, more specific questions, the questions of investigation. These help to focus our analysis and support, for example, the hermeneutic analysis of the interview transcripts. The following questions of investigation are reflected by the guidelines used for the conducted interviews.

- Which circumstances influence the proceeding of the service identification process?
- Did the understanding of services change/develop in the course of the project? Was there a change of the skill level of the project participants?
- How was the SOA implemented technically? Which software development techniques had been used?

Data Collection

Written documentation, i.e. various sorts of electronic files, and interviews are major sources of our data collection process. 19 presentations that were held between December 2006 and November 2009 provide a good overview of the general proceeding, achieved objectives, and next steps required at the time of the presentation. Furthermore, official progress reports give a structured overview of how the project was proceeding. Due to their much more formal character, these reports offer fewer insights into actual work practices, such as the identification of services.

A considerable number of published and unpublished papers have been authored by project team members in the course of the project. The variety of publications in which the papers have appeared shows the scientific and practical value of the project's outcomes. Additionally, various websites about the project or involved parties are used to retrieve background information. Some newspaper articles about the project could be found as well and are added to the case study database.

In order to retrieve additional information, particularly about the identification of services in the ADAGE project, four interviews are conducted. Three of the interviewees are project team members at the university. All of them were involved in identifying and designing services in the project. Another interviewee is a representative of the company that implemented the SOA.

The interview questions are open-ended. However, guidelines with questions are used in each case. For the three project team members at the university, the same question guideline applies, whereas a different one is used in the interview with the company's representative. The interviews serve as a valuable means to capture subjective meanings that are essential to our interpretivist approach. All labels of concepts used in our observations are retrieved verbatim from the interviews.

Analyzing Data

Based on the collected case study data, we first strive to investigate the subjective understanding of the project members. From this understanding, observations residing on a higher level of abstraction are induced (second-order concepts). Due to space limitations, we cannot give detailed insight into the subjective understanding. This is done in a working paper (Börner et al., 2012) incorporating a plethora of quotations from documents and interviews.

The ADAGE project members perceived themselves as rather innovative and archaic in a creative environment with little constraints. Due to the project setting, i.e., a public private partnership, the pressure regarding time and budget was comparably low. The developers used their freedom to follow a flexible free-style approach based on their available skills. According to the members' perception, they were able to compensate for the absence of any formal method in their development process through extensive communication. Furthermore, based on the mainly technical skills of the project members, the proceeding in the project was seen as a creative technically-driven trial-and-error endeavor.

Our observations attempt to structure the subjective meanings that were found in the raw data and support the perception of the project situation. Grounded in document studies and interviews, we arrive at several concepts that we consider as being of importance in the course of the ADAGE project, e.g., we regard the subjective understanding mentioned above as the concept "absence of a formal method." In doing so, and applying the contextualization principle (see table 1), we seek to present the subject matter in its context "so that the intended audience can see how the current situation under investigation emerged" (Klein & Myers, 1999) (p. 73).

All 16 observations describing the concepts and their relationships are shown in table 2 and are the result of activity four.

Shaping Hypotheses

In this next activity we further abstract the observations by generating hypotheses consisting of categories which are based on the concepts observed.

The concepts we identify in the observations can be regarded as instances, and thus we can use abstraction mechanisms to generalize from these observations. The resulting types or compositions form the components of our hypotheses. Hence, by applying abstraction mechanisms, we create abstract categories on the type-level, the ‘third-order concepts’ (figure 1) we have labeled categories (e.g., we can classify the concept “fine-grained services” as an instance of the category “granularity”). In so doing, we utilize mechanisms from conceptual modeling that guide the generalization by abstraction which we regard as a logical generalization.

As the categories are abstract placeholders, in other settings they will have different concepts than those we have observed in our case study. Granularity, e.g., also has the instance “coarse-grained services.” By incorporating alternative concepts, we clearly move beyond the details observed and further generalize the findings (see section 3.6).

Observation
1. The generous funding significantly influenced the way of service identification.
2. The company’s small size significantly influenced the way of service identification.
3. People’s skills significantly influenced the way of service identification.
4. The implemented SOA satisfied users, although no formal method for the identification of services was used.
5. Little SOA experience led to a technical understanding of services.
6. Personal communication was vital to meet user requirements.
7. A technical understanding of services led to very fine-grained services.
8. A technical understanding of services led to a bottom up direction of analysis.
9. The implemented fine-grained services provided flexibility for users and developers at the same time.
10. Fine-grained services supported reusability.
11. Orchestration of services significantly contributed to flexibility.
12. Fulfilled user requirements led to user satisfaction.
13. Identification and development of services were incremental and iterative processes that supported fulfillment of user requirements.
14. The flexibility provided by services led to user satisfaction.
15. User satisfaction was a success measure.
16. Reusability was a success measure.

Table 2: Observations from the Case Study

In the following, we present our hypotheses and describe how the generalization and the merging of the observations is performed, applying the abstraction mechanisms described in section 2.3. Table 3 illustrates our observations and the resulting hypotheses. Due to space restrictions, only hypotheses 1, 3, 5 and 6 will be described in detail because they are further used in the models that will be developed in section 3.6.

Hypothesis	Observation
1. Context factors significantly influence the way of SOA development and service identification.	1. The generous funding significantly influenced the way of service identification.
	2. The small company size significantly influenced the way of service identification.
	3. People's skills significantly influenced the way of service identification.
2. Not using a formal method for service identification does not necessarily lead to a failure of SOA projects.	4. The implemented SOA satisfied users, although no formal method for the identification of services was used.
3. SOA project experience leads to a different understanding of services and thus affects success measures.	5. Little SOA experience led to a technical understanding of services.
4. Personal communication can substitute utilization of formal methods.	6. Personal communication was vital to meet user requirements.
5. The understanding of services affects both the granularity of services and the direction of the service identification approach.	7. A technical understanding of services led to very fine-grained services.
	8. A technical understanding of services led to a bottom up direction of analysis.
6. The right granularity of services can affect multiple success measures and depends strongly on the project at hand.	9. The implemented fine-grained services provided flexibility for users and developers at the same time.
	10. Fine-grained services supported reusability.
7. The success of SOA projects is expressed through technical and business-oriented success measures.	15. User satisfaction was a success measure.
	16. Reusability was a success measure.

Table 3: Hypotheses Related to Observations

Figure 2 shows how the relation between concrete concepts (observations) is mapped to an abstract view that generalizes from the ADAGE case. While the lower part links instances through observations, the upper part shows the relation between types and the resulting hypotheses. Concepts (e.g., fine-grained services, reusability) that could be observed in our case study are linked through observations (table 2), e.g., O10 "Fine-grained services supported reusability". On the next abstraction level, categories such as granularity are related through hypotheses like number 6 "The right granularity of services can affect multiple success measures and depends strongly on the project at hand." These categories reflect phenomena that can be found not only in the ADAGE case but in SOA implementation projects in general. In addition, they represent „causal mechanisms" and „causal generalization" (Hedström and Ylikoski, 2010). Thus, the hypotheses are a generalization of our observations as shown in table 3.

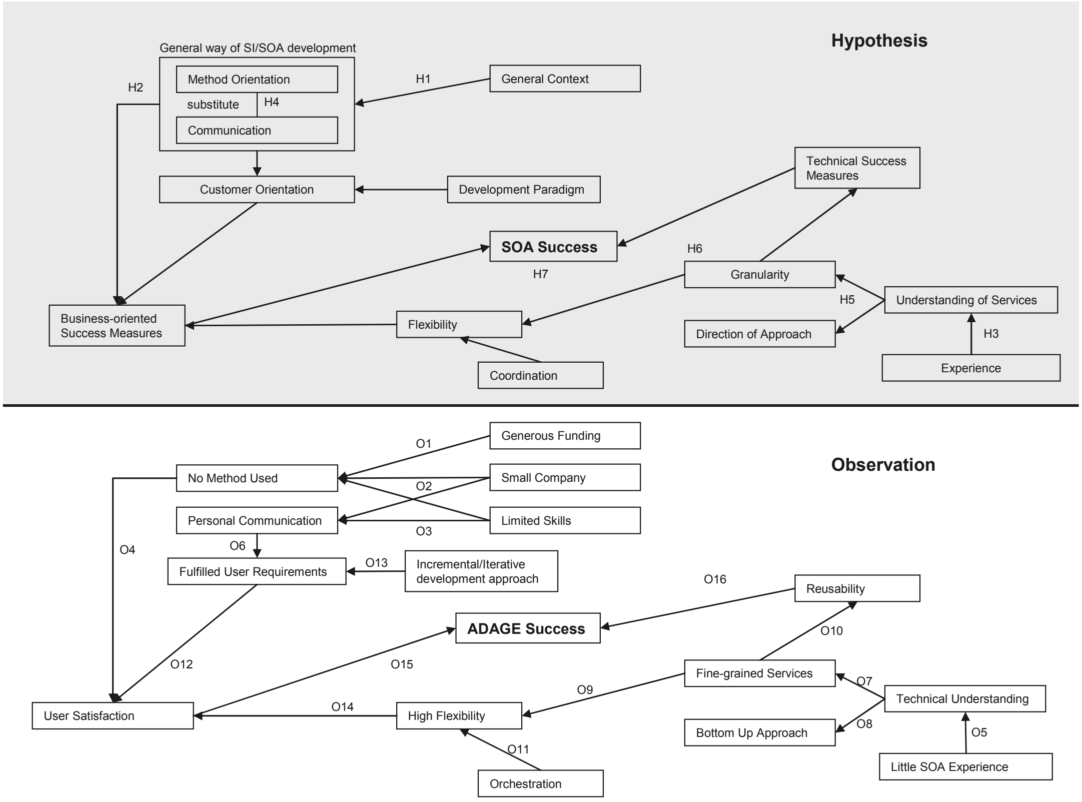

Figure 2: Interrelationships of Observations and Hypotheses

Hypothesis 1: The general context significantly influences the method of SOA development and service identification.

Certainly, there are many factors that affect the approach to service identification in general and the choice of certain methods in particular. Those factors that cannot be influenced by the project team are commonly called context factors (Bucher et al., 2007). We identify generous funding, small company size, and limited skills in ADAGE to be such context factors (observations 1 to 3).

At first, we use the abstraction mechanism ***aggregation*** to subsume the identified concepts under the new concept *ADAGE context*. Thus, generous funding, small company size, and limited skills have an "is-part-of" relation to *ADAGE context* as shown in figure 3. On the level of abstract categories these relations apply, respectively.

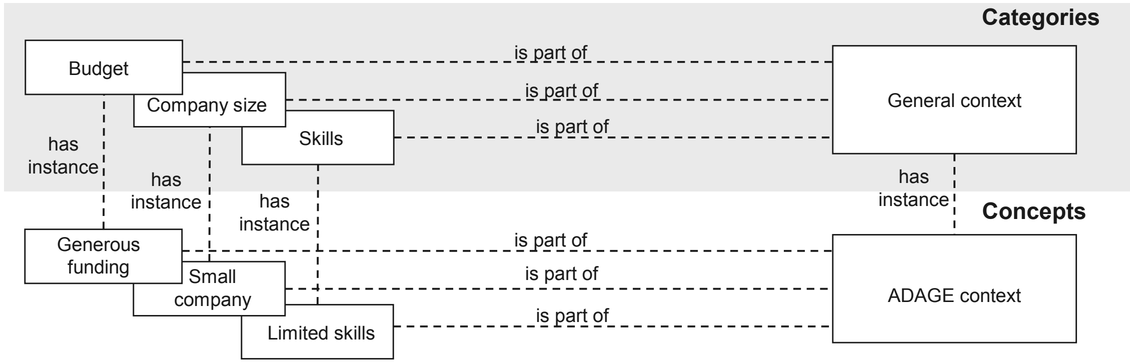

Figure 3: Applying Abstraction Mechanisms to Hypothesis 1

Applying the abstraction mechanism classification, we can generalize our concepts (instances) by determining abstract categories (types). For example, the category company size is the result of classification from the observed concept small company. Thus, there is a has-instance- (instance-of-) relationship between these two. Budget and skills are the other two categories that were directly derived from our observed concepts described in observations 1 to 3. The concept ADAGE context that is created by aggregation is transformed into the category general context by classification. All categories are related to respective concepts by "has-instance" relations. Particularly, the aggregation that sums up several concepts to the ADAGE context necessarily moves away from subjective meanings that served as the starting point for our analysis.

Our observations show that the combination of a small company and employees with limited skills (trumping the opportunities of a generous budget) leads to an absence of formal methods in service identification. This is in line with the subjective understanding of the project members (see 3.4) and certainly an extreme case. However, it seems to be plausible to assume that context factors significantly influence the way of service identification and, in the following, we will discuss the underlying mechanisms. Thus, we can e.g., tentatively hypothesize that large companies with highly skilled employees will use different methods for SOA development and the identification of services.

Hypothesis 3: SOA project experience leads to a different understanding of services and thus affects success measures.

In ADAGE, we observed that little SOA experience accompanied with the dominance of IT specialists resulted in a technical understanding of services (observation 5). Classifying the concepts observed, we assume that the level of experience influences the understanding of services. We therefore hypothesize that companies with a track record of service implementation tend to involve business departments early in the process of service identification. Therefore, the understanding of services is much more process-oriented. Through multiple cause-and-effect relationships shown in figure 2, success measures such as reusability are affected.

The task of planning a service-oriented architecture should be tackled by both business departments and an organization's IT division (Börner and Goeken, 2009b). Implementing an SOA is primarily a technical challenge. We assume that – like in the ADAGE case – many SOA projects underestimate the importance of properly planning an implementation. Hence, consideration of business processes is often poor, and employees of the IT division take the lead in these projects. Especially in companies that have little or no experience with the nature of services, the dominance of IT specialists usually leads to a technical understanding of services. More experience in this field might result in different outcomes. A project team might decide to use a process-oriented approach, a user-oriented approach, or even choose a technical approach if suitable. The most significant difference is that an experienced project team with a broader understanding of SOA is free to choose an adequate approach from a range of options. Unfortunately, this makes a prediction almost impossible. The only cause-and-effect relationship that can be established here is that more experience allows for different opportunities whereas little experience most certainly leads to a technical understanding.

Anderson et al. (2005) argue that the extent to which the enterprise architects, service developers, and operations project staff in the IT department are skilled (i.e., their experience and subsequently their understanding of services) is a critical factor for web service implementation. Looking at our observations 5, 7, and 10, we can further hypothesize that SOA experience indirectly influences reusability of services. This is confirmed by Baskerville et al. (2005), who show that "very few web services could be reused exactly as originally implemented" (p. 7) and thus, experience is essential for SOA implementations. Becker et al. (2009) state that "especially in the early phases of SOA maturity [i.e., with little experience] there is no experience about [what] a reusable service would look like" (pp. 7-8). This is another hint that SOA project experience tends to change the understanding of services and enhances reusability.

Hypothesis 5: The understanding of services affects both the granularity of services and the direction of the service identification approach.

In the ADAGE project, a technical understanding of services led to very fine-grained services and a bottom up approach stressing technical aspects of the underlying (data) infrastructure (see observations 7 and 8).

In order to develop our hypotheses, a classification leads to the abstract categories understanding of services, direction of approach and granularity. The category granularity, e.g., is the abstraction of fine-grained service, the instance that we are able to observe.

Firstly, we can hypothesize that a different understanding of services leads to different granularities. Moving on a scale from technical to business-oriented understanding, the identified services might become coarser-grained along this line. From a business perspective, services should support business processes or at least sub processes. Hence, they encompass more functionalities than an elementary service that, for example, extracts an address from a database. The latter will more likely be the outcome if service identification is conducted from a technical point of view.

Secondly, the direction of the approach is influenced by this understanding. Again, moving along the above scale will result in the application of bottom up approaches on one side and top down approaches on the other side. It is unlikely to end up with a pure instance of any of the approaches. Even in the ADAGE case, a very technical understanding did not result in a pure bottom up approach. However, the share of techniques that are typically assigned to top down approaches (such as strategic analyses) will increase with a business-oriented understanding of services. Accordingly, in their comparison of service analysis approaches, Kohlborn et al. (2009a) differentiate two types of SOA concepts, i.e., understandings of SOA. One of them is rather technical and, thus, delivers so-called software services while the other (business-oriented) one results in business services.

Hypothesis 6: The right granularity of services affects multiple success measures and depends strongly on the project at hand.

In service-oriented architectures, granularity of services is a widely discussed issue among researchers and practitioners alike. Our observations 9 and 10 indicate that fine-grained services positively contributed to both reusability and (indirectly) user satisfaction in the ADAGE case. Thus, we hypothesize that granularity indeed plays a major role for the success of SOAs in general since it potentially affects more than one success measure. The categories granularity, flexibility, and success measures that constitute hypothesis 6 were developed by classification. They all abstract the instances and concepts observed and, thus, have “is-instance-of” relationships to the concepts fine-grained services, high flexibility, user satisfaction, and reusability.

Whereas the ADAGE case shows clear advantages of fine-grained services, coarse-grained services might be the better choice in other settings. Some customers might not be interested in the flexible composition of services every time they use them. If they simply want to outsource part of a business process, they might prefer a coarse-grained service that encompasses all necessary functionalities and delivers a comprehensive result. Accordingly, Elfatatry (2007) argues that “the appropriate level of granularity for a service and its methods is relatively coarse. A service generally supports a single distinct business concept or process.” (p. 38) In the ADAGE case, fine-grained services enhanced both reusability and user satisfaction, i.e., technical and business-oriented success measures, respectively. In other cases, the granularity of services might be a trade off because effects on these two kinds of measures could be converse.

We argue that, due to different preconditions, the right granularity of a service has to be elaborated depending on the situation at hand. There is no silver bullet for right-sizing a service without considering the context of service implementation. Thus, a situation-specific approach to the choice of methods is important to provide for an adequate granularity.

Integrating Hypotheses

Through generalization of our observations, we create hypotheses as illustrated in figure 2. This is done by applying classification and aggregation, two abstraction mechanisms with a clear semantic (instance-of/part-of). Since they are applied upwards (e.g., a concept ‘is instance-of’ a category), the investigation is based on and refers to observed instances and respective concepts. Hence, both the resulting observations and hypotheses are still closely linked to the primary data we retrieved from documentation and interviews.

In the following, we will describe phenomena that move beyond this data and reach another level of abstraction. By performing another generalization and looking at causal mechanisms between concepts, we arrive at model fragments that were not necessarily obvious beforehand. Due to the fact that the categories are abstract placeholders and the hypotheses show possible and plausible causal mechanisms, we can now reason about further conceivable concepts in terms of alternative instances. This is supported by applying the abstraction mechanisms downwards (a category 'has instance' non-observed concept). Since we assume that the hypotheses describe plausible mechanisms and the placeholders as well as the abstraction mechanisms guide the derivation, this is not arbitrary but another generalization by abstraction. Because the observed concepts as well as the categories in the different hypotheses overlap, we are also able to combine the hypotheses in a model or in a small set of model fragments.

The model fragments we present in this section can best be interpreted as middle range theories (see section 2.3) because they have limited scope and are applicable to limited conceptual ranges, as they only grasp phenomena from the systems development / SOA domain. Furthermore, they refer to selected aspects in the realm of SOA development because we focus our analysis on the questions of investigation presented in 3.2. Our model fragments allow for further detailing and the formulation and discussion of alternative theories. They correspond to the "theories" introduced by Walsham (table 1), and are causal generalizations. In the following subsections, three model fragments which present an external, internal, and a success perspective are elaborated.

Contingency Model (CM)

Context factors such as budget, company size, and people's skills are consolidated in hypothesis 1. All of them ("general context") influence the application of methods, communication, and the general approach to implementing SOA projects. The contingency model in figure 4 includes our observations from the ADAGE case (underlined concepts) and alternative concepts which are derived by conceivable instantiation of the categories applying abstraction mechanisms. We assume that alternative instances of the mentioned categories will result in different ways of SI/SOA development, as depicted in figure 4.

In our model, the list of categories constituting the contingency model is limited to those we were able to derive from observed concepts. We assume that the category "general context" can also be extended by non-observed concepts and resulting categories. For instance, in a more business-driven SOA implementation project, the choice of a BPM software tool could influence the way of SI/SOA development. Since there is no BPM tool used in our case, we are not able to observe such an impact. Accordingly, the way of SI/SOA implementation will most likely include more categories than method orientation and communication which we are not able to observe.

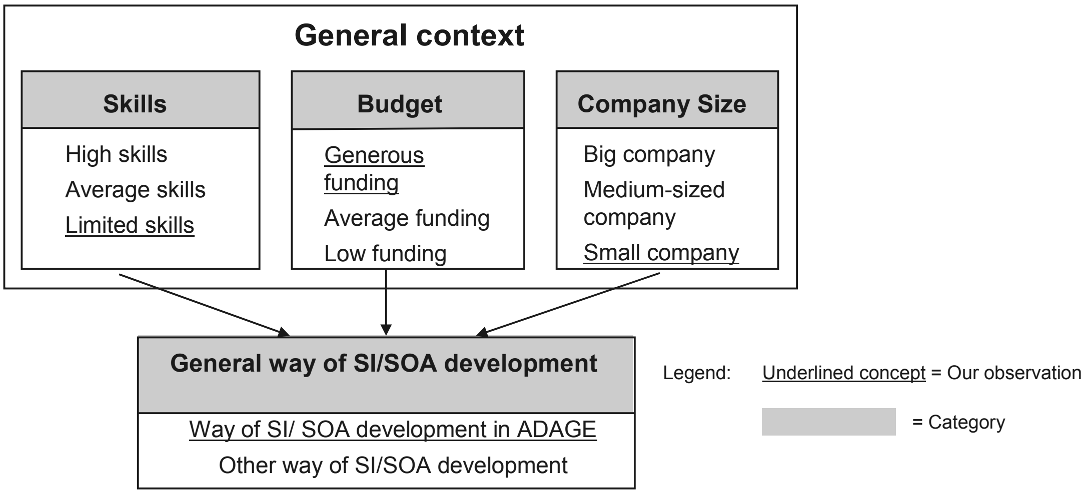

Figure 4: Influence of the General Context on the Way of SI/SOA Implementation (Contingency Model – CM)

The generalization in this contingency model follows Stoddart's (2004) notion. The general way of SI/SOA development is a socio-technical process and we generalize this process by abstracting from the observed instances. Due to the single case study design, we are only able to observe one way of SI/SOA development, i.e., the one in the ADAGE case. Subsequently, it is difficult to define more instances (concepts) of the category general way of SI/SOA development whereas additional instances for skills, budget and company size are easier to derive.

To the best of our knowledge, there are neither frameworks/models nor empirical studies that analyze situational factors in the realm of SOA. A first step towards a comprehensive list using more case studies has been made by Börner et al. (2011). Further empirical as well as conceptual research should strive to extract and develop models and causal patterns with respect to this category.

Model of Soft Factor Transition (MSFT)

The model fragment which ties together hypotheses 3, 5, and 6 is depicted in figure 5. Our observations (underlined concepts) and the resulting cause-and-effect chain show that a team with only little SOA experience produced a highly flexible service-oriented architecture and highly reusable services. In our hypotheses, experience leads to a distinct understanding of services (H3). This results in different granularities and directions of approaches (H5) and subsequently in a certain degree of flexibility and reusability (H6). Hence, the experience of project teams influences flexibility and reusability as well as the direction of the identification approach through a chain of cause-and-effect relationships as depicted in figure 5.

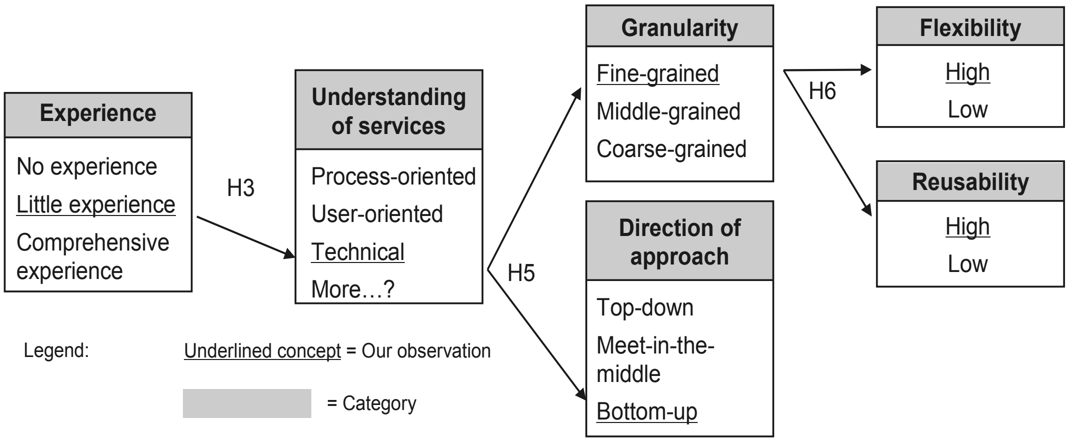

Figure 5: Causal Link of Experience and Flexibility/Reusability (Model of Soft Factor Transition – MSFT)

Finding further plausible and alternative instances for this MSFT is easier than in the case of the contingency model. The ***classification*** that is used to abstract categories from concepts is now inverted to identify new instances, i.e., concepts that could not have been observed in the case study. Again, as shown in figure 1, this step moves back from the type to the instance level but still increases the degree of abstraction from the raw data.

Outlining conceivable and alternative instances, we are able to describe project patterns as shown in figure 6. However, these patterns are not proven theories but rather hypotheses in terms of plausible tendencies. Theoretically, the number of possible patterns equals the number of all combinations of concepts. However, an identification of plausible patterns will be a contribution itself. Certain combinations of concepts might not be found in any pattern due to either conflicting goals (e.g., coarse-grained services and high flexibility) or incompatible preconditions and goals (technical understanding of services and top-down approach for their identification).

All major concepts of every category should be included but at the same time restricted in number as far as possible. Only those concepts that help to differentiate one project from another should be part of the framework. Identifying pattern could be impeded if concepts are too granular (e.g., a scale from 1 to 7 for reusability instead of “high” and “low”) or not disjunct (e.g., “technical” or “functional” understanding of services).

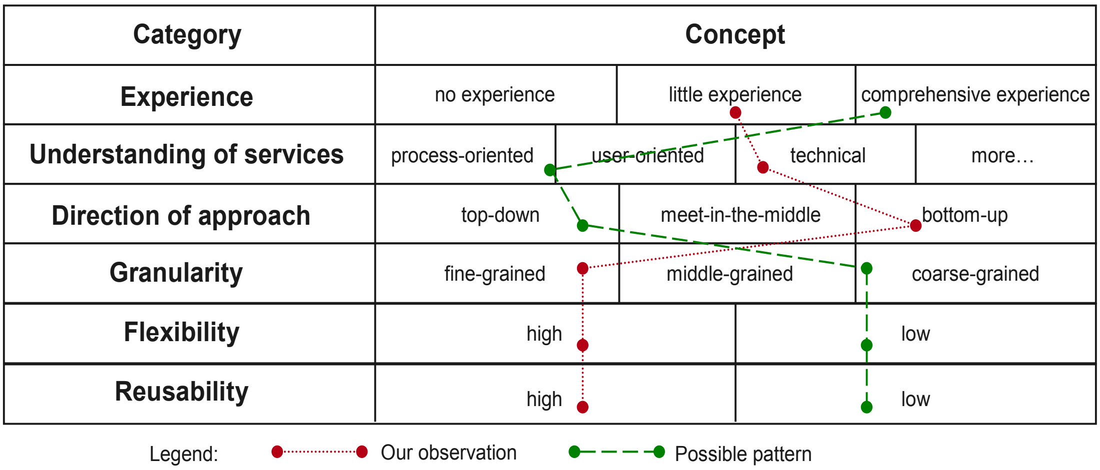

Figure 6: Project Patterns

Teams with little SOA experience and limited knowledge regarding business functions and processes are forced to rely on the experience they have, which is basically technical. We assume that developing an SOA based on the knowledge stemming from experience made with traditional, more technically-driven software engineering paradigms, for example, object-oriented programming or modules, results in fine-grained services. The understanding of services can be seen as the generative mechanism in this model fragment.

We could assume that a very experienced team might have a process-oriented understanding which leads to rather coarse-grained services that provide less flexibility. Since flexibility is regarded to be a big advantage of SOAs, this causal link is counter-intuitive. We would expect an experienced team to be able to reap the benefits of SOAs, including an enhanced flexibility. Figure 6 illustrates our observed pattern (dotted line) and the assumed possible pattern described above (dashed line).

However, we have to keep in mind that more experience does not necessarily lead to one specific understanding of services but enables an adequate choice of approach and understanding, as argued in hypothesis 3. A comprehensive experience, for instance, is not closely knit to a process-oriented understanding and could as well produce middle-grained services. A large number of patterns are thus conceivable. We argue that – depending on situation-specific circumstances in projects – the resulting patterns represent certain types of projects. An identification of such project types could lead to a targeted application of methods supporting SOA development and the identification of services. Project types can define situations in the sense of situational method engineering and would subsequently contribute to a more sophisticated service identification through the use of situational methods (Bucher et al., 2007).

Observations 7 and 8 show that the technical understanding of services in the ADAGE project lead to fine-grained services and an almost pure bottom up approach for identifying services. In this regard, the lack of experience once again leads to a rather object-oriented thinking which reveals another generative mechanism at work in our case study. In hypothesis 5, these observations are generalized. Undoubtedly, there is an effect of this understanding on granularity and the direction of service identification approaches. Actually, the question of granularity continues to be an open issue among scientists and practitioners alike and constitutes an active research area (Artus, 2006).

Success Model (SM)

Many observations and hypotheses are bound to the notion of success. Naturally, the primary objective of every project is to be completed successfully. This is also true for SOA implementation projects. But how can you evaluate the success of such a project? Our observations and hypotheses utilize several success measures that are influenced by a number of different concepts in our model illustrated in figure 2. In our observations, we find success factors such as reusability and user satisfaction. They are classified to technical and business-oriented success measures, respectively. Both success factors can be aggregated to the category SOA success. However, they take different view points and usually represent interests of different stakeholders.

On the one hand, reusability is a rather technical aspect. Subsequently, in most cases, it is pursued by the IT department that implements an SOA. The number of business processes that invoke a service and the frequency of invocation can serve as a unit of measurement. If the IT department can prove that a service is reused frequently in many processes, it is easy to argue that maintenance and development of this single service is more cost efficient than having the same functionality provided by multiple applications scattered across the organization.

On the other hand, user satisfaction is clearly a business goal. Users have little interest in cost savings or technical demands on the company's side. Instead, they appreciate flexibility and a timely satisfaction of their demands. Hence, business processes must be flexible and agile to guarantee a certain degree of user satisfaction. As argued before, a successful SOA can cater to these demands.

A number of success measures can be employed (see, e.g., the framework by Börner and Goeken (2009a)). All of them would serve as further instances of our success categories. Business departments can thus consider flexibility and time-to-market as important factors. From the technical side, autonomous services or a high degree of standardization might be important. In both cases, the measurement of success factors is far from trivial, and appropriate measures will have to be defined to operationalize the measurement of SOA success. To which extent technical and business-oriented success factors are aligned or not is an interesting question that is left to further research. Based on a literature review and expert interviews, Lee et al. (2010) identify 20 critical success factors in SOA implementation. However, their interrelationships and contribution to success are not clear.

Findings and Implications

In our analysis we identify factors and aspects (concepts and categories) relevant in the ADAGE project. With respect to our research questions our findings for SOA development projects are the following:

- We identified circumstances *(context factors)* that influenced the proceedings of the project. We were able to observe how some of these factors (e.g., funding) affected an SOA project and incorporated these observations into our hypotheses and model fragments.
- We found that the *understanding of services* and related skills significantly influence the way of SOA development. Several hypotheses and our models show that concepts such as granularity or the direction of the identification approach are more or less directly affected by the understanding of services (which is usually subject to change with increasing experience).
- Due to the absence of a method for service identification, we were not able to observe a stringent application of *software development techniques*. Software development was mostly intuitive and bound to few developers because of the limited scope of the project. An astonishing finding is that the "absence of a formal method" apparently has had no negative impact.

Overall, we have made 16 observations and subsequently developed seven hypotheses and three model fragments. Limitations stem from at least two sources:

- Due to the *single case study design*, our hypotheses and model fragments are limited and it has to be taken into account, that there are different concepts and categories to consider in more comprehensive models. To extend the models, further research is desirable.
- Additionally, the model fragments and hypotheses *must be seen as tendencies*, explaining plausible mechanisms. Their strength is that they describe possible cause-effect-relationships and give insight into the generative mechanisms at work. They are limited insofar as they are not tested yet. Hence, we neither can be sure that they are representative; nor do we know about the magnitude of relationships of concepts and/or categories.

Using multiple case studies is one possibility to find supporting or contradicting evidence for the generated hypotheses. This would enable a cross-case pattern search and underpin the validity of the results (Eisenhardt, 1989). Further interviews and document examination could be a basis for qualitative or (in large numbers) even quantitative cross-sectional analyses.

In the following, we will discuss key findings and limitations of the models and their possible interplay.

In our contingency model, we capture the influence of a number of context factors on the way of SI/SOA development. An extension of the list of factors, which form part of the general context, will be left to future research and can be achieved for instance through further case studies or expert interviews. Accordingly, more categories (and relating concepts) such as a management commitment (with the possible instances 'high', 'medium', 'low') might be discovered and it should be analyzed, if this influences the way of SI/SOA development e.g. by inducing top down approaches incorporating strategic aspects.

Moreover, based on only one case study, it is impossible to determine the extent to which a single factor contributes to certain outcomes. After conducting more case studies, a factor analysis could help to understand more reliably how the combination of identified factors affects the use of methods. Since there is already a large body of literature that provides a comprehensive list of context factors for domains different from SOA, future research should concentrate on identifying those that have a significant impact on SOA implementation projects and, in doing so, reason about the mechanisms more deeply. Quantitative research should aim at validating the resulting hypotheses and models.

The model of soft factor transition links experience with SOA design and implementation aspects (flexibility and reusability) and provides a description of a generative mechanism which is likely to be at work. One question for future research that arises from this model is if a technical (resp. business-oriented) understanding of services necessarily leads to fine-grained (resp. coarse-grained) services and a bottom up (resp. top down) approach. Most likely, more experience would enable the project team to flexibly adapt granularity, create services on different levels of granularity, or apply a top down (resp. a hybrid) approach. Further case studies would help to shed light on these cause-and-effect relationships.

Another intriguing question is whether one of these mindsets yields more success, and if this is the case, by which measure? According to most literature, business orientation is crucial for successful SOA implementations because only this focus ensures that services can support business processes. However, the ADAGE case shows that an almost pure bottom up approach and very fine-grained services stemming from a technical understanding still can result in a successful SOA implementation. Of course, the ADAGE architecture is far from being business process-oriented. Nonetheless, it can be considered a success. Since case studies can be used for generating hypotheses (like in this paper) and testing hypotheses (De Vries, 2005), a multiple case study could be useful to investigate correlations of service understanding and success in SOA projects.

Since relationships and interdependencies in our model fragments are designed based on the evidence retrieved from the ADAGE project, they might be incomplete. Literature e.g. argues that experience is one possible source to improve skills (see, e.g., Adelson and Soloway (1985), Dokko et al. (2009), Guile (2002)). Thus, further empirical evidence might support interdependencies found in literature and make an adjustment of our model necessary.

Our success model shows that multiple concepts more or less directly influence success of an SOA. This success can be viewed from different perspectives, i.e., technical or business-oriented. A model illustrating how these success factors contribute to the overall success of an SOA implementation could be a result of further empirical research.

Currently, the success measures in our model are not well operationalized. Future research should improve the conceptualization of success measures as well as their measurement. It is necessary to describe how to measure these factors and which units to use. Finally, an evaluation of the importance of a single concept (e.g., the direction of approach) and to what degree it influences the success of an SOA would be desirable.

Such quantification could be achieved by a factor analysis of context factors. Practitioners would be able to identify the setting of their specific project in advance and subsequently adjust or influence certain conditions (i.e., concepts) in order to improve the outcomes of their project. A framework giving advice on which measures could improve a project's results based on its setting is another objective for future research. A targeted application of methods based on identified project patterns (figure 6) could thus improve the probability of success for SOA projects.

The model fragments cannot be treated as being independent. On the one hand, there are evident relationships that can be traced in figure 2. Obviously, the notion of success which is operationalized by certain measures integrates the CM and the MSFT. On the other hand, there are most likely more links that have not been observed in our case. The concept of skills in the CM is probably related to what is called experience in the MSFT. Moreover, the way of SI/SOA development in the CM might be influenced by the direction of the identification approach, which is currently not part of the contingency model but of the MSFT. Hence, we assume that there are more causal links on several levels of our proposed model so that the latter will have to be adjusted after more elaborate work on this topic is performed.

Discussion, Limitations and Further Research

Our research endeavor presented in this paper can be characterized against the background of the conceptualizations of generalization we presented in section 2.2.

Regarding the types of generalization presented by Lee and Baskerville (2003), we used the two types concerned with generalizing from empirical statements because there are no appropriate theories in this field of research yet. Firstly, by formulating our observations in activity four, we generalized from data to description. Secondly, in activities five and seven, we generalized from description to theory. This second type was applied twice, namely in generating hypotheses and in developing model fragments.

The types of outcomes that are described by Walsham (1995) provided us with guidance to distinguish the results of our research. Concepts and categories on different levels of abstraction are the main elements of observations, hypotheses, and model fragments. These main elements are similar to the *concepts* that Walsham considers to be an outcome of qualitative research. However, he does not draw the distinction between different levels of abstraction. The *generation of theory* is reflected in several steps of our research process. Firstly, we developed hypotheses as an abstraction from our observations. Secondly, three models describing cause-and-effect relationships are presented as middle range theories.

By following Popay et al. (1998), our intention was to g*eneralize logically to a theoretical understanding*. In doing so, we omitted other approaches like statistical, naturalistic, analytical, and analogical generalization (see 2.3). The resulting limitations of this generalization approach are discussed in section 3.7. The generalization approach following "*generic social processes*" (Stoddart 2004) or socio-technical processes and the idea of *mechanism-based explanations* (causal mechanisms and causal generalization (see Hedström and Ylikoski 2010 and Woodward 2002)) fostered a more procedural perspective resulting in cause-and-effect relationships

In order to structure the research process, we adopted the sequence of activities from Eisenhardt (1989). Furthermore, we utilized her idea of “enfolding literature” to underpin the validity and to reach a higher conceptual level in a field where theories do not exist. This supported the logical generalization.

In addition, we extended this methodological basis stemming from the literature by the utilization of abstraction mechanisms from conceptual modeling. To the best of our knowledge, abstraction mechanisms have not been used in the realm of qualitative research yet. We believe that they provide a stable set of well described (and in some way formalized) ways to abstract from instances to types and, hence, to relate different levels of abstraction applying a clear semantic. Thus, they might reduce biased abstractions. In this respect, it could be argued that these mechanisms reduce creativity and limit abduction in the qualitative data analysis. However, we consider the guidance an advantage rather than a limitation. In all likelihood, these mechanisms have been used implicitly by researchers before, but an explicit description of the abstraction mechanisms intends to add rigor to the abstraction process to better justify the findings and to better support logical generalization and generalization by abstraction.

From our point of view, this backing in methodological work enhances reliability and traceability of our proceeding and of the findings in the case study analyses. Furthermore, it adds transparency to the basis upon which generalizations can be made and justified. On the other hand, it is useful in revealing limitations. It makes clear that usually generalization about populations is not possible. In addition, it reveals that the findings are untested theories and tendencies, “moderatum generalizations” (p.215) as Williams (2000) puts it.

The application of different approaches and conceptualizations enabled us to develop a basis upon which we can identify and justify the possibilities and limitations of generalization. Nevertheless, one can argue that this is rather an eclectic approach to enhance traceability and reliability. Hence, further research in the realm of qualitative methodology should be directed toward developing comprehensive and integrated frameworks, synthesizing different viewpoints of generalization. Sound frameworks may provide valuable guidance for endeavors in qualitative research.

As described previously, there are more abstraction mechanisms in the literature of conceptual modeling. We assume that they might be useful for qualitative research approaches as well, but a more detailed elaboration, e.g., the development of an appropriate framework for the use of abstraction mechanisms in qualitative research requires further intensive methodological work.

Conclusion

Since a common understanding of generalization in qualitative information systems research is missing, this paper offers an overview of existing approaches and conceptualizations. Different types and outcomes are described in table 1 which is referenced throughout the paper to classify the methods used and the results generated. We extended the notion of generalization by introducing abstraction mechanisms of conceptual modeling.

From a methodological perspective, we showed that abstraction mechanisms from the field of conceptual modeling can be successfully applied in conjunction with existing approaches in the realm of qualitative information systems research. In this respect, the paper represents an attempt to bring some clarity to the many different conceptualizations in the methodological discourse on generalizability and to discuss and demonstrate their applicability.

From another point of view, the paper contributes to the empirical research in SOA, especially in SOA development and service identification. Based on an explorative single case study, we developed hypotheses and model fragments that abstract from the subjective understanding of the project participants and observations made by the researchers. By referring to the aforementioned conceptualizations and mechanisms, we tried to make the research process transparent and traceable, so SOA researchers could appraise the reliability of the results.

Further research is needed and desirable regarding both perspectives. The model fragments and hypotheses described are based on case data of a single case study and should be complemented by more cases in order to extend the models. Further research should also aim at verifying and supporting the results applying qualitative and/or quantitative methods. The methodological discourse on generalizability in qualitative research should lead to comprehensive and integrated frameworks giving sound guidance to research projects in this area.

References

Adelson, B. & Soloway, E. (1985) "The Role of Domain Experience in Software Design", IEEE Transactions on Software Engineering, Vol 11 No 11, pp 1351-1360.

Analyti, A., Theodorakis, M., Spyratos, N. & Constantopoulos, P. (2007) "Contextualization as an independent abstraction mechanism for conceptual modeling", Information Systems, Vol 32 No 1, pp 24-60.

Anderson, D., Howell-Barber, H., Hill, J., Javed, N., Lawler, J. & Li, Z. (2005) "A Study of Web Services Projects in the Financial Services Industry", Information Systems Management, Vol 22 No 1, pp 66-76.

Artus, D. J. N. "SOA Realization: service design principles". [http://www.ibm.com/developerworks/webservices/library/ws-soa-design/], Accessed: 25-08-2010.

Babbie, E. (2010) The Practice of Social Research, Wadsworth, Belmont.

Baskerville, R., Cavallari, M., Hjort-Madsen, K., Pries-Heje, J., Sorrentino, M. & Virili, F. (2005) "Extensible Architectures: The Strategic Value of Service-oriented Architecture in Banking", European Conference on Information Systems, Regensburg, Germany.

Becker, A., Buxmann, P. & Widjaja, T. (2009) "Value Potential and Challenges of Service-Oriented Architectures - A User and Vendor Perspective", 17th European Conference on Information Systems (ECIS), Verona, Italy.

Becker, H. S. (1990) "Generalizing from Case Studies", in: E.W. Eisner and A. Pecking (Eds.), Qualitative Inquiry in Education: The Continuing Debate, Teacher's College Press, New York London, pp 233-242.

Benbasat, I., Goldstein, D. K. & Mead, M. (1987) "The case research strategy in studies of information systems", MIS Quarterly, No 11, pp 269-386.

Birkmeier, D., Glöckner, S. & Overhage, S. (2009) "A Survey of Service Identification Approaches", Enterprise Modelling and Information Systems Architectures, Vol 4 No 2, pp 20-36.

Börner, R., Goeken, M. & Rabhi, F. (2012) "SOA Development and Service Identification - A Case Study on Method Use, Context and Success Factors", Frankfurt School Working Paper Series, No. 189.

Börner, R., Goeken, M., Kohlborn, T. & Korthaus, A. (2011) "Context Factors for Situational Service Identification Methods", 6th International Conference on Internet and Web Applications and Services (ICIW), Xpert Publishing Services, St. Maarten, The Netherlands Antilles, pp 35-42.

Börner, R. & Goeken, M. (2009a) "Identification of Business Services - Literature Review and Lessons Learned", 15th Americas Conference on Information Systems (AMCIS), San Francisco, Calif., Paper 162.

Börner, R. & Goeken, M. (2009b) "Methods for Service Identification", in: Daniel Moldt, Juan C. Augusto and U. Ultes-Nitsche (Eds.), 7th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS), Milan, Italy, pp 76-84.

Boland, R. J., Newman, M. & Pentland, B. T. (2010) "Hermeneutical exegesis in information systems design and use", Information & Organization, Vol 20 No 1-20.

Bucher, T., Klesse, M., Kurpjuweit, S. & Winter, R. (2007) "Situational Method Engineering - On the Differentiation of "Context" and "Project Type"", in: Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.), Situational Method Engineering: Fundamentals and Experiences, Springer, Boston, pp 33-48.

Byrne, E. & Sahay, S. (2007) "Generelizations from an interpretive study: The case of a South African community-based health information system", South African Computer Journal, No 38, pp 8-19.

Darke, P., Shanks, G. & Broadbent, M. (1998) "Successfully completing case study research: combining rigour, relevance and pragmatism", Information Systems Journal, Vol 8 No 4, pp 273-289.

De Vries, E. J. (2005) "Epistemology and Methodology in Case Research: A Comparison between European and American IS Journals", 13th European Conference on Information Systems, Regensburg, Germany.

Dokko, G., Wilk, S. L. & Rothbard, N. P. (2009) "Unpacking Prior Experience: How Career History Affects Job Performance", Organization Science, Vol 20 No 1, pp 51-68.

Doolin, B. (1996) "Alternative Views of Case Research in Information Systems", Australasian Journal of Information Systems, Vol 3 No 2, pp 21-29.

Eisenhardt, K. M. (1989) "Building Theory from Case Study Research", The Academy of Management Review, Vol 14 No 4, pp 532-550.

Eisenhardt, K. M. & Graebner, M. E. (2007) "Theory building from cases Opportunities and challenges", Academy of Management Journal, Vol 50 No 1, pp 25-32.

Elfatatry, A. (2007) "Dealing with Change: Components versus Services", Communications of the ACM, Vol 50 No 8, pp 35-39.

Gadamer, H.-G. (1976) "The Historicity of Understanding", in: P. Connerton (Ed.), Critical Sociology, Selected Readings, Penguin Books, Harmondsworth, UK, pp 117-133.

Geertz, C. (1973) The Interpretation of Cultures, Basic Books, New York.

Goeken, M. (2006) Entwicklung von Data-Warehouse-Systemen. Anforderungsmanagement, Modellierung, Implementierung, Deutscher Universitäts-Verlag, Wiesbaden.

Gregor, S. (2006) "The nature of theory in information systems", MIS Quarterly, Vol 3 No 30, pp 611-642.

Guile, D. (2002) "Skill and Work Experience in the European Knowledge Economy", Journal of Education & Work, Vol 15 No 3, pp 251-276.

Hedström, P. & Ylikoski, P. (2010) "Causal Mechanisms in the Social Sciences", Annual Review of Sociology, Vol 36, pp 49-67.

Hovorka, D. S. & Lee, A. S. (2010) "Reframing interpretivism and positivism as understanding and explanation: Consequences for information systems research", 31st International Conference on Information Systems (ICIS), St. Louis.

Klein, H. H. & Myers, M. D. (1999) "A set of principles for conducting and evaluating interpretive field studies in information systems", MIS Quarterly, Vol 23 No 1, pp 67-94.

Kohlborn, T., Korthaus, A., Chan, T. & Rosemann, M. (2009a) "Identification and Analysis of Business and Software Services - A Consolidated Approach", IEEE Transactions on Services Computing, Vol 2 No 1, pp 50-64.

Kohlborn, T., Korthaus, A., Chan, T. & Rosemann, M. (2009b) "Service Analysis - A Critical Assessment of the State of the Art", 17th European Conference on Information Systems (ECIS), Verona, Italy.

Lee, A. (1991) "Integrating Positivist and Interpretive Approaches to Organizational Research", Organization Science, Vol 2 No 4, pp 342-365.

Lee, A. S. (1989) "A scientific methodology for MIS case studies", MIS Quarterly, Vol 13 No 1, pp 33-50.

Lee, A. S. & Baskerville, R. L. (2003) "Generalizing Generalizability in Information Systems Research", Information Systems Research, Vol 14 No 3, pp 221-243.

Lee, J. H., Shim, H.-J. & Kim, K. K. (2010) "Critical Success Factors in SOA Implementation: An Exploratory Study", Information Systems Management, Vol 27 No 2, pp 123-145.

Lincoln, Y. S. & Guba, E. G. (1985) Naturalist Inquiry, Sage Publications, Newbury Park London New Delhi.

Luthria, H. & Rabhi, F. A. (2009) "Building the Business Case for SOA: A Study of the Business Drivers for Technology Infrastructure Supporting Financial Service Institutions", in: Dennis Kundisch, Daniel J. Veit, Tim Weitzel and Christoph Weinhardt (Eds.), Enterprise Applications and Services in the Finance Industry, Springer, pp 94-107.

Mattos, N. M. (1989) "Abstraction Concepts: The Basis for Data Knowledge Modeling", in: Carlo Batini (Ed.), 7th International Conference on Entity-Relationship Approach: A Bridge to the User., North Holland, pp 473-492.

Merton, R. (1967). On sociological theories of the middle-range. In R. Merton (Ed.), *On Theoretical Sociology: Five essays old and new*. New York: Free Press.

Myers, M. D. (2009) Qualitative Research in Business & Management, Sage Publications, London.

Olivé, A. (2007) Conceptual Modeling of Information Systems, Springer, Berlin Heidelberg.

Parsons, J. & Wand, Y. (2008) "Using Cognitive Principles to Guide Classification in Information Systems Modeling", MIS Quarterly, Vol 32 No 4, pp 839-868.

Popay, J., Rogers, A. & Williams, G. (1998) "Rationale and standards for the systematic review of qualitative literature in health services research", Qualitative Health Research, Vol 8 No 3, pp 341-351.

Prus, R. C. (1994) "Generic Social Processes: Intersubjectivity and Transcontextuality in the Social Sciences", in: Mary Lorenz Dietz, Robert C. Prus and William Shaffir (Eds.), Doing Everyday Life: Ethnography as Human Lived Experience, Copp Clark Longman, Toronto, Ontario.

Schutz, A. (1954) "Concept and Theory Formation in the Social Sciences", The Journal of Philosophy, Vol 51 No 9, pp 257-273.

Smaling, A. (2003) "Inductive, Analogical, and Communicative Generalization", International Journal of Qualitative Methods, Vol 2 No 1, pp 52-67.

Stake, R. (1995) The art of case research, Sage Publications, Newbury Park.

Stebbins, R. A. (2001) Exploratory Research in the Social Sciences, Sage Publications, Thousand Oaks.

Stoddart, M. C. J. (2004) "Generalizability and qualitative research in a postmodern world", Graduate Journal of Social Science, Vol 1 No 2, pp 303-317.

Tsoukas, H. (2009) "Craving for Generality and Small-N Studies: A Wittgensteinian Approach towards the Epistemology of the Particular in Organization and Management Studies", in: David A. Buchanan and Alan Bryman (Eds.), The SAGE handbook of organizational research methods, SAGE, London et al., pp 285-301.

van Maanen, J. (1979) "The Fact of Fiction in Organizational Ethnography", Administrative Science Quarterly, Vol 24 No 4, pp 539-550.

Walsham, G. (1995) "Interpretive case studies in IS research. Nature and Method", European Journal of Information Systems, No 4, pp 74-81.

Walsham, G. (2006) "Doing interpretive research", European Journal of Information Systems, No 15, pp 320-330.

Weber, R. (2004) "The Rhetoric of Positivism versus Interpretivism: A personal View", MIS Quarterly, Vol 28 No 1, pp iii-xii.

Williams, M. (2000) "Interpretivism and Generalisation", Sociology, Vol 34 No 2, pp 209-224.

Woodward, J. (2002) "What is a mechanism? A counterfactual account", Philosophical Science, Vol 69, pp 366-377.

Yin, R. K. (2003) Case Study Research - Design and Methods, SAGE Publications, Thousand Oaks London New Delhi.

5.4 Artikel 9: Fragment Selection and Context Factors in Situational Methods for Service Identification

Veröffentlicht: Börner, R., Goeken, M., Kohlborn, T., Korthaus, A. (2011): Fragment Selection and Context Factors in Situational Methods for Service Identification, in: Proceedings of the 5th International IEEE Conference on Research Challenges in Information Science, 19.-21.05.2011, Guadeloupe, Frankreich, S. 335-342.

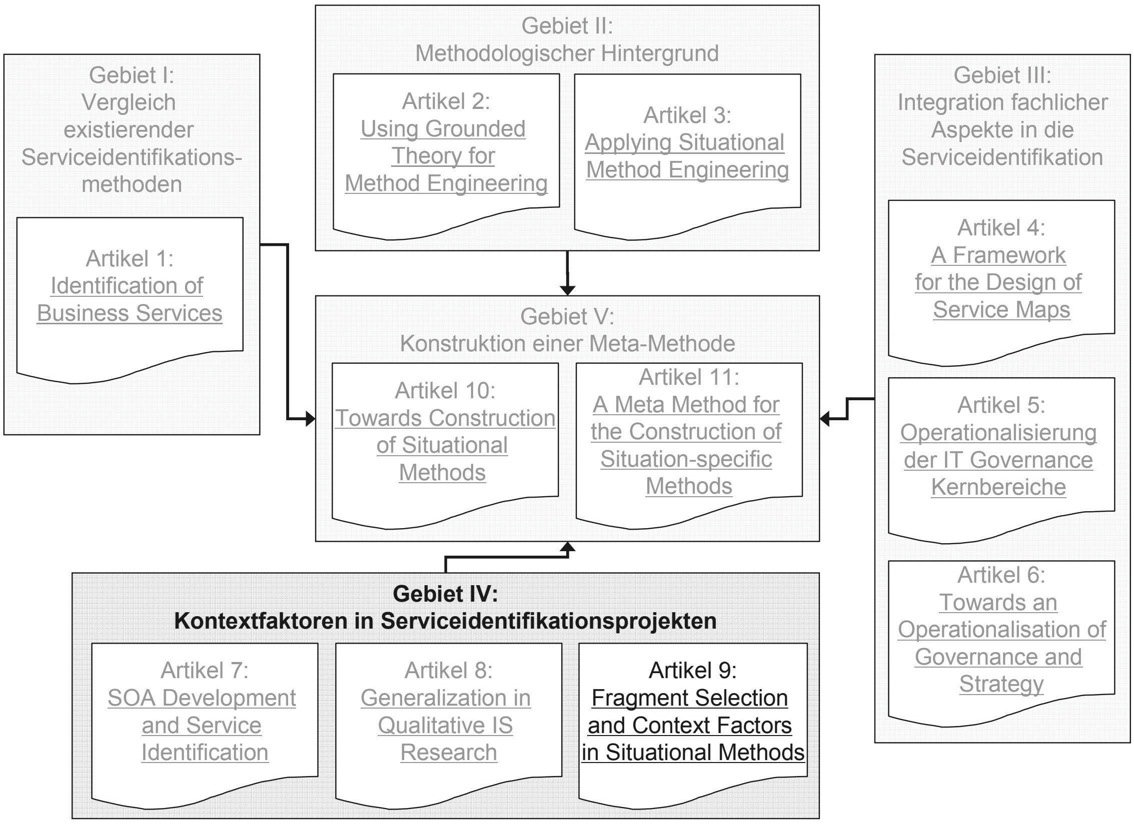

Abbildung 14: Artikel 9 im Kontext der kumulativen Dissertation

Fragment Selection and Context Factors in Situational Methods for Service Identification

René Börner, Matthias Goeken
Process Lab, IT Governance Practice Network
Frankfurt School of Finance & Management
Frankfurt/Main, Germany
[r.boerner;m.goeken]@fs.de

Thomas Kohlborn
Information Systems Discipline
Queensland University of Technology
Brisbane, Australia
t.kohlborn@qut.edu.au

Axel Korthaus
Faculty of Business and Law
Victoria University
Melbourne, Australia
axel.korthaus@vu.edu.au

Abstract

Situational method engineering is an approach to dynamically configure method fragments. In doing so, complete methods can be tailored to suit concrete project situations. Identifying such situations and selecting appropriate fragments is crucial for a successful method engineering process. In the field of service identification for service-oriented architectures most existing methods follow a one-size-fits-all approach that fails to acknowledge the broad variety of concrete circumstances that form the organizational context. In this paper, we argue that situational method engineering can be used to tailor service identification methods to particular application contexts. As a first step towards this goal, we analyze two explorative case studies and related literature to derive a basic set of relevant context factors. Moreover, the influence of these context factors on the selection and configuration of method fragments is examined. Adapting service identification methods to concrete project situations will improve their applicability and lead to a better service design.

Keywords

Keywords-Service-Oriented Architectures, Context Factors, Service Identification, Situational Method Engineering, Case Study.

I. INTRODUCTION

Service orientation is a highly recognized paradigm in enterprise architecture. The Organization for the Advancement of Structured Information Standards (OASIS) defines service-oriented architecture as a “paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains” [1]. Since the generic term “capabilities” can refer to both business functions and application functionalities, this definition supports a holistic SOA view that accommodates for two specific types of services: The term business service describes an autonomous, transformational capability that is offered to and consumed by external or internal customers for their benefit [2]. These services can have different levels of granularity ranging from comprehensive offerings (e.g. purchasing services) to fine-granular services (e.g. address verifications) [3]. The second type, software services, enables a close business and IT alignment in order to support business services and thus the agility of organizations [4]. Software services expose application functionalities that can be re-used and composed based on business needs. In order to implement SOAs successfully, an adequate identification of these services is essential.

For the last couple of years many authors have been looking at the identification of services (e.g. [5],[6]). Interestingly, most existing methods to identify services are based on a "one-size-fits-all" approach. Only few consider a configuration of methods depending on different circumstances such as the goals or various context factors of an SOA implementation. Even if context factors are considered, the scope of possible configurations is usually very limited [7]. Situational method engineering (SME) offers an opportunity to engineer service identification methods depending on situation-specific context factors of the project at hand. For this purpose, so-called method fragments are configured to methods that are adaptable to different situations.

The objective of this paper is to explore which context factors affect the selection of method fragments for service identification and how they influence the development of situational methods. A qualitative analysis approach was chosen to analyze data from two case studies inspired by the constant comparative analysis method of grounded theory [8] in order to identify relevant context factors for service identification methods. In section 2, the research method used in this paper is outlined. Section 3 describes the conducted case studies. The fourth section will identify relevant context factors and their influence on fragment selection, before section 5 concludes the paper with a summary, current limitations and an outlook for further research.

II. FOUNDATIONS

Following [9], we assume that „a richer understanding of a research topic will be gained by combining several methods together in a single piece of research or research program" (p. 241). In our research, we combined the two research methods case study research and design science research. The identification of context factors through case studies and the construction of a method supported by principles from SME can each be seen as research projects. Jointly, they are part of a more comprehensive research program (Fig. 1). The latter results in a comprehensive meta method for configuring a situational method for service identification applying design science practices and including the description of possible situations and available fragments.

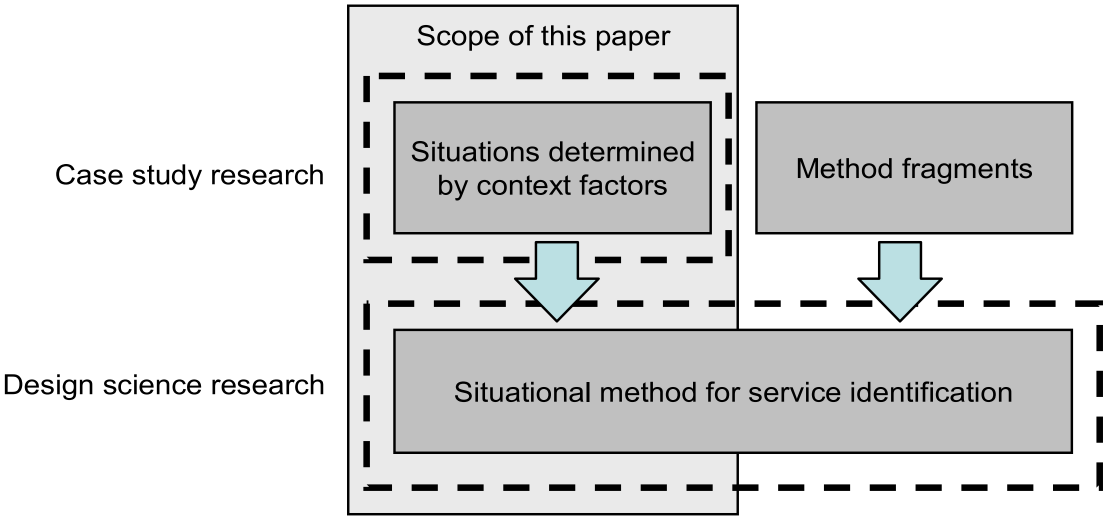

Figure 1. Scope and Research Methods

A. Case Study Research

Since the identification of relevant context factors is an explorative goal, we deliberately chose a case study approach. The latter is an appropriate method in this respect because a contemporary phenomenon is to be studied in its natural context [10], and we want to "provide descriptions of phenomena" [11]. Furthermore, case studies are particularly relevant for research in its "early, formative stages" [12, 13] which applies to the field of SOA [14, 15].

Basis for our research approach depicted in Fig. 2 is the case study data. In order to have a solid and profound basis to build upon, collected data had to be gathered from multiple sources of evidence. Using different sources serves to underpin the completeness and correctness of data. Inconsistencies can be made obvious through the use of various sources. This triangulation of data is important for the reliability of the case study's out comes [10]. Flexible and opportunistic data collection methods allow for reacting to emergent themes by adjusting data collection when necessary [16]. Besides others, project documents, presentations from project meetings and transcripts of interviews conducted with project participants are among our sources of evidence.

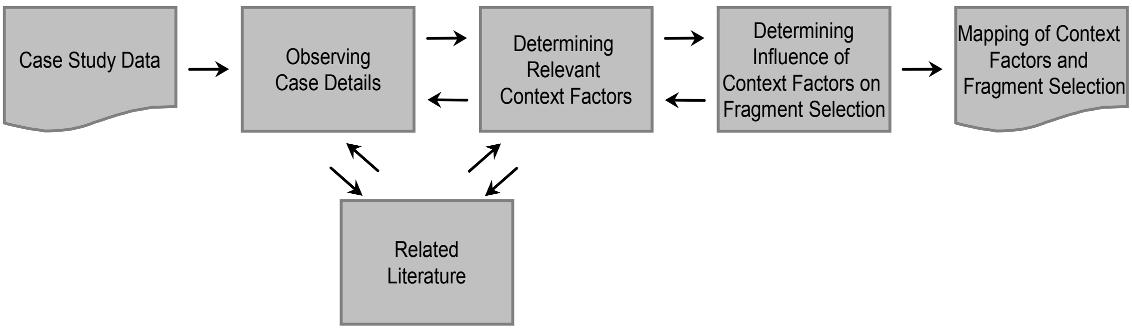

Figure 2. Research Approach

In an iterative approach we analyzed case details, determined relevant context factors based on this data and compared our findings with related literature (triangulation). The identification of concepts and patterns was conducted by employing techniques from grounded theory, for example, open and axial coding [17], and interpretative techniques [18]. Even though we did not use the coding techniques to their fullest extent, the general approach and respective tools supported the assignment of statements from the interviews and documents to concepts. The goal was to detect relevant particulars within the case data and to identify relevant concepts [16]. Examining the texts and interviews in their contexts – for example, by answering questions like 'who wrote the text, for what purpose, and what is the state of the project at the moment of writing' – allowed for deeper interpretations of the meaning of statements and comments [13].

In a next step, we used our observations to investigate the influence of context factors on the selection of method fragments. The case study data revealed that certain value parameters of the relevant context factors determine or rule out certain fragments. The final selection of method fragments naturally depends on their detailed description which is out the scope of this paper. Thus, an assignment of concrete fragments is not feasible herein but is a goal for further research. Result of our research process is therefore a mapping of context factors to their influence of fragment selection.

B. Situational Method Engineering in Service Identification

Service identification is one of the earliest activities in a service engineering process, which covers the whole lifecycle of a service. It is of particular importance, as any errors made during this activity can flow through to and build up in the design and implementation phases, which results in increased cost due to necessary rework [19]. A review of service analysis methods in general and service identification in particular by [6] reveals that none of the recently published methods is comprehensive and integrated enough to cover both SOA concepts (business *and* software services) to an adequate extent. However, as pointed out by the authors, different methods can complement each other and may have specific characteristics that make them more suitable in certain contexts.

For several years there have been efforts to guide the development of such methods in order to guarantee a high level of quality. To give this guidance is the task of *method engineering* (ME). ME is a discipline in information systems research meant to "*design, construct and adapt methods (…) for systems development*." [20]. Most approaches to ME identify activities, roles, results and techniques as important elements of methods [21].

Based on the assumption that a given method constructed at the time *t1* cannot fit all conceivable conditions and circumstances when it is used at a future time *t2*, the concept of *situational method engineering* (SME) was created. Indeed, it is quite improbable that a rigid method developed from theory is applicable in a concrete setting without modification [22]. A method should thus be configurable. To provide for this adaptability, so-called fragments are constructed and composed afterwards depending on the situation [23]. The method elements listed above are also integral parts of method fragments that will be used for SME. According to [24], a method fragment consists of precisely these elements. [25] define method fragments as "*standard building blocks based on a coherent part of a method. A situational method can be constructed by com-*

bining a number of method fragments" (p.360). For the purpose of this paper, the notion of method fragment will be defined as any reasonable combination of method elements representing a coherent part of a method.

The configuration of methods in SME is based on situations. Context factors of concrete projects are important characteristics that have to be considered in order to describe a situation at hand and subsequently configure a situational method [26]. Based on two case studies, this paper elaborates on relevant context factors for service identification projects and their influence on the configuration process.

III. CASE STUDIES

The following case studies describe two SOA implementation projects conducted in Australian companies, namely Suncorp and the Securities Industry Research Centre of Asia-Pacific (SIRCA). The explorative nature of the case studies was meant to discover relevant context factors for the identification of services. At the same time, the significantly diverse settings of both cases opened up a continuum of instantiations for these context factors [10]. At Suncorp, researchers from the Queensland University of Technology (QUT) conducted an action research study. They actively participated in the project and helped test and apply a service analysis & design (SAD) methodology developed by the researchers. We acknowledge that this may cause a researchers' bias. However, we found no evidence that the identified context factors and their effects on fragment selection would have differed if researchers had not participated in the project. In the second case study, several research methodologies were used. One of the most important sources of evidence have been interviews with SIRCA's employees and researchers from the University of New South Wales (UNSW), which were conducted shortly after the project had been completed. The interviews have been transcribed and analyzed afterwards.

A. Suncorp

In the context of an Australian ARC Linkage project titled "Service Ecosystems Management for Collaborative Process Improvement" (ARC Linkage Grant: LP0669244), some of the authors have developed a comprehensive SAD methodology for both business and software services [27]. Parts of this methodology related to service identification were applied in practice at Suncorp and the experiences gained through this exercise will build the foundation for the following discussion. Suncorp is a diversified company in the financial services sector. As one of Australia's leaders in banking, insurance, investment and superannuation focusing on retail customers and small to medium businesses, the Suncorp Group is Australia's sixth largest bank and third largest insurer. The Suncorp case study can be subdivided into three cases as the organization went through different change programs related to their take on service orientation.

The first case (1) focuses on service identification for integrating different systems. Suncorp had started a Claims Business Model Program some time ago with the intent to identify process improvements that would result in reduced leakage, reduced payments of ineligible claims, and lower handling costs. Suncorp's current systems were not flexible enough to support the required changes [28]. It was then decided that the new claims process should be implemented in a new insurance claims management system from Guidewire Software, the ClaimCenter application [29]. While the initial implementation was in support of personal home claims, implementation projects for claims in worker's compensation, personal motor, commercial property and others followed. In the ClaimCenter project, integration with a large number of external systems was required and thirteen development teams (including the external vendor and an offshore team) using different development methods had to be coordinated. The project team decided to use an SOA approach to integrate diverse systems such as policy, payments, receipting and claims. A standardization of interfaces was ought to improve reusability [28].

Against this background, the QUT project team came in and presented the consolidated SAD methodology to a solution architecture team from Suncorp's Business Technology group, which started the second case study (2). Suncorp had found that their approach to SOA and service analysis and design was rather ad-hoc and very much driven by bottom-up integration requirements of their pilot projects, potentially lacking strong alignment of the service designs with the business processes:

"The current process that we follow tends to be driven by the functional requirements and data requirements of the consumer. This results in a very entity-driven service[1] in which the consumer of the service needs to understand a lot more about the state and context of the call that they are making." (Suncorp Solution Architect)

A study protocol specified the objectives and the scope of the collaboration as well as the timeframe and the planned deliverables. In a first step, the action research study primarily focused on the identification of software services. The "motor claims" business process was chosen as input for the software service preparation and identification steps of the SAD methodology developed by QUT. To keep the scope manageable, two sub-processes, namely "claims intake" and "assessment" were selected. The researchers were provided with Suncorp's "motor claims" process models on different levels of hierarchy and additional business artefacts including the SOA Roadmap, the Insurance Domain Model and the ClaimCenter Hub System Architecture Specification. Based on this input, the researchers used the service preparation and identification steps prescribed by the SAD methodology for the two sub-processes and produced two reports that included the resulting service designs.

The third case study (3) with Suncorp extended the work that has been done previously, by applying the complete SAD methodology [27]. In particular, as part of a collaboration project between industry and university, three industry students applied the SAD methodology to derive business and software services starting from Suncorp's business strategy and capabilities. The project was driven by the desire to identify software services that not only support processes but also represent constituent elements of business

[1] An entity-driven service is built around business objects, such as customer, order, claim, etc. Thus, such a service typically offers context-agnostic operations that have a high probability to be used in different scenarios. These operations can comprise "CRUD" functionalities to manage the lifecycle of the specific entity.

services, which in turn needed to be identified first. Consulting QUT researchers along the project, the three students were able to apply the prescribed methodology and present their results to the business and IT audiences within Suncorp who largely benefited from the lessons learned of this exercise. A detailed report was provided at the end of the project.

B. SIRCA

In the context of the project "Ad-hoc DAta Grids Environments" (ADAGE), researchers from the UNSW implemented a service-oriented architecture for SIRCA. The project aimed at providing researchers an easier retrieval and analysis of heterogeneous data from different sources (grid environment) spontaneously in an unforeseeable fashion (ad-hoc). Neither business processes nor SOA were the core focus of the project. The former were hardly considered at all whereas the latter was chosen as the preferred architectural paradigm of this project. However, services (and their identification) were used as a means to meet SIRCA's requirements rather than being the subject of analysis themselves.

SIRCA provides a huge data repository containing historical financial market data such as news and trading data. Their aim is to supply this data to researchers especially at Australian and New Zealand universities. Thus, their business model is fairly simple and is covered by one business process only.

"Our processes are fairly (…) atomistic. In that way, we are very simple outfit, we are a data repository, we collect lots of data, we do fairly substandard processing to it to normalize it and make it easily accessible. And then people access the data with some fairly straight-forward enterprise in that regard." (Representative of SIRCA)

In ADAGE, services were created based on the available data. This implied a technical understanding of services, which is e.g. reflected by the synonymous use of the terms "service" and "web service" by project team members. Hence, the scope of service identification in this project was limited to software services. SIRCA employees were not thinking in terms of business processes, so that no model that could have been analyzed in the course of service identification was delivered. SIRCA's management however had some requirements in mind that should be fulfilled by services. Unfortunately, these were not documented, which makes traceability difficult. Requirements were communicated to the project team in scheduled weekly meetings and workshops. Service candidates were identified on the basis of these meetings and prototyped. In an iterative and incremental approach the functionality of these candidates was adjusted to finally meet SIRCA's requirements. A close collaboration between SIRCA's research and development department and UNSW's project team was a key to ensure the successful identification of services.

SIRCA's management did not aim at the implementation of an SOA in particular. The idea of services was basically advocated by UNSW's project team. Thus, there was no know-how on SIRCA's side as far as SOAs are concerned. On the outset of the project, a funding for three years was provided. At the end of the project in December 2009, funds for further six months were provided to implement the prototype and make it accessible to SIRCA's customers.

First and foremost, the search for services was driven by the idea to retrieve and integrate data from different sources. In a second step, project members came up with ideas which services could support researchers in analyzing data. This included, for example, building time series of financial data, merge data from different sources and visualize events. Only after that the identified and implemented services should be offered to third parties to support their business processes. Clearly, this was a requirements-driven bottom-up approach. Goals included the provision of a graphical user interface (GUI) to customers enabling them to directly invoke services in an ad-hoc fashion to analyze financial market data. This implies a distinct degree of customer interaction which influences the identification of services significantly. Certainly, SIRCA's case is not a typical example of service identification projects. Because of its rather extreme character it helps to identify possible instances of situational factors that usually cannot be found in typical cases.

IV. CONTEXT FACTORS IN SERVICE IDENTIFICATION

In order to engineer situational methods, relevant context factors that determine different categories of situations have to be identified. Hence, in this section we build on the case studies described previously to identify these context factors. The presentation of each context factor is structured as follows. Firstly, observations from the case studies are the basis for identifying context factors. Secondly, findings in related literature are briefly discussed where applicable to support the relevance of the encountered factors. Thirdly, an analysis of how these context factors influence the selection of fragments is conducted. Table 1 summarizes the results.

In SIRCA's case, all services were clearly meant to be exposed to researchers from associated universities, i.e. to external ***service consumers***. A graphical user interface (GUI) provides the opportunity for users to combine these services. Thus, users can choose themselves which services they need to use in order to analyze their data. In Suncorp's first case services were identified to integrate system functionalities. Hence, service consumers were purely internal. Functionalities that had to be accessed by using different applications were wrapped and can now be invoked as services. Conversely, the second and third case aimed at both internal and external service consumers as certain services were intended to be used by end-consumers, internal systems and/or internal entities such as departments within Suncorp.

Context Factor	Parameter Value	Influence on Fragment Selection
Service consumer	- internal - external - both	- inter-organizational service maps are not applicable for purely internal services - "line of visibility" particularly important for external consumers - Data flow analysis combined with legal check for externally provided services
Budget	- restrictive budget - generous funding	- fragments supporting strategic aspects might not be used due to a restrictive budget - a generous funding enables comprehensive use of fragments - restrictive budgets often result in the selection of IT-oriented fragments that demand less resources
SOA concepts	- business services - software services - hierarchy of services	- BPM techniques and respective fragments are essential for a business service-oriented identification - technical fragments such as asset analysis might be sufficient for the identification of software services - a comprehensive, hybrid approach using fragments with both top down and bottom up techniques is necessary for a hierarchy of services
SOA maturity level	- SIMM level 1-3 - SIMM level 4-7	- fragments focused on SOA governance have only little meaning for maturity levels 1-3 - for SIMM levels 4-7 e.g. inter-organizational service maps are important
Compliance	- general laws - industry-specific regulations - internal policies	- all organizations have to ensure compliance with general laws such as consumer data privacy and must use respective fragments when service consumers are outside the company - a fragment that analyzes the necessity of industry-specific approaches should always be used - depending on the industry, additional fragments are needed - if there are internal policies, fragments that e.g. ensure consistent naming of services have to be applied
IT department	- existent - not existent	- inputs like architectural concepts are only available from IT departments and restrict the use of methods - fragments are not applicable if they demand roles such as IT administrator
Interaction	- customer interaction - employee interaction	- for the interaction with employees, roles prescribed by fragments have to be available - fragments delivering swim lane diagrams and analyzing the "line of visibility" are particularly useful if customer interaction is pivotal

TABLE I. Context Factors and Their Influence on Fragment Selection

As shown in the case studies, services can be provided for different service consumers, e.g. other divisions (internally), third parties (externally) or both. Services can be used to integrate heterogeneous enterprise applications [30, 31] and simplify the access to certain functionalities for staff, i.e. for internal customers only. If this is known a priori, a number of activities and results such as the creation of an inter-organizational service map are not applicable in this situation. Moreover, there might be legal constraints that only apply if services are offered to third parties. Passing on customer data for instance must be permitted by the customer in some countries. Thus, fragments that demand an analysis of respective laws and regulations are only necessary where such data is passed on to third parties. An analysis of consumer interaction can be important for internal and external service provision. A fragment analyzing the "line of visibility" is much more important if services are exposed to external customers [32] and would add lots of value in cases like SIRCA's. However, they are less relevant in a context as given in Suncorp's first case.

In both projects the ***budget*** seemed to play an important role and we perveived that budgeting has a significant impact when it comes to choosing necessary steps and general proceeding of a method for service identification. Generally, a setting as encountered in SIRCA's case with a budget that allows for an extensive time frame of three years provides the opportunity for a thorough and systematic application of identification methods. You would expect the utilization of many techniques in order to ensure a high quality of implemented services. A detailed analysis of all available strategic and technical documents would be typical in such circumstances. However, in SIRCA's case the absence of such documents naturally dominated the generous time scope and made the application of many techniques impossible. Again, the head of research and development certainly devoted enough resources in SIRCA's case but still failed this broad analysis. The following citation is an excerpt of an interview with a UNSW researcher conducted for the analysis of the ADAGE project.

"We didn't do the asset identification, because we didn't have anyone who knew about that, but if we had then, we would have done it. So you are constrained by the cost it takes to develop, but also you were constrained by people's skills." (Researcher at the UNSW)

While the first project was funded internally, the second and third cases at Suncorp were basically developed in a collaborative environment between Suncorp and QUT based on mutual in-kind contributions. As such, the cases could be treated as pilots and were not associated with any costs other than research and development project budget for Suncorp. Budget information about the first case is confidential and cannot be reported here. However, due to the partly academic character of these projects, funding restrictions were not a problem. At Suncorp, this availability of resources was used to apply QUT's identification method diligently.

Literature broadly confirms that the budget has implications on the number of available staff, the time pressure and the possibility to incorporate external help from consultants [33].The higher the project sponsor's position in the company's hierarchy, the more likely is a generous funding. This allows for a proper analysis especially of strategic aspects and the inclusion of business processes and a comprehensive use of fragments. An initiation of a service identification project by the middle management (which is commonly accompanied by smaller budgets) often results in more pragmatic or technically-driven SOA implementations. Fragments dealing with Business Process Management as well as business process driven approaches (top down) are likely to be omitted in such cases. Instead, technically-oriented fragments analyzing applications and IT functionalities (bottom up) are used since they often promise quicker results. To make up for limited employee skills, a larger budget is necessary because the company has to rely on external support. If funding is limited, certain fragments cannot be used due to this lack of skills.

The case studies showed clearly that different understandings of services based on different ***SOA concepts*** influence the proceeding of service identification significantly. In Suncorp's first case aiming at the integration of different systems, the underlying technical SOA understanding implied a focus that lay purely on developing reusable software services. As part of the second case – with a Business Process Management perspective in mind – software services that support business processes or at least subprocesses were the goal of the analysis. The third case at Suncorp focused on identifying software services that support business services. Rarely, the focus on software services is as clear as in SIRCA's case.

A system integration approach like in Suncorp's first case concentrated on software services, so that method fragments reflecting a more business-related SOA understanding and targeting the identification of business services were not considered. In the second case, service identification had to include activities related to the analysis of process models and involved such fragments with not only IT-related, but also business-related staff roles. In Suncorp's third project both SOA concepts have been addressed. If the service identification is limited to a rather technical point of view like in SIRCA's case, the set of method fragments to be considered will be limited. Frequently, a hierarchy of services is the outcome of an identification process [34]. Higher-level services that support business processes compose finer-grained software services that adhere to the technical preconditions of underlying systems and data. In order to achieve this more complex type of an SOA, a broader range of method fragments has to be used complementarily. This is reflected in many hybrid (or "meet-in-the-middle") approaches that can be found in related literature [5, 35].

We observed different degrees of previous experience in the examined cases as far as SOA and service orientation is concerned. Based on this observation, we concluded that the ***SOA maturity level*** a company has achieved plays a role in the configuration of methods for service identification. SIRCA did not have any services at all when the project was initiated. At Suncorp, SOA maturity can be considered as relatively low. First projects had been conducted in the area of implementing software services. However, these projects did not aim at understanding business requirements and their impact on service-orientation but rather understanding the integration requirements of existing applications. It can be conjectured that the maturity related to the adoption of service-oriented concepts will increase over time, which will change the scope of the SOA understanding and consequently the way service identification has to be conducted, as can be seen in the two latter Suncorp cases.

The SOA maturity level can be distinguished following the Service Integration Maturity Model (SIMM) by [36]. Levels 1-3 describe a rather low service orientation whereas companies with a SIMM level of 4-7 are more advanced in the field of SOA. If the latter is the case, an analysis of service maps among different divisions of a company can be essential to build an enterprise-wide SOA [37]. A fragment providing such an analysis cannot be sensibly used if there is no service orientation and subsequently (almost) no services. The same is true for all fragments dealing with SOA governance and inter-organizational aspects of SOA.

Compliance issues can arise from both legal obligations and regulatory restrictions as well as from internal policies and may require corresponding service identification method fragments that address these issues. In SIRCA's case no particular regulatory or legal requirements had to be considered. However, the Reuters market data provided by SIRCA must not be used by everyone. It might only be used for academic purposes. The academic institution has to pay a subscription fee to SIRCA to give their employees access to the data. Hence, restricted access to data and the intended use of services must be considered when services are identified. As far as Suncorp is concerned, for example the general insurance reform act and related laws and regulations issued by the Treasury Department of the Australian Government are industry-specific requirements. Since implemented services in Suncorp's case are exposed to customers, confidential treatment of sensitive customer data had to be guaranteed by a proper service design. Furthermore, as already indicated, Suncorp generally follows an agile approach to developing services. Thus, methods related to the identification, design and implementation of services have to comply with the agile paradigm.

In case services are provided to third parties, a fragment that guarantees customer data privacy and security has to be applied. Therefore, interactions with and data flows towards all service consumers have to be analyzed. Certain industries such as banking, insurance or pharmaceuticals have to adhere to additional, stricter regulations and should use respective fragments. Internal policies – such as restrictions on software development methods – can make some fragments inapplicable. Other fragments may be necessary to fulfil for instance internal naming conventions.

In SIRCA, methods had to be adapted to account for the lack of an ***IT department***. Suncorp is structured along the lines of business but has an IT division that takes care of a company-wide IT architecture and infrastructure. In small companies, responsibility for IT is commonly distributed all over the organization and departments tend to implement isolated IT solutions or so-called silos. Larger organizations usually have a designated IT unit. Hence, the existence of an IT department and thus the degree of centralization of the IT infrastructure is an important context variable.

On the one hand, a high degree of centralization or the existence of a central division supervising and governing IT implementation throughout a company usually leads to more transparency. Frequently, at least some information on applications and data is readily available. This can be used as input for service identification method fragments. On the other hand, some fragments demand certain roles such IT administrators or newly composed units consisting of business and IT employees [38]. In a small company that lacks an IT department, these method fragments are often not applicable. The company size (frequently considered to have an important influence on SOA implementations [39]) and the geographic scope of operations are thus closely linked to the existence of a central IT department and subsequently not considered as context factors themselves.

Due to the fact that in all cases participation, exchange and contribution of service consumers differed notably, we hypothesized that varying degrees and forms of ***interaction*** with both customers and employees necessitate the use of different method fragments. In SIRCA's case, for instance, employees are not directly involved in service delivery because the services are very fine-grained and fully automated. The coarser-grained services are, the greater is the possibility that they are only semi-automated or manual and subsequently interact with employees. Customer interaction is of high importance in SIRCA's case because the ad-hoc composition of services is a primary goal. However, due to their fine-grained nature, services themselves are executed independently from users, i.e. no customer interferes directly in a service. At Suncorp, employees were only involved to showcase the developed methodology and gain information about current practices at Suncorp. As part of the third project, different employees at Suncorp were involved to identify business services.

Getting access to business roles was difficult in Suncorp's case but required by the used method fragments. Where this is impossible, a different method configuration is necessary. In general, a customer interaction can be obligatory in some places or can happen "on demand" if required by the service or desired by the customer [40]. In automated services, possible customer interaction has to be foreseen and planned for. If customer interaction is a major issue for the identification of services, suitable method fragments are crucial for a successful implementation. Results such as swim lane diagrams that show interfaces to customers are one example.

All context factors described above were found in the two case studies. Their effect on the selection of method fragments for service identification is summarized in table I. Particularly the case study at SIRCA revealed some more potential context factors. Since their relevance could not be observed in the cases at Suncorp, these factors were omitted from the discussion in this paper. Probably, there are interdependencies and relationships between the identified context factors. Analyzing these relationships and identifying relevant combinations that determine the situational configuration of methods is out of the scope of this paper and will be the focus of future work.

V. CONCLUSION

Not only literature but also experience shows that methods in information systems research should be configurable depending on the situation at hand, i.e. in a situation-specific way. This also holds true for service identification methods in service-oriented architectures. In order to support a situation-specific configuration of such methods, situations have to be defined by situational factors. The latter determine which so-called method fragments should be used in the course of an identification process.

Based on qualitative research using grounded theory, the data of two case studies was analyzed in this paper to identify seven context factors, compare them with existing literature and describe how actual instances of these factors can influence the selection of method fragments. Due to the explorative nature of the two case studies, there is no guarantee that the derived list of factors is comprehensive. Further case studies might reveal other important factors. Moreover, the relevance of the identified factors cannot be proven by our case studies. Only the application of a complete situational method to a service identification project could attest their relevance.

As a prerequisite for situational methods, method fragments that can be combined have to be created. Thus, either parts of existing methods have to be identified as feasible method fragments or new fragments have to be created [41]. Therefore, it is necessary to investigate which activities, techniques, roles, results and sequences can be combined reasonably. Designing these fragments to meet the requirements of situation-specific service identification is part of ongoing research [42]. The so designed method fragments have to be configured to suit concrete situations that can be characterized by the context factors identified in this paper. First hints at the influence of context factors on fragment selection are given herein. How exactly context factors must be weighed or which context factors may dominate others must be elaborated in more detail.

Finally, a comprehensive situational method for service identification needs to be developed. It should define how to configure existing method fragments depending on the situation at hand. In order to evaluate the quality of a so configured method, more empirical research should be carried out. In contrast to the two case studies used to derive context factors exploratively, further case studies or action research should apply a newly created situational method to prove its applicability and evaluate its concept.

REFERENCES

[1] OASIS, "Reference Model for Service Oriented Architecture 1.0," http://docs.oasis-open.org/soarm/v1.0/soa-rm.pdf

[2] M. Rosemann, E. Fielt, T. Kohlborn and A. Korthaus, "Business Service Management," in Smart Services CRC White Paper no 1, 2009.

[3] A. Sehmi and B. Schwegler, "Service-oriented modeling for connected systems - part 1," The Architecture Journal, vol. 7, pp. 33-41, 2006.

[4] L. Cherbakov, G. Galambos, R. Harishankar, S. Kalyana and G. Rackham, "Impact of service orientation at the business level," IBM Systems Journal, vol. 44, pp. 653-668, 2005.

[5] R. Börner and M. Goeken, "Methods for Service Identification," in 7th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS), pp. 76-84. Milan, Italy, 2009.

[6] T. Kohlborn, A. Korthaus, T. Chan and M. Rosemann, "Service Analysis - A Critical Assessment of the State of the Art," in 17th European Conference on Information Systems (ECIS), Verona, Italy, 2009.

[7] R. Börner, "Applying Situational Method Engineering to the Development of Service Identification Methods," in 16th Americas Conference on Information Systems (AMCIS), Paper 18. Lima, Peru, 2010.

[8] B. G. Glaser and A. L. Strauss, "The Discovery of Grounded Theory: Strategies for Qualitative Research. Chicago: Aldine Publishing Company, 1967.

[9] J. Mingers, "Combining IS Research Methods: Towards a Pluralist Methodology," Information Systems Research, vol. 12, pp. 240-259, 2001.

[10] R. K. Yin, "Case Study Research - Design and Methods," 3 ed. vol. 5. Thousand Oaks London New Delhi: SAGE Publications, 2003.

[11] P. Darke, G. Shanks and M. Broadbent, "Successfully completing case study research: combining rigour, relevance and pragmatism," Information Systems Journal, vol. 8, pp. 273-289, 1998.

[12] I. Benbasat, D. K. Goldstein and M. Mead, "The case research strategy in studies of information systems," MIS Quarterly, pp. 269-386, 1987.

[13] M. D. Myers, "Qualitative Research in Business & Management. London: Sage Publications, 2009.

[14] H. Luthria and F. A. Rabhi, "Building the Business Case for SOA: A Study of the Business Drivers for Technology Infrastructure Supporting Financial Service Institutions," in Kundisch, D., et al. (eds.) Enterprise Applications and Services in the Finance Industry, pp. 94-107. Springer, 2009.

[15] R. A. Stebbins, "Exploratory Research in the Social Sciences. Thousand Oaks: Sage Publications, 2001.

[16] K. M. Eisenhardt, "Building Theory from Case Study Research," The Academy of Management Review, vol. 14, pp. 532-550, 1989.

[17] A. Strauss and J. M. Corbin, "Basics of qualitative research: Grounded theory procedures and techniques. Thousand Oaks, CA: Sage Publications, 1990.

[18] G. Walsham, "Interpretive case studies in IS research. Nature and Method," European Journal of Information Systems, vol., pp. 74-81, 1995.

[19] S. Inaganti and G. K. Behara, "Service Identification - BPM and SOA Handshake," BPTrends, vol. 3, pp. 1-12, 2007.

[20] J. Ralyté, S. Brinkkemper and B. Henderson-Sellers, "Situational Method Engineering: Fundamentals and Experiences," in Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences, pp. V-VI. Boston: Springer, 2007.

[21] M. Goeken, "Entwicklung von Data-Warehouse-Systemen. Anforderungsmanagement, Modellierung, Implementierung. Wiesbaden: Deutscher Universitäts-Verlag, 2006.

[22] M. N. Aydin, "Examining Key Notions for Method Adaptation," in Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences, pp. 49-63. Boston: Springer, 2007.

[23] J. Ralyté and C. Rolland, "An Approach for Method Engineering," in 20th International Conference on Conceptual Modelling, pp. 471-484. Springer, Yokohama, Japan, 2001.

[24] M. Cossentino, S. Gaglio, B. Henderson-Sellers and V. Seidita, "A metamodelling-based approach for method fragment comparison," in 11th International Workshop on Exploring Modeling Methods in Systems Analysis and Design at CAiSE, Luxemburg, 2006.

[25] P. J. Agerfalk, S. Brinkkemper, C. Gonzalez-Perez, B. Henderson-Sellers, F. Karlsson, S. Kelly and J. Ralyté, "Modularization Constructs in Method Engineering: Towards Common Ground?," in Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences, pp. 359-368. Boston: Springer, 2007.

[26] T. Bucher, M. Klesse, S. Kurpjuweit and R. Winter, "Situational Method Engineering - On the Differentiation of "Context" and "Project Type"," in Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences, pp. 33-48. Boston: Springer, 2007.

[27] T. Kohlborn, A. Korthaus, T. Chan and M. Rosemann, "Identification and Analysis of Business and Software Services - A Consolidated Approach," IEEE Transactions on Services Computing, vol. 2, pp. 50-64, 2009.

[28] J. A. Couzens, "Implementing an Enterprise System at Suncorp Using Agile Development," in 20th Australian Software Engineering Conference, Gold Coast, Australia, 2009.

[29] Guidewire, "Guidewire Claimcenter - the Flexible Claims Management System for Property & Casualties Insurers," http://www.guidewire.com/our_solutions/claimcenter

[30] T. Erl, "Service-Oriented Architecture - A Field Guide to Integrating XML and Web Services. Upper Saddle River, NJ: Prentice Hall, 2004.

[31] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy and K. Holley, "SOMA: A method for developing service-oriented solutions," IBM Systems Journal, vol. 47, pp. 377-396, 2008.

[32] K. Klose, R. Knackstedt and D. Beverungen, "Identification of Services - A Stakeholder-Based Approach to SOA Development and Its Application in the Area of Production Planning," in 15th European Conference on Information Systems (ECIS), pp. 1802-1814. St. Gallen, Switzerland, 2007.

[33] J. Becker, R. Knackstedt, D. Pfeiffer and C. Janiesch, "Configurative Method Engineering," in 13th Americas Conference on Information Systems (AMCIS), Paper 56. Keystone, Colo., 2007.

[34] N. Josuttis, "SOA in der Praxis - System-Design für verteilte Geschäftsprozesse," 1. ed. Heidelberg: dpunkt.verlag, 2008.

[35] D. Birkmeier, S. Glöckner and S. Overhage, "A Survey of Service Identification Approaches," Enterprise Modelling and Information Systems Architectures, vol. 4, pp. 20-36, 2009.

[36] A. Arsanjani and K. Holley, "The Service Integration Maturity Model: Achieving Flexibility in the Transformation to SOA," in IEEE International Conference on Services Computing, pp. 515. Chicago, IL, 2006.

[37] F. Kohlmann and R. Alt, "Aligning Service Maps - A Methodological Approach From the Financial Industry," in 42nd Hawaii International Conference on System Sciences (HICSS), pp. 1-10. Hawaii, 2009.

[38] R. Börner, S. Looso and M. Goeken, "Towards an Operationalisation of Governance and Strategy for Service Identification and Design," in 13th IEEE International EDOC Conference, pp. 180-188. Auckland, New Zealand, 2009.

[39] D. Sedera, "Does Size Matter? Enterprise System Performance in Small, Medium and Large Organizations," in 2nd Workshop on 3rd Generation Enterprise Resource Planning Systems, pp., Copenhagen, Denmark, 2008.

[40] M. Leyer and J. Moormann, "Facilitating operational control of business services: A method for analysing and structuring customer integration," in 21st Australasian Conference on Information Systems (ACIS), Paper 42. Brisbane, Australia, 2010.

[41] B. Henderson-Sellers and J. Ralyté, "Situational Method Engineering: State-of-the-Art Review," Journal of Universal Computer Science, vol. 16, pp. 424-478, 2010.

[42] R. Börner, "Towards Construction of Situational Methods for Service Identification," in Ralyté, J., Mirbel, I., Deneckère, R. (eds.) ME 2011 - IFIP International Federation for Information Processing, pp. 204-218. Paris: Springer, 2011.

5.5 Zwischenfazit

Die erste Fallstudie betrachtete ein SOA-Implementierungsprojekt, das Wissenschaftler der University of New South Wales in einem kleinen australischen Unternehmen, welches auf die Bereitstellung von Börsendaten spezialisiert ist, durchgeführt haben. Auf Basis vollständig transkribierter Interviews sowie Projektdokumentationen wurden mit Techniken der Grounded Theory Konzepte identifiziert und in einem Modell festgehalten. Dazu zählen auch so genannte Kontextfaktoren, die die Vorgehensweise in Softwareprojekten entscheidend beeinflussen.

Ergänzend dazu wurden in einer Action Research-Studie bei einem der größten australischen Versicherungsunternehmen Kontextfaktoren identifiziert. Anschließend wurden diese sowohl mit den Ergebnissen der ersten Studie als auch mit solchen in der Literatur vorhandenen Kontextfaktoren verglichen. Das Resultat ist eine Liste mit für die Serviceidentifikation relevanten Kontextfaktoren, ihren Ausprägungen sowie den Einfluss auf die Auswahl von Methodenfragmenten.

6. Gebiet V: Konstruktion einer Meta-Methode

6.1 Einleitung

Dieses letzte Gebiet konsolidiert die Ergebnisse aller vorangegangenen Artikel und entwickelt daraus die Meta-Methode. Der zehnte Artikel geht insbesondere auf die Methodenkonstruktion vor dem Hintergrund aktueller Erkenntnisse des Situational Method Engineering ein. Dazu werden alle notwendigen Bestandteile (Situationen, Fragmente, etc.) beleuchtet und mit den domänenspezifischen Inhalten der Serviceidentifikation verknüpft.

In Artikel 11 werden Vorgehensweisen zur Konfiguration der Meta-Methode detailliert betrachtet. Des Weiteren steht die Entwicklung von Methodenfragmenten zur Serviceidentifikation im Mittelpunkt. Schließlich wird mit der Anwendung der Meta-Methode auf einen konkreten Fall demonstriert, dass sie in der Praxis umsetzbar ist und wie eine situationsspezifische Methode konstruiert werden kann.

6.2 Artikel 10: Towards Construction of Situational Methods for Service Identification

Veröffentlicht: Börner, R. (2011): Towards Construction of Situational Methods for Service Identification, in: J. Ralyté, J./Mirbel, I./Deneckère, R. (Hrsg.), ME 2011, IFIP AICT 351, IFIP International Federation for Information Processing, Springer, S. 204-218.

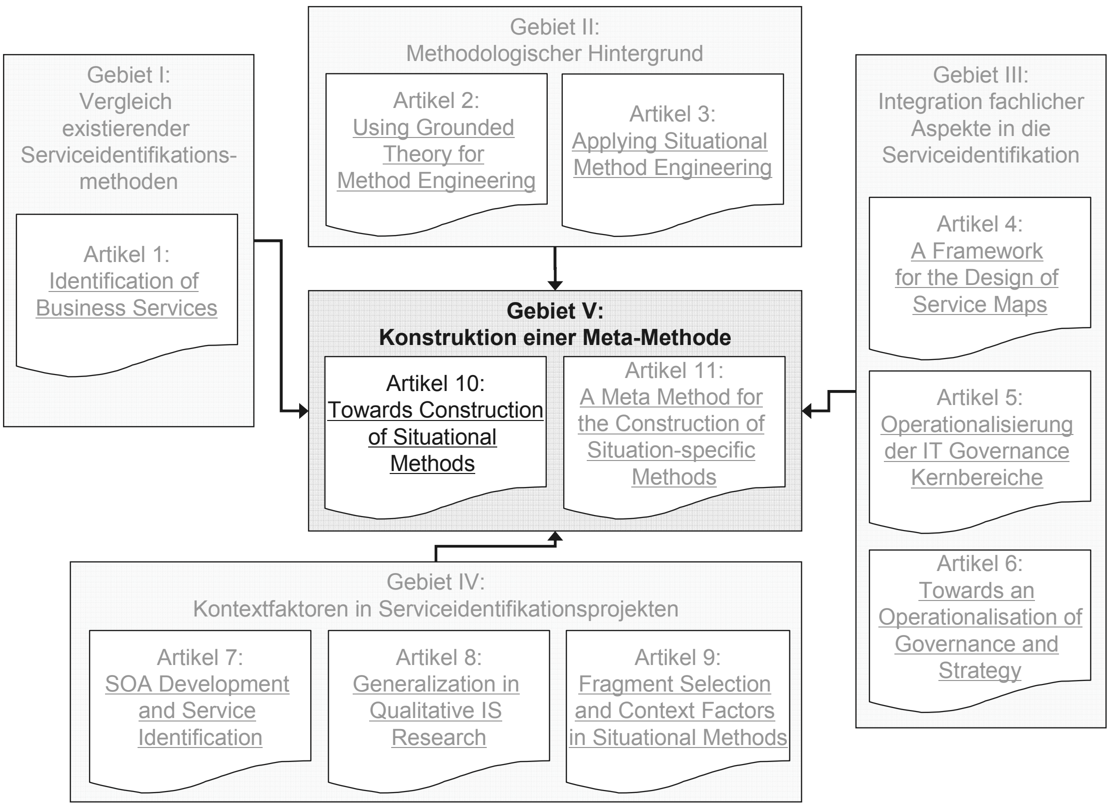

Abbildung 15: Artikel 10 im Kontext der kumulativen Dissertation

Towards Construction of Situational Methods for Service Identification

René Börner

ProcessLab, Frankfurt School of Finance & Management,

Sonnemannstraße 9-11, 60314 Frankfurt am Main, Germany

r.boerner@fs.de

Abstract

The service-oriented paradigm plays an increasingly significant role in designing and governing IT architectures in organizations. The identification of services belongs to the most important parts of the service management lifecycle and is essential for the successful implementation of service-oriented architectures (SOA). However, existing methods for service identification mostly ignore situation-specific factors for such projects. Situational method engineering can be used to design a meta method to support the development of situation-specific methods for service identification. Based on a literature review and two case studies, this paper elaborates on context factors and SOA implementation goals being constituting elements of situations. These situations are tailored to the service identification domain. Applying this meta method in concrete project situations will help to engineer appropriate methods for service identification.

Keywords

Situational method engineering, service-oriented architectures, service identification, context factors, meta method, SOA implementation goals

1 Introduction

Service orientation is currently a dominating paradigm for enterprise and IT architectures. Service-oriented architectures promise a greater flexibility of IT and a faster adoption to changing business needs. The identification of services is one of the most important steps for successful SOA implementations and many authors have developed methods for this purpose (for an overview see [1]). Interestingly, most of these methods are based on a one-fits-all approach and do not consider a configuration of methods depending on different circumstances.

The field of situational method engineering (SME) offers opportunities to overcome these shortcomings. A central aspect behind it is that a fixed method is not suitable for all situations that occur in reality. Thus, methods have to be adaptable to different kinds of situations. This paper elaborates on context factors and SOA implementation goals to identify situations in the field of service identification. These are necessary to support a reasonable configuration of fragments that are developed as part of the meta method for the configuration of methods for service identification. Identifying situations is pivotal for this meta method and the focus of this paper. Designing suitable fragments complements the meta method and will be discussed briefly herein.

The paper is structured as follows: Section 2 discusses related work in the fields of SME and service identification, outlines the research design and describes the scope of this paper. In section 3, context factors and SOA implementation goals – both being defining parts of a situation – are elaborated in detail. The design of method fragments for service identification is presented in section 4. Section 5 reflects on limitations, proposes avenues for future research and provides a conclusion.

2 Foundations of a New Meta Method

This section discusses related literature in the field of SME and service identification. The research process used to derive the context factors is outlined. Furthermore, scope and goals of the meta method are defined.

2.1 Situational Method Engineering and Configurability in Existing Approaches

It is commonly accepted that no universal method constructed at time (t_1) can fit every conceivable situation in which it is applied in time (t_2) [2]. Actually, it is quite improbable that a rigid method developed from theory is applicable in a concrete setting without modification [3] and therefore, the concept of SME emerged (see e.g. [4]). The central aspect behind it is that a fixed method is not suitable for all situations that occur in reality. Thus, methods have to be adaptable to different kinds of situations. To support this adaptability, smaller parts of a method – so called method fragments – are created and can be composed depending on the situation at hand [5]. Method fragments consist of the four elements activity, technique, role, and result [6].

Unfortunately, the term method fragment is inconsistently used in literature [7]. Ågerfalk et al. [8] define method fragments as "standardized building blocks based on a coherent part of a method" (p. 360). A situational method can be constructed by combining a number of method fragments. For the purpose of this paper, any reasonable combination of method elements representing a coherent part of a method shall be referred to as method fragment [8].

According to Bucher et al. [9], there are two adaptation mechanisms to engineer a situational method, namely *situational method configuration* and *situational method composition*. Situational method configuration follows the so called adaptive principle. This means that a base method is created at design time (t_1) and configured in certain contexts at time (t_2). For situational method configuration, situational changes to a base method have to be foreseen and planned when a situational method is developed at time (t_1). In contrast, situational method composition provides for a spontaneous combination of method fragments (orchestration) that does not have to be foreseen at (t_1). There is no pre-defined base method that is adapted. Instead, method fragments are combined and aggregated as required at (t_2). Börner [10] suggests a third possibility, in which precomposed methods are assigned to situations in (t_2). Although methods are defined in (t_1) already, there is no pre-configured base method in this approach.

Many authors agree that characteristics of a project have to be defined in order to describe a situation [11-13]. Still, according to [9] they do not explicitly say what constitutes a situation. For the purpose of this paper, Börner's [10] concept of a situation is used. It defines *context factors* and *SOA implementation goals* as two determining factors that constitute a situation.

The analysis and comparison of existing service identification literature is the basis for the development of a meta method for situational methods conducted in this paper. Additionally to the literature analyzed by [1], two currently published approaches [14, 15] were included in this paper. All approaches were examined considering their description of activities, roles, techniques and results as well as the configurability of the presented methods. The first four criteria were chosen because they are commonly used elements for methods and method fragments [6, 16]. Configurability is frequently regarded important in SME literature. However, in the field of SOA and especially in service identification a lack of configurability of methods can be stated.

Since a*ctivities* are the focus of all compared approaches, they are described precisely in most cases. Many *techniques* and intermediate *results* are usually provided as well. The most striking feature is the almost complete absence of *roles*. Although many authors discuss the importance of business and IT alignment, all compared service identification approaches tend to underestimate the significance of properly assigning roles to respective activities and techniques. Roles will play an important role when method fragments are assigned to situations.

The *configurability* describes the possibility to choose adequate fragments that suit the situation at hand and arrange them in an adequate sequence. The latter means that activities used in an approach do not have to be executed in a linear order but can be used iteratively and hence allow for loops or iterations. Although hints on possible configurations can be found in some places, none of the authors explicitly incorporates the former into their approach. Since the goal of this paper is to guide the engineering of situational methods, configurability of service identification methods will be considered a crucial feature.

2.2 Research Design

The development of a meta method for service identification methods presented herein is based on a hybrid research approach combining several research methods to gain a richer understanding of the topic [17]. Construction of the meta method supported by SME belongs to the realm of *design science*. The derivation of relevant context factors builds on both *desk research*, i.e. literature reviews, and *case study research*. Since the identification of relevant context factors is an explorative goal, a case study approach was deliberately chosen to give the results an empirical grounding. Case studies are appropriate in this respect because they "provide descriptions of phenomena" [18]. Furthermore, case studies are particularly relevant for research in its "early, formative stages" [19] which applies to the field of SOA [20].

The case studies were conducted in two SOA implementation projects in Australian companies providing completely different environments. One of the companies is a small data provider; the other is one of Australia's biggest insurance companies. The significantly diverse settings of both cases opened up a continuum [21] of instantiations for identified context factors, i.e. their parameter values. At the insurer, researchers conducted an action research study. They actively participated in the project and helped test and apply a service analysis & design methodology developed by them previously. In the second case study, the most important sources of evidence have been interviews that were conducted shortly after the project had been completed. The data provider's employees and researchers were the interview partners. The researchers had helped the data provider to implement an SOA in order to enable the retrieval and analysis of heterogeneous data from different sources (grid environment) spontaneously in an unforeseeable fashion (ad-hoc).

The interviews have been transcribed afterwards and analyzed along with all other documentation and reports. In an iterative approach, relevant context factors were determined based on this data and compared with related literature. The identification of such factors (concepts) was conducted by employing techniques from grounded theory, for example, open and axial coding [22], and interpretative techniques [23]. Even though these coding techniques were not used to their fullest extent, the general approach and respective tools supported the assignment of statements from the interviews and documents to concepts. The goal was to detect relevant particulars within the case data and to identify relevant concepts [24], i.e. context factors.

2.3 Goals and Scope of the Meta Method

Due to the lack of configurability in existing approaches, this paper argues that situational method engineering can support methods for service identification that suit certain project situations and are thus situation-specific. Particularly, context factors (including their respective parameter values) and SOA implementation goals that jointly determine a situation are the focus. They provide the basis for the intended meta method that is subject to ongoing research.

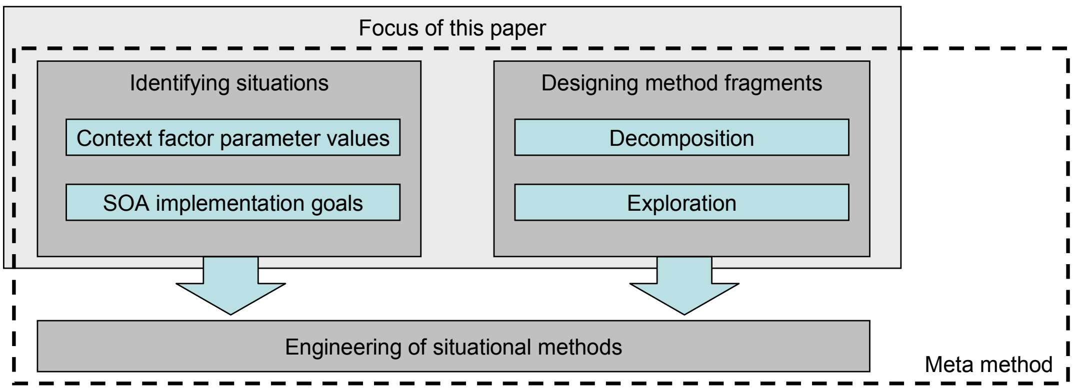

Figure 1. Scope of the Meta Method and Focus of This Paper

This meta method encompasses the identification and description of context factors, their value parameters and SOA implementation goals. Together, combinations of these parts constitute a situation. Every instantiation of a method will rely on the situations that will be developed in section 3 since context factors, value parameters and SOA implementation goals belong to the service identification domain. Thus, the latter are design elements of this meta method. Apart from situations, the meta method also includes descriptions of method fragments. Principles of fragment design will be outlined and shown exemplarily in section 4.

Figure 1 illustrates that the meta method encompasses the identification of situations and the design of method fragments. Both are necessary for the configuration of situational methods. Focus of this paper is the identification of situations in the domain of service identification based on relevant context factors and SOA implementation goals. Moreover, the design of method fragments as part of a meta method will be shown exemplarily herein. Therefore, two ways of method re-engineering, namely decomposition and exploration, are presented. A generally valid description of how to engineer situational methods in any conceivable situation is left to further research.

3 Identifying Situations

Following [10], the combination of context parameters and SOA implementation goals determines situations. Figure 2 illustrates the five necessary steps (a) to (e) to identify situations. These steps are introduced briefly herein (for a more detailed explanation see [10]):

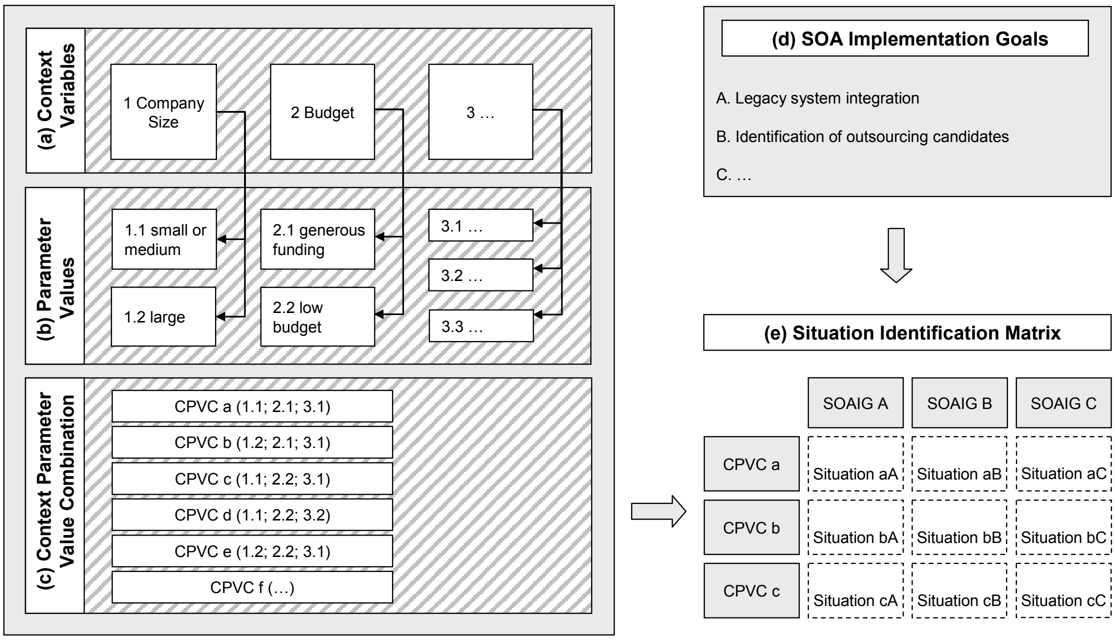

Figure 2. Five Steps to Identify Situations (following [10], p. 5)

(a) The context variables (contingency factors) that can influence the SOA implementation project at hand are defined.

(b) The context variables identified in (a) have certain parameter values (instantiations) that are defined in this second step.

(c) All context parameter values are combined with one another. These context parameter value combinations (CPVC) serve as one input for the situation identification matrix.

(d) Possible SOA implementation goals are identified in this step. They are the second input for the situation identification matrix.

(e) The situation identification matrix illustrates all possible combinations of CPVCs and SOAIGs retrieved in steps (c) and (d).

Context factors, respective parameter values and SOA implementation goals that are relevant to the service identification domain, and therefore an important part of the meta method presented herein, will be introduced in the following subsections. A combination of concrete parameter values (e.g. a small company with low budget, no industry-specific legal restrictions producing only one product) and an SOA implementation goal (provision of services for third parties) constitutes a specific situation.

3.1 Context Factors

Based on related literature and two recently conducted case studies described in section 2.2, the importance of the most influential context factors is discussed in the following. Although these context factors might not be unique to the field of SOA, their consideration in service identification methods will improve applicability of the latter.

It is a generally held belief that the ***company size*** is a considerable contextual factor in many kinds of software projects [25] such as service identification in SOA implementation projects. Whereas Sedera [26] proposes three classes of company size, this paper follows Welsh and White [27] who suggest only two. Hence, companies are differentiated into *small and medium-sized enterprises* on the one hand and *large companies* such as multi-national enterprises on the other hand. The former shall comprise organizations with up to 250 employees [28].

The ***budget*** of a project plays an important role when it comes to choosing necessary steps and general proceeding of a method for service identification. Generally, a generous budget that allows for an extensive time frame provides the opportunity for a thorough and systematic application of identification methods. One would expect the utilization of many techniques in order to ensure a high quality of implemented services. A detailed analysis of available strategic and technical documents would be typical in such circumstances. Literature confirms that the budget has implications on the number of available staff, the time pressure and the possibility to incorporate external help from consultants [29].

Some activities have to be carried out regardless of the budget. Still, there might be obligatory as well as optional techniques which support the activity. Consequently, an exhaustive use of techniques would only be chosen if financial resources are easily available. The parameter values *generous funding* and *low budget* will thus be used for this context variable.

Depending on a company's strategy, services can be provided for different ***service consumers***, for example other divisions (internally), third parties (externally) or both. Strategically important services that lead to a competitive advantage in the market should only be available to end-consumers but not to competitors. A clear distinction between services offered to internal customers, other companies in a value chain or end-consumers is indispensable. If it is known in advance that a service will be offered internally only, a number of activities and results such as the creation of an inter-organizational service map are not applicable in this situation. Moreover, there might be legal constraints that only apply if services are offered to third parties and thus demand an examination of these regulations. An analysis of consumer interaction is always essential whereas the “line of visibility” is much more important if services are exposed to external customers [30].

For the purpose of this paper, service consumers will be divided into *internal* and *external consumers*. Additionally, the case of a service being offered to *both* is considered.

Skills and experience with both service-oriented architectures and business process management significantly influence the proceeding of service identification. Limited SOA experience on the side of employees often leads to a technical SOA understanding. Project teams that are more familiar with the service-oriented paradigm are more likely to succeed in combining technical aspects with a Business Process Management perspective in mind. Hence, software services that support business processes or at least sub-processes can be the goal of their analysis. Their identification usually includes activities related to the analysis of process models and involves not only IT-related, but also business-related staff roles. If the service identification is limited to a rather technical point of view, the set of method fragments to be considered will therefore be a different one.

The configuration of situational methods and choice of fragments is affected, if for example certain roles cannot be occupied by available employees. Limited employee skills can necessitate external support by consultants. Although this enables the application of certain fragments, this option might be limited by the project budget and might therefore not be feasible. Parameter values for this context factor are *SOA skills available, BPM skills available, both skills available* and *none available*.

Furthermore, the ***SOA maturity level*** a company has achieved is seen as a further influential factor on the delivery strategy of SOA [31]. Thus, it plays an important role in the configuration of methods for service identification.

SOA maturity models are used to classify the status of SOA implementations within a company. This paper will use the Service Integration Maturity Model (SIMM) [32] to distinguish advanced organizations with *level 4 to 7* from less mature organizations (*level 1 to 3*). The former are likely to use more sophisticated and strategy-oriented fragments. The latter usually use more technically-oriented techniques and thus other fragments.

On the one hand, ***compliance*** issues can arise from legal obligations and regulatory restrictions. These differ among countries and especially companies that operate in more than one country have to consider legal demands arising from that. In many countries, all companies have to obey certain rules as far as the confidentiality of customer data is concerned. Additionally, some industries such as banking or pharmaceuticals have to adhere to special regulations. Finally, regulations can arise from the fact that a company is listed on a stock exchange, i.e. it also depends on its legal form. On the other hand, internal policies may require corresponding method fragments that address issues like service ownership. Three parameter values will be used for this context factor, namely *standard legal compliance*, *special regulations due to industry*, *legal form or international operations* and *internal policies*.

Another important context variable is the *existence* of a designated ***IT department*** and thus the degree of centralization of the IT infrastructure. In a small company that lacks an IT department, methods have to be adapted to accommodate for this circumstance. Larger organizations usually have such an IT division or are structured along the lines of business. On the one hand a high degree of centralization or the existence of a central division supervising and governing IT implementation throughout a company usually leads to more transparency. Frequently, at least some information on applications and data is readily available. This can be used as input for service identification method fragments. On the other hand, some fragments demand certain roles such IT administrators or newly composed units consisting of business and IT employees (see also [33]). In a small company that lacks an IT department, these method fragments are frequently not applicable.

Context Variable	**Parameter Value**
1 Company size	1.1 Small or medium-sized enterprise
	1.2 Large company
2 Service consumers	2.1 Internal consumer
	2.2 External consumer
	2.3 Internal and external consumers
3 Budget	3.1 Generous funding
	3.2 Low budget
4 Skills and experience	4.1 SOA skills available
	4.2 BPM skills available
	4.3 Both skills available
	4.4 None available
5 SOA maturity level	5.1 SIMM level 1-3
	5.2 SIMM level 4-7
6 Compliance	6.1 Standard legal compliance
	6.2 Special regulations
	6.3 Internal policies
7 IT department	7.1 Existent
	7.2 Not existent
8 Interaction	8.1 Customer interaction
	8.2 Employee interaction
	8.3 Customer and employee interaction
9 Organizational structure	9.1 One product company
	9.2 Multiple product company

Table 1. Context Variables and Respective Parameter Values

Varying degrees and forms of ***interaction*** with both customers and employees necessitate the use of different method fragments. In some cases employees are not directly involved in service delivery because the services are very fine-grained and fully automated. The coarser-grained services are, the greater is the possibility that they are only semi-automated or manual and subsequently interact with employees. Customer interaction can be of high importance when the composition of services by the end user is a primary goal. In general, a customer interaction can be obligatory in some places or can happen “on demand” if required [34]. If customer interaction is a major issue for the identification of services in a situation at hand, respective method fragments (e.g. swim lane diagrams that show interfaces to customers) are crucial for a successful implementation. Thus, *customer interaction*, *employee interaction* and a *combination of both* are differentiated for the purpose of this paper.

In a company specialized on one product only, an analysis of a service’s reusability is trivial in most cases. The same analysis is much more complex when looking at companies with a wide range of products. An organization can, e.g., be structured by products (business lines), regions, functions or customer groups [35, 36]. Even a multidimensional structure combining two or more dimensions of the above is not uncommon. Thus, the ***organizational structure*** can be an important factor when it comes to service identification. Herein, *one product companies* and *multi product companies* are differentiated.

Table 1 gives an overview of the context variables used and their respective parameter values. After identifying the context factors that are one part of a situation, the next section will elaborate possible goals for the implementation of service-oriented architectures that are the second constituting part of a situation.

3.2 SOA Implementation Goals

The second constituting element of a situation are SOA implementation goals. Depending on the purpose of an SOA implementation, the identification of services can necessitate the application of different method fragments. Many such goals can be found in related literature and the case studies also confirmed some of them. In the following, these goals and their influence on a situational method configuration will be discussed.

The ***integration of legacy systems*** is a frequently mentioned goal of SOA implementations [37, 38]. Especially in medium-sized and large enterprises, IT architectures have developed over years or even decades. In the absence of a central governing body, manifold isolated applications were developed and implemented which led to a plethora of problems. New functionalities and updates have to be made separately for each system which causes high maintenance costs. In some cases, it is difficult to find specialists who are able to administer for instance cobol code. Due to their restricted function-oriented view, employees do not know about IT systems of other divisions. This redundancy causes high costs because of unnecessary licensing fees.

Hence, integrating existing applications plays a major role in enterprise IT architectures and is one of the reasons for SOA implementation projects. In this case, techniques such as asset analysis and results that illustrate dependencies of the existing IT infrastructure are crucial parts of the service identification. The knowledge of IT experts about technical interfaces is indispensable.

The ***identification of outsourcing candidates*** is another goal for SOA implementations [39]. In this case, costs, performance and strategic relevance of services must be analyzed. On the one hand, based on a business process analyses the exact scope of the outsourcing activity has to be defined. A strategic make-or-buy decision determines which parts of the process are performed within the organization and which parts shall be outsourced to service providers. On the other hand, an outsourcing candidate needs clearly defined technical interfaces. Inputs and outputs of automated services provided by a third party have to be explicated in service level agreements. These outsourcing considerations demand fragments that produce for instance inter-organizational service maps and incorporate strategic aspects as well as detailed technical descriptions.

The ***agility and flexibility of business processes*** is a competitive advantage and strongly tied to the concept of SOA [40]. An alignment of business and IT is a necessary precondition to achieve this flexibility. Therefore, a company's strategy, i.e. a business process perspective, has to be considered. An enterprise-wide governance of the IT infrastructure is indispensable to provide for this agility. Hence, fragment results such as service ownership models [41] have to be used.

In contrast to an enhanced flexibility on process services level, the ***standardization*** of basic services is meant to avoid redundancies in development and maintenance of IT and thus to reduce costs significantly [42]. The goal is to improve efficiency by reusing a service in as many processes as possible. However, a customer should not be limited in his choice of varieties. A faster processing through increased efficiency should lead to a higher customer satisfaction. Therefore, services that directly interact with customers should not be standardized. This makes fragments for the analysis of the line of interaction and the line of visibility indispensable.

A completely different perspective is taken by companies that aim at the ***provision of services for third parties***. The former specialize on a small part of a value chain concentrating on their core competencies. These companies are able to generate economies of scale by providing services for many other companies typically – not necessarily – belonging to the same industry sector. Hence, the focus here is again on inter-organizational and strategic instruments. Services must be easily exposable to third parties, i.e. interfaces have to be well-defined and performance has to be readily measurable. Method fragments should thus concentrate on interaction, interface analysis and the strategic value of providing a service to third parties.

4 Designing Method Fragments for Service Identification

In order to design situation-specific methods for service identification, method fragments that support this identification have to be provided. There are basically two possibilities to design these method fragments [43]. On the one hand, fragments can be re-engineered from existing methods. On the other hand, they can be designed from scratch in case no experience exists, i.e. no fragments or elements can be retrieved from existing approaches.

As shown in section 2.1, literature provides a number of methods for service identification. Although they include many effective method elements and fragments, the lack of configurability is a major shortcoming. Thus, the following design of method fragments will concentrate on re-engineering of existing methods rather than on ad-hoc construction. Ralyté [44] identifies two ways to design method fragments from existing methods, namely *decomposing models from existing methods* and *exploring different possibilities to apply a model* (p. 5). In the following, both will be introduced. Two examples of method fragments will show the applicability to the service identification domain.

4.1 Decomposition

Identifying fragments through decomposition is supposed to be easier than creating new ones through exploration and should thus be the first step. Most of the fragments that can be found in existing service identification approaches concentrate strongly on the result of activities and are thus strongly product-driven. In these cases the process part – including roles and techniques – has to be conceptualized in order to obtain fragments. In cases where a fragment is identified by process model decomposition, the product part has to be elaborated since the processes are already available [44]. The following is one example of a fragment that was decomposed from an existing method [30] and enhanced as far as the process part is concerned.

Fragment 1: Overview of Existing Process Models

Description:	If there are documented business processes, these should be used for the further analysis to save time and money if possible.
Input:	Meaningful documents of existing business processes from formerly conducted Business Process Management (BPM) projects
Preconditions:	If no BPM projects had been conducted before, it is necessary to identify business processes before using this fragment.
Taken from:	Klose, Knackstedt, Beverungen (2007)
Design:	In the first phase “preparation” of their approach, Klose et al. [31] include the task “prepare existing process models” into their procedure model (p.1804). Since the authors describe activity, techniques and results, the fragment is identified by product decomposition. Only the role to perform the activity has to be added in order to complete the fragment.
Activity:	Preparation
Role:	Employee of the business department
Technique:	Prepare existing process models
Result:	Consolidated and complete set of hierarchical process models, modeling conventions

Figure 3. Method Fragment 1

4.2 Exploration

After identifying as many fragments as possible in this first step, exploration is used to find additional fragments on the basis of the elements used in existing approaches. Thus, concrete activities, roles, techniques and results found in different sources are extracted and subsequently used to design new fragments. A comprehensive overview of elements cannot be provided herein, but examples for these constituting elements of methods will be given in the following. The most important sources are the literature on service identification and the two case studies described in section 2.

Activities: Many approaches use activities such as *service analysis* and *service categorization*. *Preparation* is also a common activity to be found in literature. However, activities like *goal definition* or *develop SOA strategy* can be found in only one approach, respectively. Some authors use the word activity for very detailed descriptions of how something has to be done. In the definition used herein this would rather be a technique.

Furthermore, one and the same activity might have different names in different approaches. This makes consolidation a difficult task.

Roles: Despite their importance, roles only occur in four of the seven compared approaches. Sometimes they are hard to identify as such because the notion of *consumer view* might be used where *employee of the business department* would be a better description. Besides the *employee of the IT department*, roles like *project manager* were important in the case studies. Related literature additionally suggest new roles such as a *service design unit* for certain activities [33].

Techniques: Consolidating all techniques utilized in literature is difficult since most approaches offer plenty of techniques with sometimes overlapping components and scopes. However, there seem to be some typical and wide-spread techniques that are common to many approaches such as *decomposition of business processes* and *asset analysis*. A couple of other techniques that were frequently encountered are *goal service modeling* and *use case modeling*. It is noteworthy that the scope of the listed techniques can differ considerably. Using a *governance questionnaire* is a straight forward and unambiguous procedure with limited scope and little room for interpretation. The *decomposition of business processes* is much more complex and likely to yield different outcomes depending on who actually conducts the task.

Results: Similar to the significant number of techniques, there are many results presented as part of the methods. These results are outputs of respective activities and techniques and can be an input for the next activity. Thus, they are a crucial link between method fragments. The results themselves are quite different in nature and reach from technical *interface descriptions* to comprehensive *SOA strategy documents* or *network models* on an inter-organizational level. *Use cases, reference processes* and *activity diagrams* are examples for other results.

Fragment 2: IT Governance Analysis

Description:	An organization can have manifold demands when it comes to implementing new IT infrastructures. Using agile methods for example could be one imperative. Technical restrictions, programming language, interfaces or naming conventions can impose restrictions on IT projects. Hence, these IT governance issues can be covered by a questionnaire and are incorporated in this fragment. Employees of the business and the IT department jointly forming a so-called service design unit (SDU) form an important role to successfully design services based on this questionnaire.
Input:	Information on IT governance, IT strategy, SOA strategy
Preconditions:	Derived from an organization's strategy, the IT strategy must be defined and documented before an analysis for service identification can be performed. Often, conventions and principles regarding the implementation and development of IT are not explicated in readily available documents. Therefore, it might be necessary to interview (IT) managers in order to retrieve necessary information.
Taken from:	Klose, Knackstedt and Beverungen (2007), Kohlborn, Korthaus, Chan and Rosemann (2009), Arsanjani, Ghosh, Allam, Abdollah, Ganapathy and Holley (2008), Kohlmann and Alt (2007), Alter, Börner and Goeken (2009)
Design:	Since the scope of this fragment quite wide, elements have been selected from different approaches. Domain decomposition for instance could be found in three existing methods. Special roles and techniques such as an SDU or a governance questionnaire, respectively, have been taken from related literature that does not present a comprehensive method for service identification but deals with governance aspects in general.
Activity:	Service design
Role:	SDU, employee of the IT department, (IT) manager
Technique:	Governance questionnaire, domain decomposition, naming
Result:	Naming conventions, service ownership list, modeling conventions, design principles

Figure 4. Method fragment 2

Based on the elements identified previously, more fragments can be designed. Elements that are used in fragments created through exploration are taken from more than one existing approach because if they were to be found in one single approach, the fragment could have been derived by decomposition as shown in section 4.1. Fragment 2 is one example for a fragment designed by exploration.

5 Conclusion and Further Research

This paper outlined the necessity of designing situation-specific methods for service identification since a literature review attested a missing configurability of existing approaches. Hence, a meta method should guide the engineering of such situational methods. An important part of this meta method is the definition of situations in the domain of service identification. Thus, the identification and discussion of context factors and SOA implementation goals was the centerpiece of this work. The idea of how to design method fragments was explained briefly and shown at two examples.

On the way to creating a meta method for the construction of situation-specific methods for service identification there are a number of limitations that should be considered. The identified context factors are based on an extensive literature research. Moreover, their significance was supported by two case studies where qualitative research methods were used. An investigation of relationships and interdependencies of these context factors and the SOAIG is subject to ongoing research.

The number of situations has to be restricted to make the approach feasible. Following [10], the context factors, their parameter values and SOA implementation goals presented herein lead to 17,280 situations. Therefore, the relevance of context factors should be scrutinized through further research. The same is true for parameter values und SOA implementation goals. The aforementioned analysis of interdependencies is also likely to reduce the number of context factor parameter values and thus the number of situations that have to be considered.

Decomposed method fragments extracted from existing methods usually have been applied to real-life projects before and are thus quite reliable. Those fragments created through exploration could be criticized for being an arbitrary combination of elements without proper foundation. Both the exemplary fragments presented in section 4 and further ones that are currently developed are based on literature and the experience of two case studies. Proving quality is difficult and indeed, completeness of a method fragment base that is proposed by many authors [45] cannot be guaranteed. Thus, it is important to feed back information from projects that will use the meta method in future. This will improve fragments and give them a stronger empirical grounding.

As demonstrated, ideas from SME can contribute significantly to the field of service identification by supporting the design of situational methods. In order to build a comprehensive meta method, experience and expert knowledge from the service-oriented domain have to be incorporated in this meta method. Further case studies or action research could support an empirical validation of this meta method.

References

1. Börner, R., Goeken, M.: Identification of Business Services - Literature Review and Lessons Learned. In: 15th AMCIS, Paper 162. San Francisco, California (2009)
2. Fitzgerald, B., Russo, N.L., O'Kane, T.: Software Development: Method Tailoring at Motorola. Communications of the ACM. 46, 65-70 (2003)
3. Aydin, M.N.: Examining Key Notions for Method Adaptation. In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences, pp. 49-63. Springer, Boston (2007)
4. Harmsen, F., Brinkkemper, S., Oei, H.: Situational Method Engineering for Information System Project Approaches. In: Verrijn-Stuart, A.A., Olle, T.W. (eds.) Methods and Associated Tools for the Information Systems Life Cycle, pp. 169-194. Elsevier Science B.V., Amsterdam (1994)
5. Ralyté, J., Rolland, C.: An Approach for Method Engineering. In: 20th International Conference on Conceptual Modelling, pp. 471-484. Springer, Yokohama, Japan (2001)
6. Cossentino, M., Gaglio, S., Henderson-Sellers, B., Seidita, V.: A metamodelling-based approach for method fragment comparison. In: 11th International Workshop on Exploring Modeling Methods in Systems Analysis and Design at CAiSE, Luxembourg (2006)
7. Sunyaev, A., Hansen, M., Krcmar, H.: Method Engineering: A Formal Approach. In: 17th International Conference on Information Systems Development, Paphos, Cyprus (2008)
8. Agerfalk, P.J., Brinkkemper, S., Gonzalez-Perez, C., Henderson-Sellers, B., Karlsson, F., Kelly, S., Ralyté, J.: Modularization Constructs in Method Engineering: Towards Common Ground? In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences, pp. 359-368. Springer, Boston (2007)

9. Bucher, T., Klesse, M., Kurpjuweit, S., Winter, R.: Situational Method Engineering - On the Differentiation of "Context" and "Project Type". In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences, pp. 33-48. Springer, Boston (2007)
10. Börner, R.: Applying Situational Method Engineering to the Development of Service Identification Methods. In: 16th AMCIS, Paper 18. Lima, Peru (2010)
11. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information & Software Technology. 38, 275-280 (1996)
12. Punter, T., Lemmen, K.: The MEMA-model: towards a new approach for Method Engineering. Information & Software Technology. 38, 295-305 (1996)
13. Karlsson, F., Agerfalk, P.J.: Method Configuration - Adapting to Situational Characteristics while Creating Reusable Assets. Information & Software Technology. 46, 619-633 (2004)
14. Kohlborn, T., Korthaus, A., Chan, T., Rosemann, M.: Identification and Analysis of Business and Software Services - A Consolidated Approach. IEEE Transactions on Services Computing. 2, 50-64 (2009)
15. Inaganti, S., Behara, G.K.: Service Identification - BPM and SOA Handshake. BPTrends. 3, 1-12 (2007)
16. Goeken, M.: Entwicklung von Data-Warehouse-Systemen. Anforderungsmanagement, Modellierung, Implementierung. Deutscher Universitäts-Verlag, Wiesbaden (2006)
17. Mingers, J.: Combining IS Research Methods: Towards a Pluralist Methodology. Information Systems Research. 12, 240-259 (2001)
18. Darke, P., Shanks, G., Broadbent, M.: Successfully completing case study research: combining rigour, relevance and pragmatism. Information Systems J. 8, 273-289 (1998)
19. Benbasat, I., Goldstein, D.K., Mead, M.: The case research strategy in studies of information systems. MIS Quarterly. 269-386 (1987)
20. Luthria, H., Rabhi, F.A.: Building the Business Case for SOA: A Study of the Business Drivers for Technology Infrastructure Supporting Financial Service Institutions. In: Kundisch, D., et al. (eds.) Enterprise Applications and Services in the Finance Industry, pp. 94-107. Springer (2009)
21. Yin, R.K.: Case Study Research - Design and Methods. 3 ed. Vol. 5. SAGE Publications, Thousand Oaks London New Delhi (2003)
22. Strauss, A., Corbin, J.M.: Basics of qualitative research: Grounded theory procedures and techniques. Sage Publications, Thousand Oaks, CA (1990)
23. Walsham, G.: Interpretive case studies in IS research. Nature and Method. European Journal of Information Systems. 74-81 (1995)
24. Eisenhardt, K.M.: Building Theory from Case Study Research. The Academy of Management Review. 14, 532-550 (1989)
25. DeLone, W.H.: Determinants Of Success For Computer Usage In Small Business. MIS Quarterly. 12, 50-61 (1988)
26. Sedera, D.: Does Size Matter? Enterprise System Performance in Small, Medium and Large Organizations. In: 2nd Workshop on 3rd Generation Enterprise Resource Planning Systems, Copenhagen, Denmark (2008)
27. Welsh, J.A., White, J.F.: A amall business is not a little big business. Harvard Business Review. 59, 18-32 (1981)
28. Laukkanen, S., Sarpola, S., Hallikainen, P.: Enterprise size matters: objectives and constraints of ERP adoption. J of Enterprise Information Management. 20, 319-334 (2007)
29. Becker, J., Knackstedt, R., Pfeiffer, D., Janiesch, C.: Configurative Method Engineering. In: 13th AMCIS, Paper 56. Keystone, Colorado (2007)

30. Klose, K., Knackstedt, R., Beverungen, D.: Identification of Services - A Stakeholder-Based Approach to SOA Development and Its Application in the Area of Production Planning. In: 15th ECIS, pp. 1802-1814. St. Gallen, Switzerland (2007)
31. Terlouw, J., Terlouw, L., Jansen, S.: An Assessment Method for Selecting an SOA Delivery Strategy: Determining Influencing Factors and Their Value Weights. In: 4th International Workshop on BUSITAL, Amsterdam, The Netherlands (2009)
32. Arsanjani, A., Holley, K.: The Service Integration Maturity Model: Achieving Flexibility in the Transformation to SOA. In: IEEE International Conference on Services Computing (SCC'06), pp. 515. Chi-cago, IL (2006)
33. Börner, R., Looso, S., Goeken, M.: Towards an Operationalisation of Governance and Strategy for Service Identification and Design. In: 13th IEEE International EDOC Conference, pp. 180-188. Auckland, New Zealand (2009)
34. Leyer, M., Moormann, J.: Facilitating operational control of business services: A method for analysing and structuring customer integration. In: 21st ACIS, Paper 42. Brisbane, Australia (2010)
35. Pitts, R.A., Lei, D.: Strategic Management. 4 ed. South-Western, Mason, Ohio (2005)
36. Schermerhorn, J.R.: Management for Productivity. Wiley, New York, NY (1993)
37. Erl, T.: Service-Oriented Architecture - A Field Guide to Integrating XML and Web Services. Prentice Hall, Upper Saddle River, NJ (2004)
38. Heutschi, R.: Serviceorientierte Architektur. Springer, Berlin Heidelberg (2007)
39. Beverungen, D., Knackstedt, R., Müller, O.: Entwicklung Serviceorientierter Architekturen zur Integration von Produktion und Dienstleistung. WI. 50, 220-234 (2008)
40. Papazoglou, M.P.: Service-Oriented Computing: Concepts, Characteristics and Directions. In: 4th International Conference on Web Information Systems Engineering, pp. 3-12. IEEE Computer Soci-ety, Rome, Italy (2003)
41. Kohlmann, F., Alt, R.: Deducing Service Ownerships in Financial Networks. In: 15th AMCIS, Paper 518. San Francisco, CA (2009)
42. Bieberstein, N., Bose, S., Walker, L., Lynch, A.: Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals. IBM Systems Journal. 44, 691-708 (2005)
43. Henderson-Sellers, B., Ralyté, J.: Situational Method Engineering: State-of-the-Art Review. Journal of Universal Computer Science. 16, 424-478 (2010)
44. Ralyté, J.: Towards Situational Methods for Information Systems Development: Engineering Reusable Method Chunks. In: International Conference on Information Systems Development (ISD'04), pp. 271-282. Vilnius Technika, Vilnius, Lithuania (2004)
45. Deneckère, R., Iacovelli, A., Kornyshova, E., Souveyet, C.: From Method Fragments to Method Services. (2009)

6.3 Artikel 11: A Meta Method for the Construction of Situation-Specific Methods for Service Identification

Börner, R. (2011): A Meta Method for the Construction of Situation-Specific Methods for Service Identification, Unveröffentlichter Arbeitsbericht.

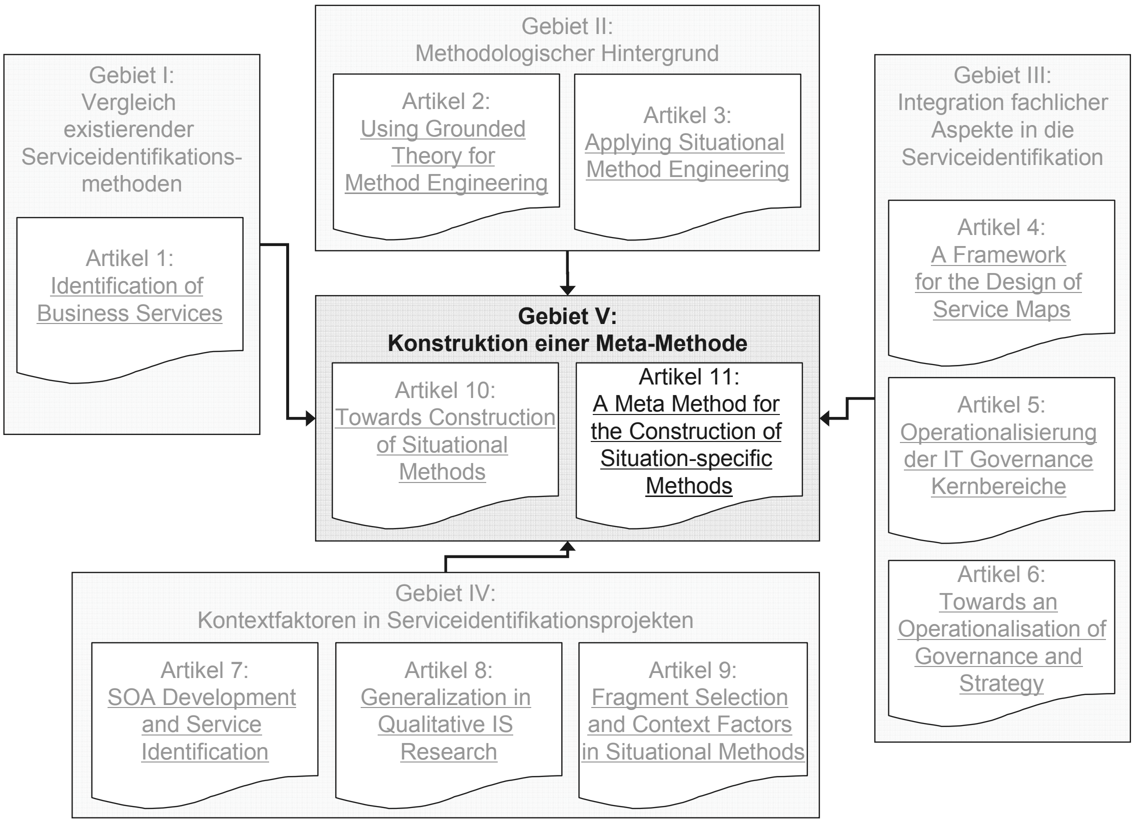

Abbildung 16: Artikel 11 im Kontext der kumulativen Dissertation

A Meta Method for the Construction of Situation-Specific Methods for Service Identification

René Börner

ProcessLab, Frankfurt School of Finance & Management,

Sonnemannstraße 9-11, 60314 Frankfurt am Main

r.boerner@fs.de

February 2011

Table of Contents

Index of Figures.....2
Index of Tables.....2
1 Introduction.....3
2 Design Science and Research Methodologies.....4
3 Foundations of a New Method.....7
3.1 Basics of ME and SME.....7
3.2 Existing Methods for Service Identification.....12
3.3 Method vs. Meta Method.....14
4 Identifying Situations.....15
4.1 Context Factors.....16
4.2 SOA Implementation Goals.....22
4.3 Restricting the Number of Situations.....23
5 Method Fragments for Situation-Specific Methods for Service Identification.....25
5.1 Existing Method Re-engineering.....26
5.2 Ad-hoc Construction.....30
5.2.1 Identifying Elements.....30
5.2.2 Designing Fragments.....34
6 Engineering of Situational Methods.....39
7 Application at the Example of efiport AG.....43
7.1 Introduction of the Scenario.....43
7.2 Identifying the Situation.....45
7.3 Engineering a Situational Method.....46
7.4 Above and Beyond Method Configuration.....50
8 Limitations.....54
9 Conclusion.....55
References.....56

Index of Figures

Figure 1: Relationship Among IS/IT Artifacts……………………………………… 6
Figure 2: Concept of a Method…………………………………………………… 8
Figure 3: Concept of a Method Fragment…………………………………………9
Figure 4: Adaptation Mechanisms………………………………………………. 10
Figure 5: Concept of a Situational Method……………………………………… 11
Figure 6: Scope and Design of the Meta Method…………………………………...15
Figure 7: Five Steps to Identify Situations……………………………………….16
Figure 8: Approaches for Method Fragment Construction………………………… 26
Figure 9: Situation Identification Matrix…………………………………………... 41
Figure 10: Methods Covering Similar Situations………………………………….. 42
Figure 11: Client-Server Architecture of CLM……………………………………. 44
Figure 12: Context Factor Parameter Values for efiport/CLM……………………..46
Figure 13: Method Configuration for the efiport Case……………………………...49
Figure 14: "New Product Process" on Top Level………………………………….. 50
Figure 15: The Activity "Definition in CLM" in BPMN…………………………...51
Figure 16: Adding a Service Layer………………………………………………… 53

Index of Tables

Table 1: Comparison of Existing Approaches for Service Identification…………..12
Table 2: Context Factors and Respective Parameter Values………………………. 21
Table 3: Effects of Pruning on the Number of Situations…………………………..24
Table 4: Activities of Method Fragments………………………………………….. 31
Table 5: Roles of Method Fragments……………………………………………….32
Table 6: Techniques of Method Fragments………………………………………... 33
Table 7: Results of Method Fragments…………………………………………….. 34
Table 8: Adaptation Mechanisms for Situational Methods…………………………39
Table 9: Fragments for Service Identification……………………………………... 48
Table 10: Excerpt of Service Candidate List………………………………………. 52

1 Introduction

Since the introduction of information technology (IT) in organizations over 50 years ago, the management and governance of IT infrastructures and architectures has become ever more important. Many so-called architecture paradigms came and went. Mainframes and terminals were replaced by client-server architectures, fat clients by thin clients, local data bases by integrated data warehouses and monolithic applications by enterprise service buses. The latest trend is service-oriented architectures (SOA).

In the age of globalization, IT architectures must be capable of supporting business processes that span several organizations and thus encompass multiple systems and even system landscapes. These systems are heterogeneous, characterized by both, intended and unintended redundancies; responsibilities are distributed among many people. Due to this complexity, a harmonization and detailed planning (which was envisaged by IT architects in earlier days) is no option nowadays. Furthermore, customer demands and markets are changing more rapidly which makes a fast adaptation of business processes essential for a company's success (Becker et al., 2004). The service-oriented paradigm and service-oriented architectures promise an increased agility and flexibility to support business processes.

The concept of a service is to encapsulate implementation details and to offer a coherent business service to the outside world through a well-defined interface. The scope of a service can range from comprehensive offerings (e.g. purchasing services) to fine-granular services (e.g. address verification) (Sehmi and Schwegler, 2006). On a technical level, software services can be reused and composed and thus support the agility of business processes (Cherbakov et al., 2005). Furthermore, SOAs provide for interoperability, i.e. functionalities can be implemented platform-independently. Legacy systems can be wrapped in a service and their functionalities can be offered through an interface using standards such as Web Service Description Language (WSDL). Finally, loose coupling of services guarantees a minimum of dependencies so that changes to services or failures have only little impact on the whole system.

Many authors agree that the identification of services is one of the most important parts in the service lifecycle (Kohlborn et al., 2009b; Börner and Goeken, 2009b). Any errors made during this identification phase can build up in the design and implementation phases and result in high costs due to necessary rework (Inaganti and Behara, 2007). Hence, it is of utmost importance to support the identification of services with adequate methods. Many existing methods have been analyzed by several authors (Kohlborn et al., 2009b; Börner and Goeken, 2009a; Birkmeier et al., 2009) but none of the former consider project specifics that occur in real-life settings satisfactorily. These one-size-fits-all approaches show little flexibility and thus offer inappropriate support for concrete projects. The importance of contingency factors in software development methods has been stated by many authors already (Avison and Wood-Harper, 1991; Davis, 1982) but has not yet been acknowledged in the field of service identification. In order to provide better methods for service identification, this paper proposes a meta method that helps to design such methods considering the circumstances of a project at hand. Eventually, this should lead to more suitable methods, better identified services and more successful SOA implementations.

The remainder of this paper is structured as follows: The discipline of design science is briefly introduced in section 2. Research methodologies used herein are explained to underpin scientific soundness. Section 3 describes the scope, goals and foundations of the meta method to be developed. For this purpose, it contains an overview of (situational) method engineering as well as a comparison of existing service identification methods. Section 4 and 5 form the core of this paper since they elaborate all necessary parts of a situation-specific method for service identification. On the one hand, context factors and SOA implementation goals are used to identify respective situations. On the other hand, method fragments, particularly for service identification, are designed. Section 6 shows how a situational configuration of these fragments can be performed. An exemplary application of the meta method is presented in section 7. Afterwards, section 8 reflects limitations of the research approach and the findings of this paper and discusses avenues for further research. Section 9 provides a conclusion.

2 Design Science and Research Methodologies

Goal of this paper is the development of a meta method for the configuration of situational methods for the identification of services. Thus, it can be assigned to the field of design science research (Hevner et al., 2004). A major objective is creating an artifact (the meta method) that gives methodological support for real-life projects. This implies a prescriptive character for the results presented herein. Models, methods and prototypes that are commonly results of design science research are called artifacts (Simon, 1996). These artifacts cannot be tested against reality because they are “to-be” models. For this reason, Hevner et al. (2004) demand more rigor in the development of these artifacts and postulate seven guidelines (p. 83). They are ought to serve as a common ground and a methodological foundation in design science research. In the following, these guidelines are discussed in respect to this paper.

1. **Design as an Artifact**: Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.

As outlined previously, this design science working paper will produce a meta method to support the identification of services in service-oriented architectures. Situational method engineering (SME) is used as a vehicle to configure this method and tailor it to suit a project at hand. Additionally, one instantiation in form of a concrete identification method will be presented.

2. **Problem Relevance**: The objective of design-science research is to develop technology-based solutions to important and relevant business problems.

Section 1 showed the increasing importance of the service-oriented paradigm. Since service identification is crucial for successful SOA implementations, the method developed in the following addresses relevant business problems. The application to a real-life project within a company demonstrates the relevance to businesses.

3. **Design Evaluation**: The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.

This paper aims at thoroughly designing a situational method and focuses on the design process. A comprehensive evaluation is clearly out of scope. However, the meta method will be applied to a real-life project. The application to one case can be seen as a first step towards a more comprehensive evaluation although some authors (Peffers et al., 2008) explicitly separate this demonstration from the evaluation of the artifact.

4. **Research Contributions**: Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.

The paper builds on an analysis of existing approaches for service identification. A major shortcoming of these approaches is a lack of adaptability to concrete projects. Through the utilization of ideas from the field of situational method engineering, the paper contributes to existing design methodologies.

5. **Research Rigor**: Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.

The focus of the work presented in this paper will be on the construction of the artifact, i.e. the meta method. Design decisions will therefore be presented as transparent and traceable as possible. Based on an exhaustive literature review and two case studies in the field of SOA development, design decisions are justified as good as possible. In their approach for theory-driven design research, Gehlert et al. (2009) compare constructs with previously tested hypotheses in order to justify design decisions. Since empirically tested hypotheses that could be compared to our constructs are scarce, they are substituted by the knowledge retrieved from literature and the experience from two case studies. A further evaluation would be desirable but is out of the scope of this paper.

6. **Design as a Search Process**: The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.

For every part of the research process adequate research methods have been applied. The need for a situational method and the identification of method fragments have been elaborated in an extensive literature research. Context factors that determine a situation in the sense of situational method engineering have been additionally derived from two case studies. The latter are particularly relevant for research in its “early, formative stages” (Benbasat et al., 1987; Myers, 2009) which applies to the field of SOA (Luthria and Rabhi, 2009; Stebbins, 2001).

7. **Communication of Research**: Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

In the first place, this working paper addresses a technology-oriented audience since it describes the application of a meta method to an exemplary case. Result of this application is a method to identify services. Although the management relies on good methods to implement a service-oriented architecture in an organization, the degree of abstraction used in this paper does not typically address management-oriented audiences.

However, the exemplary application to an information system and the subsequent identification of services is certainly interesting for the management and is thus presented adequately.

Many researchers in ISR have used the above guidelines since. Nonetheless, the latter are subject to controversy among scientists. Zelewski (2008), for example, criticizes that method guided work is sometimes wrongfully equated with scientific work just because it follows a set of guidelines (p. 174). Hevner et al. themselves “advise against mandatory or rote use of the guidelines” (p. 82). In so far, they should be interpreted as an orientation rather than a strict set of rules. However, many reviewers for both academic conferences and journals demand a discussion of these guidelines in design science papers. In case a guideline is not adhered to, authors are at least expected to argue why. Although following these guidelines may restrict creativity, it creates a common basis for design science research and provides a certain degree of comparability.

A number of authors describe process elements for the development of a design science research method similar to Hevner et al. (2004). Depending on their origin in different disciplines, some of them concentrate on the theoretical bases (Nunamaker et al., 1991; Walls et al., 1992) whereas others focus on applied problems (Archer, 1984; Eekels and Roozenburg, 1991). Peffers et al. (2008) present a synthesis of seven approaches in order to “provide a nominal process for the conduct of DS research” (p. 50) and to “provide researchers with a mental model or template for a structure for research outputs” (p. 50). Five of their six common design process elements can be mapped to Hevner et al.’s guidelines. One additional element is demonstration which shows “the use of the artifact to solve one or more instances of the problem” (Peffers et al., 2008) (p.55).

This paper will demonstrate how a meta method can be applied to one real-life project delivering a concrete instantiation of a situational method for service identification. Thus, *demonstration* is the core of efforts. Similarly, Rosemann and Vessey (2008) advocate an applicability check as an integral part of the research process. The example provided in this paper can be seen as such. An evaluation through multiple case studies, action research or quantifiable performance measures is out of the scope of this paper.

According to Gregor and Jones (2007), design science includes three phenomena of interest as depicted in Figure 1:

1. Instantiations (material artifacts): These artifacts are physically existent and can be a piece of hardware, software, an information system or “the series of physical actions (the processes or interventions) that lead to the existence of a piece of hardware, software, or an IS” (p. 321). Thus, the application of a method that consists of activities and leads to one of the above outcomes is considered a material artifact.

2. Theories (abstract artifacts): Constructs, methods and models (all of them not having a physical existence) belong to this type of artifact. Theories in ISR are sometimes referred to as models (Dubin, 1978). Gregor and Jones (2007) argue that “a theory can be about both the principles underlying the form of the design and also about the act of implementing the design in the real world” (p. 322).

3. Human subjective understanding of artifacts: Human beings can create theories and constructs by observation and inference from existing material artifacts. However, this is only one role human beings play in the relationship depicted in Figure 1. They can also use these abstract artifacts to build instantiations (e.g. software) in the real world.

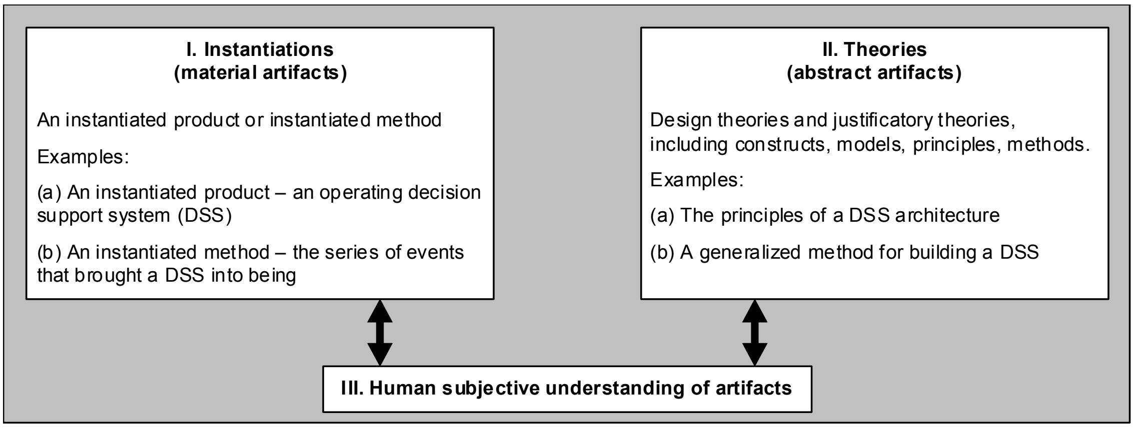

Figure 1. Relationship Among IS/IT Artifacts (following Gregor & Jones, 2007, p. 321)

In this paper, the application of a meta method for situation-specific configuration of service identification methods is presented. This meta method describes how a method has to be created. It utilizes context factors and SOA implementation goals (SOAIG) to identify situations which determine a configuration of method fragments. “Design theories can be about artifacts that are either **products** (…) or **methods**” (Gregor and Jones, 2007) (p. 322). Thus, in the notion of Gregor and Jones, the meta method presented herein constitutes a theory about methods. By applying this meta method to an exemplary case, an instantiation, i.e. a concrete (instantiated) method, is created. The latter is a material artifact since it prescribes the proceeding of service identification in one clearly defined case. Hence, this paper deals with both types of artifacts.

In order to develop a meta method for the construction of service identification methods in this paper, an exhaustive literature review in the fields of method engineering and service identification has been conducted. Hence, components that are used for the meta method – including context factors, parameter values, SOA implementation goals, fragments and their elements, etc. – are primarily based on desk research (also called library research), i.e. “research that is based mainly on the review of existing literature” (Palvia et al., 2003) (p. 291). Wherever possible, these findings are supported by using additional (empirical) research methods such as case studies.

Context factors, their parameter values and SOA implementation goals have been collected as comprehensible as possible. Whereas method elements have also been captured extensively, the fragments developed herein are exemplary in character. Based on the identified elements, a plethora of fragments could be developed by combining these elements. The fragments used in this paper are examples of reasonable combinations of elements. The former serve to demonstrate how fragments a) can be designed using the identified elements and b) can be combined to situational methods. Of course, the list of fragments is not complete and there are certainly more fragments that can be used for service identification methods.

3 Foundations of a New Method

This section will describe basic concepts of method engineering (ME) and situational method engineering because these are the foundations of the herein developed meta method. Since static methods can hardly suit any concrete project situation, configurability is a major aspect of this meta method. Thus, existing approaches will be scrutinized in this regard. Finally, the scope of the paper and the meta method will be outlined to give the reader a better orientation.

3.1 Basics of ME and SME

In information systems research (ISR), methods describe a systematic approach to the solving of problems. A method is commonly understood as a systematic way of acting that leads to verifiable results (Greiffenberg, 2003) or as a process built upon a set of rules, yielding solutions to a particular problem type (Teubner, 1999). Similarly, Becker et al. (2001) characterize a method as a tuple of a task and a set of rules: the former defines the problem type that is addressed by the method, and the latter describes prescribed actions aiming at an adequate solution to the problem (Bucher and Klesse, 2006).

In IS research a "method is an approach to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products" (Brinkkemper, 1996) (pp. 275–276). The use of methods is very important for the development of IS because they support the developers of IS by providing systematic development approaches (Avison and Fitzgerald, 1995). Due to this importance, the field of method engineering – which is described in the following – has emerged.

Since the early 1990s there have been efforts to guide the development of methods for information systems development (ISD) in order to guarantee a high quality of software applications. The task of method engineering is to give this guidance. According to Brinkkemper (1996), "method engineering is the engineering discipline to design, construct and adapt methods, techniques and tools for the development of information systems" (p. 276).

Besides the St. Gallen approach (Heym, 1993; Gutzwiller, 1994), concepts by Brinkkemper (1996), Brinkkemper et al. (1999) and Karlsson (2002) are the most frequently used approaches in ME. All of them identify activities, roles, results and techniques as important elements of a method (Goeken, 2006). Figure 2 exemplarily shows Gutzwiller's approach.

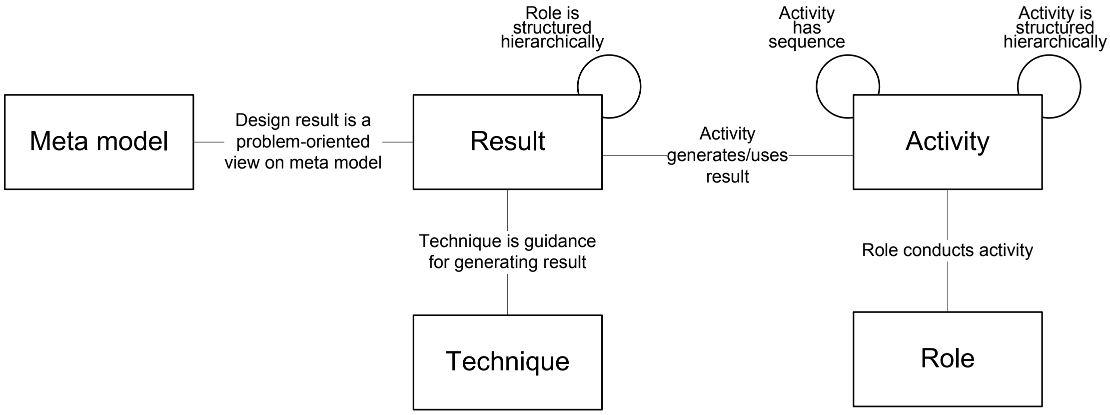

Figure 2. Concept of a Method (following Gutzwiller, 1994, p. 13)

Since these elements play an important role for the further proceeding, for the purpose of this paper they are defined as follows:

- *Activity*: Unit of execution that produces a result by facilitating respective techniques and notations.
- *Role*: Conducting one or more activities can be aggregated in one role. Hence, the role describes who carries out which activity to produce a result.
- *Result*: An artifact that is produced carrying out an activity by using certain techniques. A result can be used as an input for the next activity.
- *Technique*: A procedure that describes in detail how to perform an activity in order to support the generation of a result.

It is commonly accepted that no universal method constructed at time (t_1) can fit every conceivable situation in which it is applied in time (t_2) (Brooks, 1987; Fitzgerald et al., 2003). Actually, it is quite improbable that a rigid method developed from theory is applicable in a concrete setting without modification (Aydin, 2007). Therefore, the idea of situational method engineering emerged (see e.g. Harmsen et al. (1994)). The central aspect behind it is that a fixed method is not suitable for all situations that occur in reality. Thus, methods have to be adaptable to different kinds of situations. To provide for this adaptability, smaller parts of a method – so called method fragments – are created and can be composed at time (t_2) depending on the situation at hand (Ralyté and Rolland, 2001). According to Cossentino et al. (2006), the four elements explained previously constitute a method fragment as depicted in Figure 3.

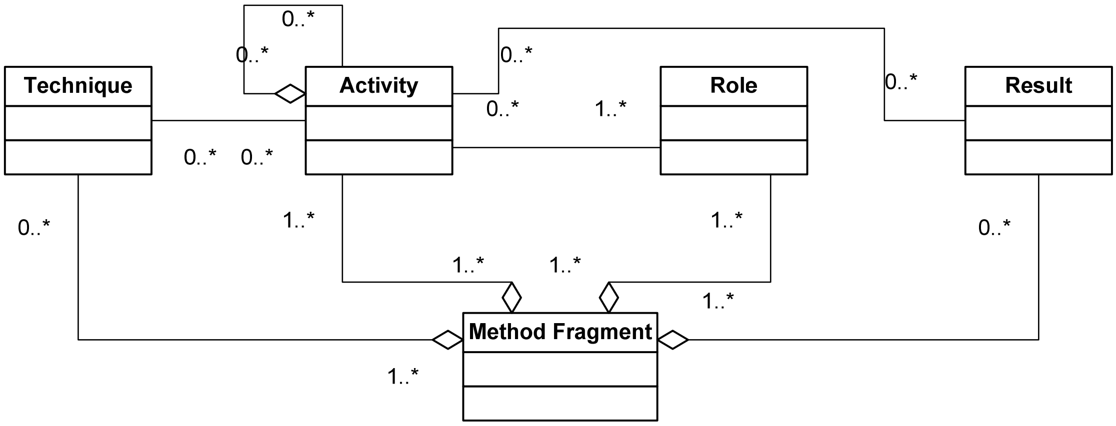

Figure 3. Concept of a Method Fragment

Unfortunately, the term method fragment is used inconsistently in literature (Sunyaev et al., 2008). Ågerfalk et al. (2007) define method fragments as "standardized building blocks based on a coherent part of a method" (p. 360). A situational method can be constructed by combining a number of method fragments.

Some authors differentiate method fragments into two types, namely product fragments and process fragments (Deneckère et al., 2009; Brinkkemper et al., 2001; Ralyté and Rolland, 2001b). A product fragment represents results such as diagrams, tables or models. A process fragment describes stages, activities and tasks that have to be performed to produce a product fragment (Cossentino et al., 2006) and is thus closely related to activities, roles and techniques that are used in this paper to build fragments. "Process fragments require product fragments and on the other hand process fragments produce product fragments" (Sunyaev et al., 2008) (p. 4). On top of this, Cronholma and Ågerfalk (1999) even introduce a concept fragment that is used in both other fragment types.

Based on Ågerfalk et al. (2007), for the purpose of this paper any reasonable combination of method elements representing a coherent part of a method shall be referred to as method fragment. A fragment can thus encompass multiple instances of its elements, i.e. more than one role, technique etc. However, in this paper most fragments will consist of only one technique because a subsumption of techniques in one fragment would decrease the options for configuration.

Another concept that can be found in literature is the method chunk. Ralyté and Rolland (2001) for instance use the term synonymously with method fragment. Others use the notion of method chunk to describe a combination of product fragments and process fragments (Rolland et al., 1998; Plihon et al., 1998; Mirbel and Ralyté, 2006). Since the discrimination between product fragments and process fragments is not important for the purpose of this paper, the term method chunk will not be used in the following. Figure 3 illustrates the concept of a method fragment as used herein.

According to Bucher et al. (2007), there are two adaptation mechanisms to engineer a situational method, namely *situational method configuration* and *situational method composition*. Situational method configuration follows the so called adaptive principle.

This means that a base method is created at design time (t_1) and configured in certain contexts at time (t_2). If situational method configuration is used, situational changes to a base method have to be foreseen and planned when a situational method is developed at time (t_1). In contrast, situational method composition provides for a spontaneous combination of method fragments (orchestration) that does not have to be foreseen at (t_1). There is no pre-defined base method that is adapted. Instead, method fragments are combined and aggregated as required at (t_2). Figure 4 illustrates the two approaches as proposed by Bucher et al. (2007).

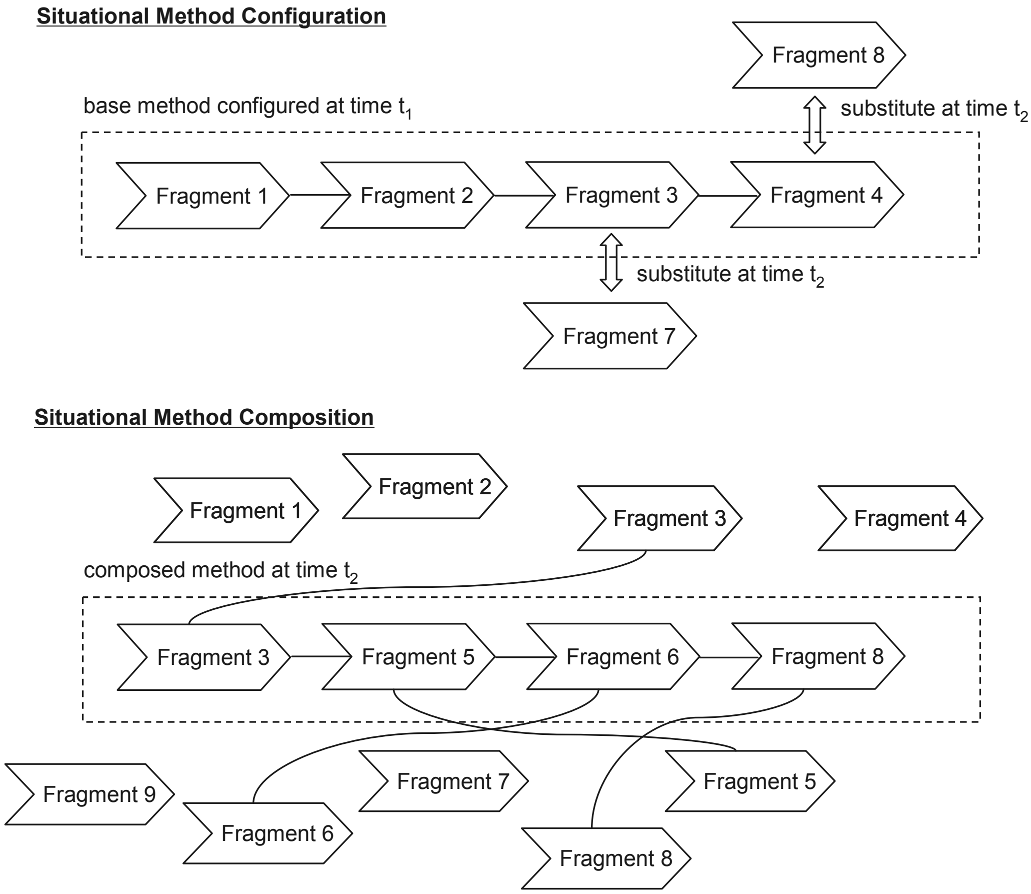

Figure 4. Adaptation Mechanisms

Adaptation mechanisms have been differentiated into configuration and composition so far. This classification depends on the existence of a base method. Another possibility to distinguish such mechanisms is the construction time of the method, i.e. the time when expert knowledge is utilized to configure or compose method fragments, respectively. Section 6 will provide a more detailed description of how to apply these mechanisms. Along with the two mechanisms by Bucher et al. (2007), a third adaptation mechanism by Börner (2010) will be discussed.

The previously described mechanisms are used to tailor methods to situations at hand. Many authors agree that – in order to describe these situations – characteristics of a project have to be defined (Brinkkemper, 1996; Punter and Lemmen, 1996; Rolland and Prakash, 1996; van Slooten and Hodes, 1996; Karlsson and Agerfalk, 2004). Still, according to Bucher et al. (2007) they do not explicitly say what constitutes a situation.

The definition of a situation is essential if preconfigured methods shall be assigned to situations. Therefore, Bucher et al. (2007) define *project type* and *context* as two determining factors that constitute a situation. A project type differs depending on the type of transformation of the underlying work system. Building a data warehouse from scratch compared to integrating existing databases is one example of two project types. In contrast, context factors are circumstances that influence a situation. Neither are they affected by the application of the method, nor can they be influenced by the project team. Each context factor has parameter values that will be explained in detail in section 4. For the purpose of this paper, Bucher et al.'s concept of a situation is adapted to the problem of service identification. Since their factor *project type* is rather generic, it is replaced by *SOA implementation goal*. Using these domain-specific goals enables a better description of situations in the field of service identification. The identification of situations in section 4 is thus based on context parameter value combinations (CPVC) and SOA implementation goals (SOAIG).

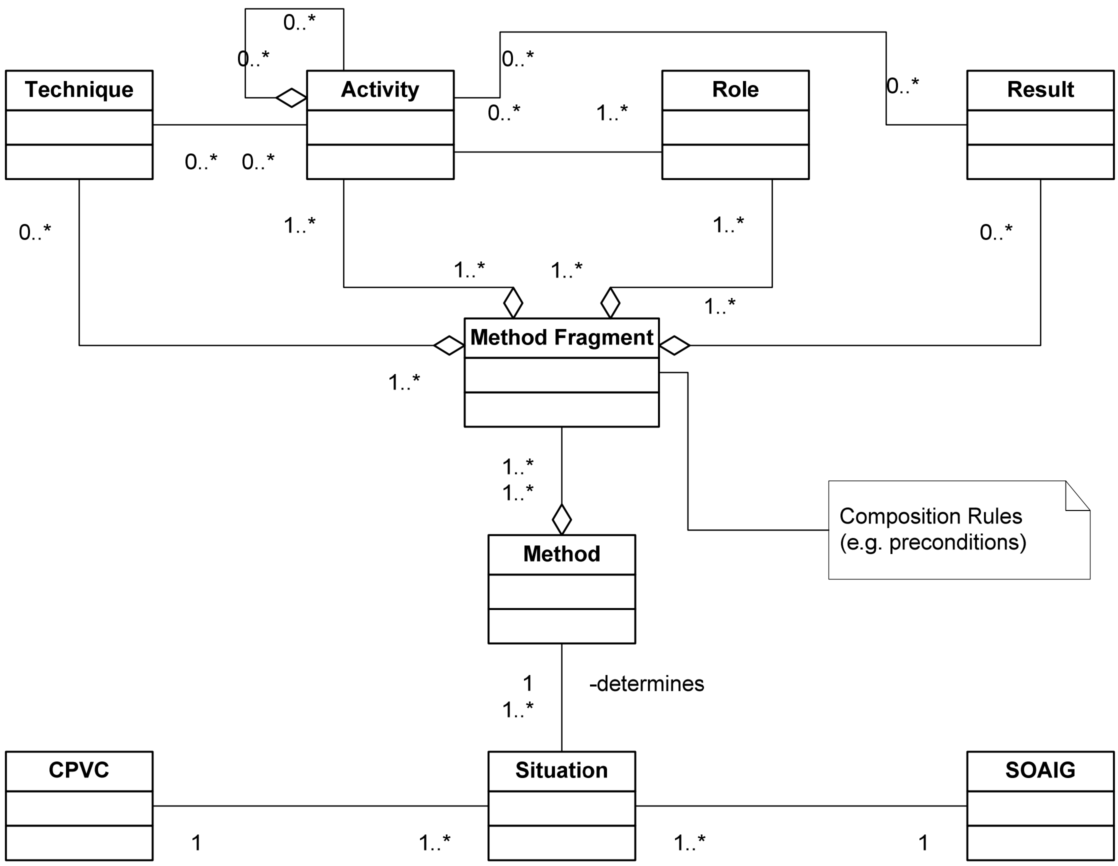

Figure 5. Concept of a Situational Method (Börner, 2010, p. 4)

A meta method for service identification methods as presented herein describes how situations are identified based on domain-specific context factors and SOAIGs (see Figure 5). Meta methods are further used to guide the process of selecting and integrat-ing different method fragments into consistent methods for the situation at hand (Hofstede and Verhoef, 1997). This can be supported by different adaptation mecha-nisms. Frequently, fragments are taken from different methods which is appropriate to combine the strengths of various approaches (Brinkkemper et al., 1999; Goldkuhl et al., 1998).

The fragment construction approach by Ralyté (2004) elaborated in section 5 goes even further by extracting single elements from existing fragments to create new fragments and improve situational methods.

This section discussed manifold aspects of ME and SME. As a result of the aforementioned, the herein created meta method for service identification will use fragments to support situational method engineering. These fragments will consist of the elements activity, result, role and technique as depicted in Figure 3 and will be developed in section 5. Adaptation mechanisms (Figure 4) will be used to tailor methods for service identification to concrete situations as described in section 6. The latter are determined by context factors and SOAIGs as shown in Figure 5. Both will be presented in more detail in section 4.

3.2 Existing Methods for Service Identification

In the service management lifecycle, service identification is an essential step that takes place in an early stage. Since any errors made in this step can affect the success of SOA implementations, it is necessary to guide service identification through adequate methods. As discussed in the previous section, configurability of these methods is very important to support projects in their respective context. The analysis and comparison of existing service identification methods is the basis for the development of a situational method conducted in this paper. Starting point for the comparison is a literature review by Börner and Goeken (2009a) who analyzed five service identification approaches. The criteria used in this analysis were mainly retrieved from Allen (2006), Erl (2004) and Josuttis (2008). Since the focus of that paper was slightly different, only selected criteria will be used herein. The latter form the core of the results presented in Table 1.

In the meanwhile, more approaches for service identification have been published. Two of them are added to the five taken from Börner and Goeken (2009a) and complement the analysis. The first one by Kohlborn, Korthaus, Chan and Rosemann (2009a) has been published in IEEE Transactions on Services Computing and aims to include both business services that support business processes and software services that fit the underlying IT architecture. The second article "BPM and SOA Handshake" by Inaganti & Behara (2007) focuses on the integration of Business Process Management (BPM) and SOA.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Activities	++	++	++	++	+	++	++
Roles	--	--	--	o	--	o	+
Results	+	++	+	++	+	++	+
Techniques	+	o	+	++	-	++	o
Flexibility (Sequence)	--	--	--	Iterative, fractal	partly iterative	--	--
Configurability	-	--	o	-	--	-	--
-- not existent - only implicitly o mentioned + defined/used ++ special focus							
Legend: (1) Klose et al. (2007), (2) Böhmann & Krcmar (2005), (3) Winkler (2007), (4) Arsanjani et al. (2008), (5) Kohlmann & Alt (2007), (6) Kohlborn et al. (2009), (7) Inaganti & Behara (2007)							

Table 1. Comparison of Existing Approaches for Service Identification

The opportunity to rearrange the sequence of activities will be referred to as flexibility in the following. This means that activities used in an approach do not have to be executed in a linear order but can be used iteratively and hence allow loops or a fractal approach. The notion of flexibility is thus closely related to the sequence of activities which is an important element for situational methods. In contrast, the criterion configurability shall express the possibility to choose adequate fragments that suit the situation at hand.

Looking at elements of fragments in the sense of situational method engineering, the description of activities is the focus of all compared approaches. However, it is noticeable that the granularity of what is called "activity" varies significantly among the approaches. Especially where different names for activities are used, this will make a consolidation into fragments even more difficult. Techniques are commonly described satisfactory. Particularly, Kohlborn et al. (2009a) and Arsanjani et al. (2008) divide their activities into detailed description of techniques that can or should be used. Accordingly, their results are well-defined and they thus provide quite a number of examples. Arsanjani et al. (2008) consider service identification itself as a technique being a "reusable part of a larger method" (p. 4). However, this definition differs significantly from the one used herein because the former encompasses a wide range of activities. Its scope is therefore much wider compared to the techniques described in this paper. All other identification approaches provide many intermediate results as well so that the collection of techniques and results found in this paper is of considerable length. Again, consolidation will be challenging since frequently many different names are used for one and the same purpose.

The most striking feature is the almost complete absence of roles in the compared approaches. Klose et al. (2007) talk of a business point of view and an IT perspective without mentioning people involved in SOA projects. Similarly, Arsanjani et al. (2008) employ a provider and a consumer view in their method and thus implicitly hint at the existence of different roles for certain activities or techniques. Only in Inaganti & Behara's (2007) approach, real-life project roles such as consultants or architects can be found. In some cases these roles are assigned to activities such as value chain analysis. There is a plethora of literature dealing with the importance of business and IT alignment and thus with communication aspects between business and IT departments (Kashanchi and Toland, 2008; Moody, 2003; Henderson and Venkatraman, 1993; Galliers et al., 1994). However, all discussed service identification approaches tend to underestimate the significance of properly assigning roles to respective activities and techniques.

The sequence of activities is usually linear. Commonly, the proposed methods are divided into three or four phases that are run through one after the other. The service identification takes place in a straight forward proceeding without loops or iterations. Kohlmann & Alt (2007) however, provide the opportunity of iterations at certain points of their method. Only Arsanjani et al. (2008) present an iterative, fractal procedure where "no rigid sequence is implied" and "fractal phases contain capabilities that can be leveraged as needed in different sequences" (p. 5). Apart from these exceptions, all approaches assume a linear sequence of activities.

Finally, the configurability of service identification methods is subject to comparison. Klose et al. (2007) look at a manufacturing company. The outsourcing and visibility potentials for stakeholders are the core of their approach. Techniques that consider concepts such as the line of visibility and the line of interaction (Shostack, 1982) are an integral part of their method and not substitutable. A configuration for less strategic service identification is not envisaged. Böhmann & Krcmar (2005) describe especially their modularization matrix in detail using an exemplary case from the IT service industry. It remains questionable though if their techniques and results can be transferred to other cases without configurations to their method. Most likely, some kind of configuration will be necessary to adapt the identification approach to other projects but this aspect is not discussed in their article. Winkler (2007) incorporates alternative steps in her service design activity and thus takes different preconditions into account. However, the scope of configuration is very limited since only minor design changes are possible.

Arsanjani et al. (2008) acknowledge the necessity of iterations in the process of identifying services. In many places they argue that results are intermediate, e.g. when they talk about service candidates that have to be evaluated in a further step and might have to be refined. However, they do not provide configurability based on circumstances that might occur in an SOA implementation project. Kohlmann & Alt (2007) successfully apply all parts of their methods in real-life projects at Swiss banks and thus do not discuss any configurations to their approach. Kohlborn et al. (2009a) do not explicitly deal with configuration. However, their detailed description hints clearly at the fact that an adaptation of their approach is necessary in some places to suit the needs brought about by different projects. The application of their method to an exemplary company shows even more explicitly that a configuration of certain elements is unavoidable. One example is interviews that have to be conducted if documents are not provided. The results of these interviews are a necessary input for further activities. In Inaganti & Behara's (2007) approach, a straight forward proceeding that does not take into account any project specifics is prescribed.

In the following, configurability of service identification methods will be considered as a crucial feature to adapt methods to a situation at hand and thus support the identification of services as good as possible. The necessity for research in this area is, for instance, expressed by Fitzgerald et al. (2003): "The absence of practice-based research in software development, and in method tailoring in particular, is surprising in an applied field" (p. 67).

The lack of configurability that was observed in existing approaches can be partly explained because the methods do not claim to be universal. For instance, since many of them have an industry-specific background, there is no need to consider certain activities and techniques that might play a role in other industries. Klose et al.'s (2007) approach for example stems from the manufacturing industry. Hence, it is unnecessary to analyze regulatory aspects that are important for financial service providers. Such an analysis is essential for banks and insurance companies that have to adhere to strict regulations when it comes to outsourcing services. Of course, a universal method has to support these kinds of circumstances by offering respective fragments and provide for this configurability. Since Klose et al.'s (2007) approach focuses on manufacturing, an industry-specific configuration is not provided because it is out of their scope.

The same is true for Böhmann & Krcmar's (2005) method that concentrates on IT service provision. Besides this lack of configurability concerning industry specifics there are other reasons that limit the configurability of existing approaches due to their scope and background. Kohlmann & Alt (2007), e.g., analyze the company's role within an inter-organizational value network. Thus, network models and reference processes are important and obligatory results in their approach, i.e. they are a fixed part of their proceeding. In contrast, for Winkler's (2007) method these results are completely irrelevant since her service identification is almost object-oriented and a configuration that includes strategic elements on an organizational level is therefore unsuitable. The two examples show that the SOA implementation goal is a crucial aspect for selecting certain fragments of a method and should be taken into consideration by a situational method.

3.3 Method vs. Meta Method

Due to the lack of configurability in so far existing approaches, this paper argues that situational method engineering can support methods for service identification that suit certain projects situations and are thus situation-specific. "Meta modeling is a widely established means for developing conceptual modeling methods" (Becker et al., 2006) (p. 85). In the same way, meta modeling can support the development of service identification methods. In order to understand the further proceeding, it is important to differentiate two levels of method design in this paper.

Firstly, the meta method encompasses the identification and description of context factors and SOA implementation goals. Together, combinations of both constitute a situation. Every instantiation of a method will rely on the situations that will be elaborated in section 4 since context factors and SOA implementation goals belong to the service identification domain. Thus, the latter are design elements of this meta method. Apart from situations, the meta method also includes descriptions of method fragments. These fragments and their constituting elements will be derived from existing methods for service identification. The transfer of ideas from situational method engineering to the domain of service identification and thus, a comprehensive description of this meta method is the primary goal of this paper.

Secondly, a method for the identification of services will be developed in an exemplary application scenario. For this purpose, a service identification project for a software called Campus & Learning Management System (CLM) in the company efiport AG will come under scrutiny. This example serves to demonstrate two aspects. Primarily, it shows the applicability of the meta method by configuring one service identification method suitable for the situation of CLM. Furthermore, this concrete method is used to identify services in the CLM case. Thus, it illustrates that an instantiation of the meta method can actually be used to identify services in an exemplary case. Although this identification of services is out of the original scope of this paper, it underpins the applicability and validity of the meta method described herein.

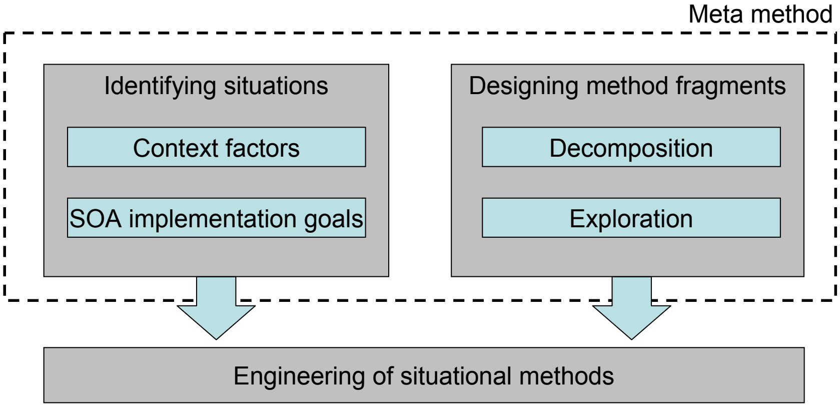

Figure 6. Scope and Design of the Meta Method

Figure 6 illustrates that the meta method encompasses the identification of situations as well as the design of method fragments. Both are necessary ingredients for a subsequent configuration of situational methods. A generally valid description of how to configure situational methods in any conceivable situation – and thus how to assign service identification methods to situations – is out of the scope of this paper and left to further research.

4 Identifying Situations

As depicted in Figure 5, a situational method is the product of the factors *method fragment* and *situation*, i.e. method fragments are chosen depending on the situation that is encountered in a project at hand. In the following, respective situations will be identified. According to section 3.1, the approach by Bucher et al. (2007) is modified to suit the problem of service identification (Börner, 2010). The combination of context parameter values and SOA implementation goals thus determines situations. Figure 7 illustrates the five necessary steps (a) to (e) to arrive at the situations.

These steps are introduced briefly herein (for a more detailed explanation see Börner (2010)):

(a) The context factors (contingency factors) that can influence the SOA implementation project at hand are defined.

(b) The context factors identified in (a) have certain parameter values (instantiations) that are defined in this second step.

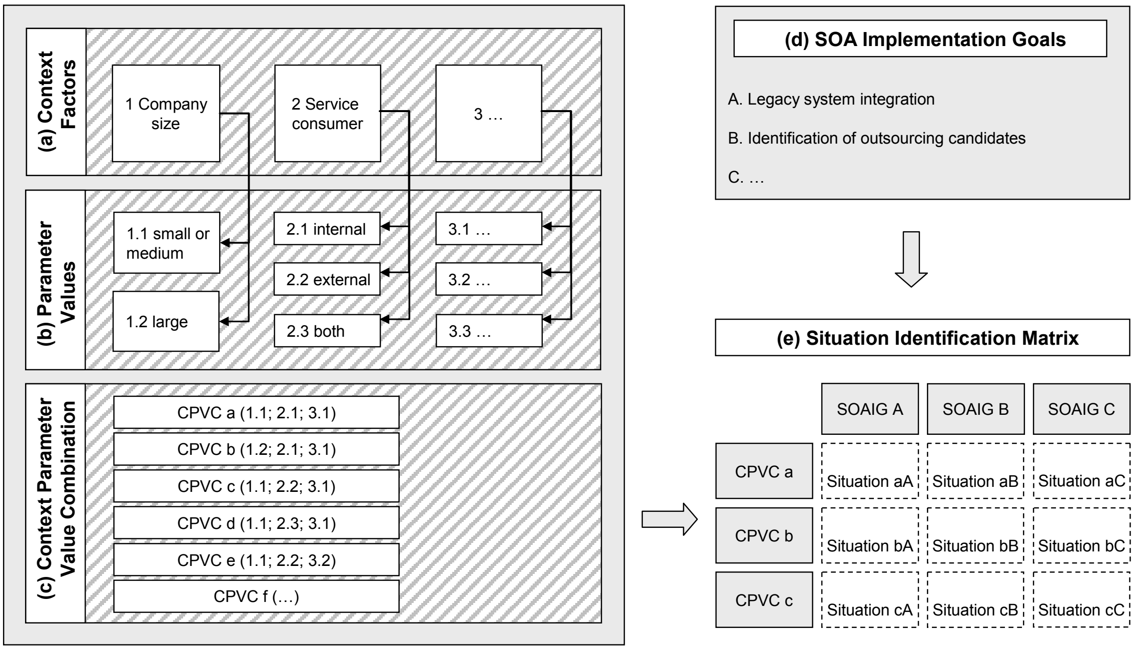

Figure 7. Five Steps to Identify Situations (following Börner, 2010, p. 5)

(c) All context parameter values are combined with one another. These context parameter value combinations (CPVC) serve as one input for the situation identification matrix.

(d) A number of possible SOA implementation goals are identified in this step. They are the second input for the situation identification matrix.

(e) The situation identification matrix illustrates all possible combinations of CPVCs and SOAIGs retrieved in steps (c) and (d).

Context factors, respective parameter values and SOA implementation goals that are unique to the service identification domain and therefore an important part of the meta method presented herein, will be introduced in the following subsections.

4.1 Context Factors

Based on related literature, Börner (2010) already presents five exemplary context factors to demonstrate the five steps of identifying a situation depicted in Figure 7. The search for relevant context factors has been advanced by two recently conducted case studies (Börner et al., 2011a; Börner et al., 2011b). However, the herein presented list of context factors cannot claim to be complete. Even though – by desk research and case studies – this list was compiled as comprehensive as possible, further research (e.g. more case studies) could reveal additional context factors. Moreover, interdependencies between context factors cannot be ruled out and should be investigated more closely.

In the following, identified context factors based on the previously mentioned sources are discussed. Furthermore, possible parameter values for each context factor are defined. As far as the derivation of these parameter values is concerned, two types can be differentiated. First, for so-called deductive parameter values, number and type of possible values can be deduced logically. Second, inductive parameter values have to be argued reasonably based on experience and literature. It is important to define parameter values in such a way that a change of values has a significant impact on the choice of fragments for service identification. In order to make the combination of context parameter values traceable for the reader, numbers will be assigned to all context factors and their parameter values.

It is a generally held belief that the ***company size*** is a considerable contextual factor in many kinds of software projects (Lai, 1994; DeLone, 1988; Raymond, 1985). The same holds true for service identification in SOA implementation projects. Company size usually has a geographic component, i.e. larger companies are more likely to act in different locations or even internationally. Subsequently, fragments producing different results might be necessary to fulfill certain demands in large companies. Due to its inductive character, assigning parameter values to this context factor is not trivial and will be discussed in the following.

Many criteria could be deployed for classifying company size in ISR. Usually, a continuum of large, middle and small companies is used (Sedera, 2008). The number of employees is a popular indicator for the size of a company and often used as the sole classification criteria. Therefore, companies with more than 250 employees are classified as large, those with more than 50 employees as medium and all others as small (Chau, 1995; Laukkanen et al., 2007). However, a number of other criteria could be employed to differentiate company size with respect to the research problem. For instance, Sedera (2008) argues for the number of users of an information system as a proxy for incorporating company size. This paper proposes a differentiation of small and medium-sized companies on the one hand and large companies such as multi-national enterprises on the other hand based on the number of employees following Welsh and White (1981). Consequently, small and medium-sized companies shall have less than 250 employees, large companies more than that. The geographic scope of operations is often closely linked to the company size and subsequently not considered as context factor itself.

Context factor 1: Company size

Parameter value 1.1: Small or medium-sized company

Parameter value 1.2: Large company

Depending on a company's strategy, services can be provided for different ***service consumers***, for example other divisions (internally), third parties (externally) or both. Services can be used to integrate heterogeneous enterprise applications (Erl, 2004; Arsanjani et al., 2008) and simplify the access to certain functionalities for staff, i.e. for internal consumers only. If this is known a priori, a number of activities and results such as the creation of an inter-organizational service map are not applicable in this situation. Moreover, there might be legal constraints that only apply if services are offered to third parties and thus demand an analysis of these regulations.

Varying degrees and forms of interaction with both customers and employees necessitate the use of different method fragments. In some cases employees are not directly involved in service delivery because the services are very fine-grained and fully automated. The coarser-grained services are, the higher is the possibility that they are only semi-automated or manual and subsequently interact with employees. Customer interaction is important when the composition of services by the end user is a primary goal. Customer integration into business processes is considered important in marketing literature (Kleinaltenkamp, 2000; Zeithaml and Bitner, 1996). An analysis of the “line of interaction” (Shostack, 1982) is essential in both consumer cases. The “line of visibility” is much more important if services are exposed to external customers (Klose et al., 2007). In general, a customer interaction can be obligatory in some places or can happen “on demand” if required (Leyer and Moormann, 2010). If customer interaction is a major issue for the identification of services in a situation at hand, related method fragments (like number 14) are crucial for a successful implementation. Swim lane diagrams that show interfaces to customers are one example for a result of such fragments.

For the purpose of this paper, service consumers will be divided into internal and external consumers. Additionally, the case of a service being offered to both is considered. It could be argued that the parameter value *internal and external consumers* encompasses *external consumers* because all additional demands have to be covered. However, if only *external consumers* are considered to use a service, there might be internal policies that do not have to be taken into account which makes certain fragments obsolete. The deductive derivation of parameter values covers all cases that could occur in real-life projects. Subsequently, the following parameter values will be used for identifying situations.

Context factor 2: Service consumers

Parameter value 2.1: Internal service consumers

Parameter value 2.2: External service consumers

Parameter value 2.3: Internal and external service consumers

The ***budget*** of a project plays an important role when it comes to choosing necessary steps and general proceeding of a method for service identification. Usually, a generous budget that allows for an extensive time frame provides the opportunity for a thorough and systematic application of identification methods. You would expect the utilization of many techniques in order to ensure a high quality of implemented services. A detailed analysis of all available strategic and technical documents would be typical in such circumstances.

The higher the project sponsor’s position in the company’s hierarchy, the more likely is a generous funding that allows for a proper analysis especially of strategic aspects and the inclusion of business processes. An initiation of a service identification project by the middle management often results in more pragmatic or technically-driven SOA implementations facilitating different fragments. Literature confirms that the budget has implications on the number of available staff, time pressure and the possibility to incorporate external help from consultants (Becker et al., 2007).

Some activities have to be carried out regardless of the budget. Still, there might be obligatory as well as optional techniques which support an activity. Consequently, an exhaustive use of techniques would only be chosen if financial resources are easily available. A differentiation based on absolute numbers measured in currency units is not feasible. Two parameter values will thus be used for the budget.

Context factor 3: Budget

Parameter value 3.1: Generous funding

Parameter value 3.2: Low budget

Skills and experience with both service-oriented architectures and Business Process Management significantly influence the proceeding of service identification. Different understandings of services based on different SOA concepts are a consequence of the former. Limited SOA experience on the side of employees often leads to a technical service understanding. The focus might thus be solely on developing reusable software services so that method fragments reflecting a more business-related SOA understanding and targeting the identification of business services are not considered. Project teams that are more familiar with the service-oriented paradigm are better able to combine technical aspects with a Business Process Management perspective in mind. Hence, software services that support business processes or at least sub processes can be the goal of their analysis. Their identification usually includes activities related to the analysis of process models and involves not only IT-related, but also business-related staff roles. If the service identification is limited to a rather technical point of view, the set of method fragments to be considered will therefore be a different one. Frequently, a hierarchy of services is the outcome of an analysis (Josuttis, 2008). Higher-level services that support business processes compose finer-grained software services that adhere to the technical preconditions of underlying systems and data. In order to achieve this more complex type of an SOA, a broader range of method fragments has to be used complementarily.

According to Inaganti and Behara (2007), a technical perspective – focusing the data level – leads to the application of bottom-up techniques and respective fragments. However, to avoid problems stemming from pure bottom-up approaches many authors advocate hybrid approaches that cover top-down techniques such as business process analysis as well as bottom-up methods such as asset analysis (Klose et al., 2007; Kohlmann and Alt, 2007). Others state that – in contrast to top-down modeling – (pure) bottom-up techniques typically lead to fine-grained services or functional modules (Erradi et al., 2007; Kohlborn et al., 2009b).

The configuration of situational methods and choice of fragments will be affected if certain roles cannot be occupied by accessible employees. Limited employee skills can necessitate external support by consultants. Although this enables the application of certain fragments, this option might be limited by the project budget and might therefore not be feasible. The availability of SOA and BPM skills is of deductive character and will be represented by parameter values as follows.

Context factor 4: Skills and experience

Parameter value 4.1: SOA skills available

Parameter value 4.2: BPM skills available

Parameter value 4.3: Both skills available

Parameter value 4.4: None available

Furthermore, the ***SOA maturity level*** a company has achieved is seen as a further influential factor on the delivery strategy of SOA (Terlouw et al., 2009).Thus, it plays an important role in the configuration of methods for service identification. However, it can be conjectured that the maturity related to the adoption of service-oriented concepts will increase over time, which will change the scope of the SOA understanding and consequently the way service identification is conducted. Many companies are still function-oriented (Vergidis et al., 2008) and employees only consider individual tasks with a limited scope in their daily work. In order to implement successful service-oriented architectures, they will have to get used to the concept of services.

SOA maturity models are used to classify the status of SOA implementations within a company. Broadly accepted models are the SOA Maturity Model (SOAMM) comprising five levels (Bachman, 2005) and the Service Integration Maturity Model (SIMM) differentiating between seven levels (Arsanjani and Holley, 2006). The latter will be used in the following to demonstrate how the maturity of SOA influences the way of services identification. The first three levels of SIMM focus on function orientation, whereas the latter four focus on service orientation (Arsanjani and Holley, 2006). Although they could be divided in more or different classes, for the purpose of this paper, SOA maturity levels will be divided into the following two classes. Advanced organizations with level 4 to 7 (simple services, composite services, virtualized services and dynamically reconfigurable services) are distinguished from less mature organizations (level 1 to 3). The former are likely to use more sophisticated and strategy-oriented fragments that, for instance, produce service maps (Kohlmann et al., 2010). The latter usually use more technically-oriented techniques and thus other fragments.

Context factor 5: SOA maturity level

Parameter value 5.1: SIMM level 1-3

Parameter value 5.2: SIMM level 4-7

On the one hand, ***compliance*** issues can arise from legal obligations and regulatory restrictions. These differ among countries and especially companies that operate in more than one country have to consider legal demands arising from that. In almost any country, confidentiality of customer data is protected by respective laws. Additionally, some industries such as banking or pharmaceuticals have to adhere to special regulations. Finally, regulations can arise from the fact that a company is listed on a stock exchange, i.e. it depends on its legal form. On the other hand, internal policies may require corresponding method fragments that address issues like service ownership (Kohlmann and Alt, 2009b).

A method fragment analyzing the customer interaction and the flow of (potentially) confidential data is often necessary. If services are exposed to customers, confidential treatment of sensitive customer data has to be guaranteed by a proper service design. The following inductive parameter values will be used in this paper.

Context factor 6: Compliance

Parameter value 6.1: Standard legal compliance

Parameter value 6.2: Special regulations due to industry, legal form or international operations

Parameter value 6.3: Internal policies

Another important context factor is the existence of a designated ***IT department*** and thus the degree of centralization of the IT infrastructure. In a small company that lacks an IT department, methods have to be adapted to accommodate for this circumstance. Larger organizations usually have such an IT division or are structured along the lines of business. On the one hand, a high degree of centralization or the existence of a central division supervising and governing IT implementation throughout a company usually leads to more transparency. Frequently, at least some information on applications and data is readily available. This can be used as input for method fragments. On the other hand, some fragments demand certain roles such as IT administrators or newly composed units consisting of business and IT employees (see also Anderson et al. (2005) and Börner et al. (2009)). In a small company that lacks an IT department, these method fragments are frequently not applicable. These parameter values are deductive since an IT department can either exist or not.

Context factor 7: IT department

Parameter value 7.1: Existent

Parameter value 7.2: Not existent

In a company specialized on one product only and operating in basically one region, an analysis of a service's reusability is trivial in most cases. The same analysis is much more complex when we look at companies with a wide range of products. An organization can, e.g. be structured by products (business lines), regions, functions or customer groups (Pitts and Lei, 2005; Schermerhorn, 1993). Even a multidimensional structure combining two or more dimensions of the above is not uncommon. The ***organizational structure*** can be an important factor when it comes to service identification. In complex structures, e.g. service ownerships among different domains have to be analyzed (Kohlmann and Alt, 2009b). Successful enterprises should be able to use their core competencies for many of their products (Prahalad and Hamel, 1990) across multiple organizational units. Thus, designing reusable services requires much more effort and different method fragments in complex structures. All of the above mentioned criteria could be used to differentiate organizations based on their structure. In the following, one-product companies are distinguished from multi-product companies.

Context factor 8: Organizational structure

Parameter value 8.1: One-product company

Parameter value 8.2: Multiple-product company

Table gives an overview of the herein used context factors and their respective parameter values. The parameter values of some context factors, namely service consumers, skills and experience, IT department and interaction are deductive, i.e. there cannot be any other parameter values than those given in the table. Furthermore, the categorization of a given situation is unambiguous in these cases. For instance, there can be no argument about the existence of an IT department. For all other context factors, namely company size, budget, SOA maturity level, compliance and organizational structure, the parameter values are derived inductively. Their number and concrete characteristics are based on literature and experience and have to be argued in every single case. For instance, the SOA maturity level could be divided in up to seven parameter values (one for each level). Even the decision for two parameter values still leaves the opportunity to distinguish between levels 1-3 and 4-7, levels 1-4 and 5-7, etc. The same holds true for the definition of what is a large company and what is not. Especially in the cases of inductive parameter values, more research is necessary.

Context Factor	Parameter Value
1 Company size	1.1 Small or medium-sized enterprise
	1.2 Large company
2 Service consumers	2.1 Internal consumers
	2.2 External consumers
	2.3 Internal and external consumers
3 Budget	3.1 Generous funding
	3.2 Low budget
4 Skills and experience	4.1 SOA skills available
	4.2 BPM skills available
	4.3 Both skills available
	4.4 None available
5 SOA maturity level	5.1 SIMM level 1-3
	5.2 SIMM level 4-7
6 Compliance	6.1 Standard legal compliance
	6.2 Special regulations
	6.3 Internal policies
7 IT department	7.1 Existent
	7.2 Not existent
8 Organizational structure	8.1 One-product company
	8.2 Multiple-product company

Table 2. Context Factors and Respective Parameter Values

Furthermore, some context factors are likely to have a higher impact on fragment selection than others. In practice, this leads to a different weight of context factors for the configuration of fragments. Interdependencies and relationships between context factors can also play an important role. A study by Börner, Goeken and Rabhi (2011b) made a first step towards an interaction model. Even a complete dominance of one context factor that assumes a certain parameter value over others is conceivable. Missing skills can for instance dominate a generous budget since no team member is able to use certain techniques that could complement basic fragments to achieve better results.

4.2 SOA Implementation Goals

The second constituting element of a situation is the SOA implementation goal. Depending on the purpose of an SOA implementation, the identification of services can necessitate the application of different method fragments. A number of such goals can be found in related literature and at least two of them could be confirmed by case studies. Analogous to the list of context factors, the list of identified SOAIGs might not be exhaustive. However, too many SOAIGs could obstruct an effective identification of situations since many goals might have the same or a similar impact on the selection of method fragments. In the following, goals and their influence on a situational method configuration will be discussed.

The ***integration of legacy systems*** is a frequently mentioned goal of SOA implementations (Erl, 2004; Heutschi, 2007; Arsanjani et al., 2008). Especially in medium-sized and large enterprises (Offermann, 2009), IT architectures have developed over years or even decades. In the absence of a central governing body, manifold isolated applications were developed and implemented which led to a plethora of problems. New functionalities and updates have to be made separately for each system which causes high maintenance costs. In some cases, it is difficult to find specialists who are able to administer, for instance, COBOL code. Often, the same functionalities are provided in different systems of many divisions. Due to their restricted function-oriented view, employees do not know about IT systems of other divisions. This redundancy causes high costs because of unnecessary licensing fees. Even more annoying for customers is inconsistent data stored in different databases. If, for example, a customer's address was correctly updated in the loan account system, his account statement will be send to his new address. At the same time, his credit card statement might be send to his old address because the data was not updated in the respective system.

Hence, integrating existing applications plays a major role in enterprises' IT architectures and is one of the reasons for SOA implementation projects. In this case, techniques such as asset analysis and results that illustrate dependencies of the existing IT infrastructure are crucial parts of the service identification. The knowledge of IT experts about technical interfaces is indispensable.

The ***identification of outsourcing candidates*** is another goal of SOA implementations (Beverungen et al., 2008). In this case, costs, performance and strategic relevance of services must be analyzed. On the one hand, based on a business process analyses, the exact scope of the outsourcing activity has to be defined. A strategic make-or-buy decision determines which parts of the process are performed within the organization and which parts shall be outsourced to service providers. Furthermore, the benchmarking of processes and services plays an important role. The management needs a perfect overview of an organization's process costs. These have to be compared to the price an external provider charges for the service in order to make an outsourcing decision. On the other hand, an outsourcing candidate needs clearly defined technical interfaces. Inputs and outputs of automated services provided by a third party have to be fixed in service level agreements. This is important, for instance, to enable straight-through-processing for fully automated processes. This kind of outsourcing considerations demands fragments that produce inter-organizational service maps and incorporate strategic aspects as well as detailed technical descriptions. An analysis of stakeholders is just as important (Mahfouz et al., 2010).

The ***agility and flexibility of business processes*** is a competitive advantage (Schelp and Aier, 2009) and strongly tied to the concept of SOA (Papazoglou, 2003; Heutschi, 2007). An alignment of business and IT is a necessary precondition to achieve this flexibility (Becker et al., 2009). Therefore, a strategic point of view, i.e. a business process perspective, has to be considered. An enterprise-wide governance of the IT infrastructure is necessary to provide for this agility. Hence, fragment results such as inter-organizational service maps or service ownership models (Kohlmann and Alt, 2009b) have to be used. If several service maps exist, an analysis across different divisions of a company is essential to build an enterprise-wide SOA (Kohlmann and Alt, 2009a).

In contrast to an enhanced flexibility on process services level, the ***standardization*** of basic services is meant to reduce redundancies in development and maintenance of IT and thus reduce costs significantly (Bieberstein et al., 2005; Legner and Heutschi, 2007). The goal is to improve efficiency by reusing a service in as many business processes as possible. In the best case, the options for customization of products are not affected by this standardization of services. A customer should not be limited in his choice of varieties. Instead, a faster processing through increased efficiency should lead to a higher customer satisfaction. Therefore, services that directly interact with customers should not be standardized. This makes techniques (and respective fragments) for the analysis of the line of interaction and the line of visibility indispensable.

A completely different perspective is taken by companies that aim at the ***provision of services for third parties***. The former specialize on a small part of a value chain concentrating on their core competencies. These so-called layer players (Heuskel, 1999) are able to generate economies of scale by providing services for many other companies typically – but not necessarily – belonging to the same industry sector. These cross company value networks become increasingly important e.g. in the banking industry (Kohlmann, 2007). Hence, the focus here is again on inter-organizational and strategic instruments. Services must be easily exposable to third parties, i.e. interfaces have to be well-defined and performance has to be readily measurable. The latter supports unambiguous service level agreements in terms of price and performance. Method fragments should thus concentrate on interaction, interface analysis and the strategic value of providing a service to third parties.

4.3 Restricting the Number of Situations

Following steps (a) to (e) depicted in Figure 7, all context factors, respective parameter values and SOA implementation goals are combined with each other. The eight context factors and their 20 parameter values used herein lead to 1,152 context parameter value combinations. Multiplied by five SOAIGs, this makes a total of 5,760 situations. The process illustrated in Figure 7 offers many occasions to reduce the number of situations to a manageable scope. The following description of five pruning opportunities demonstrates how effective pruning at different stages can be by using numbers from this service identification case (Table 3). The earlier a "pruning" of variables takes place, the stronger will be the impact on the total number of situations.

First Pruning: Relevant Context Factors. In section 4.1 relevant context factors were retrieved from literature and case studies. All these factors more or less influence the character of a project and thus the service identification method that should be used. In order to keep the number of situations manageable, it is important to decide which context factors really affect the choice of method fragments. If they were found to have no impact on a situation, discarding for example the last context factor (with two parameter values) would reduce the final number of situations by factor two from 5,760 to 2,880.

Second Pruning: Discriminating Parameter Values. All context factors identified in (a) have respective parameter values. The latter have to be defined in step (b). The discussion of the context factor *company size* already shows that two or three parameter values would be justifiable according to related literature. For the purpose of this paper, a differentiation into small and medium-sized enterprises and large companies was deemed appropriate. A finer-grained classification – in this case into three classes: small, medium and large – would have increased the complexity and should only be used if the choice of the method is significantly influenced by that. Extending the first pruning above, it could be assumed that internal policies (being a parameter value of *compliance*) do not influence the choice of fragments and BPM skills of employees (being a parameter value of *skills and experience*) are irrelevant as well. This reduction of parameter values would decrease the number of situations from 2,880 to 1,440.

Third Pruning: Context parameter value combinations. In step (c), all parameter values are combined to constitute one dimension of the Situation Identification Matrix (Figure 9). Before adding all CPVCs on the y-axis of the matrix, some of the combinations can be discarded. For instance, a company with a SIMM level of 5 or above combined with no SOA skills among its staff is not reasonable. Therefore, this context parameter value combination can be removed from the matrix. From case to case combinations that can be excluded have to be determined. Certainly, steps (a) and (b) have a much stronger influence on the total number of combinations than the exclusion of single combinations described here. It would take an enormous effort to check every CPVC in order to find and exclude unreasonable ones. It can be assumed that only the most contradictory combinations will be discarded. This may apply to a maximum of 20 CPVCs and reduce their number to 268. Subsequently, this leads to a total of 1,340 situations.

Fourth Pruning: Influential SOA Implementation Goals. Similar to the choice of context factors described in (a), relevant SOA implementation goals have been chosen. These goals form the second dimension of the matrix. Five goals have been discussed and were deemed influential in the previous section. If, for example, *standardization* demanded for the same fragments as the *integration of legacy systems*, this would lead to an omission of the former goal from the matrix. Since only four SOAIGs were left, the number of situations would be cut to 1,072.

Fifth Pruning: Situations in the Identification Matrix. The Situation Identification Matrix is used to illustrate all possible situations. Generally, all combinations of CPVCs and SOAIGs should lead to a situation. Still, there are some combinations that can be discarded. The goal of integrating legacy systems, for example, cannot reasonably be combined with any CPVC that includes characteristics of a small start-up company without an existing IT infrastructure. Hence, not all positions in the matrix are filled with one situation (Figure 9). Like in the third pruning, the number of unreasonable combinations that can be identified is rather small. There might be another 20 obvious combinations that can be discarded. The final number of situations would thus be reduced to 1,052.

	Original	After 1st Pruning	After 2nd Pruning	After 3rd Pruning	After 4th Pruning	After 5th Pruning
Number of Context Factors	8	7	7	7	7	7
Number of Parameter Values	20	18	16	16	16	16
Number of CVPC	1,152	576	288	268	268	268
Number of SOAIG	5	5	5	5	4	4
Number of Situations	**5,760**	**2,880**	**1,440**	**1,340**	**1,072**	**1,052**

Table 3. Effects of Pruning on the Number of Situations

The five options of pruning shown in Table 3 offer different starting points for restricting the number of possible situations in a meta method for the construction of service identification methods. Options three and five can be applied to the presented meta model immediately in order to eliminate unreasonable combinations. Examples have been shown in the respective paragraphs. Unfortunately, these options have the least impact on the total number of situations. The leverage for reducing this number is much bigger for options one, two and four. Since all context factors, their respective parameter values and SOA implementation goals have been identified as being relevant, these options are inapplicable at the current stage of research. However, if further research unveils that, e.g., two of the identified context factors have a negligible impact on the situation, the first pruning could be used. Analogously, if parameter values or SOAIGs are deemed irrelevant by future studies, the second and fourth pruning could be used, respectively.

The goal of a situation-specific configuration of methods for service identification is to create methods that suit the situation at hand. Most likely, it is impossible to cover all situations that can occur in real-life projects. In order to cover as many situations as possible, context factors, their parameter values and SOAIGs were collected comprehensively. Since all of them are deemed relevant, a meta method described herein will have to deal with the full set of 5,760 situations. Only few exceptions (representing unreasonable combinations as described in pruning steps three and five) can be removed from this total number. Further research should thus try to support or disprove the significant influence of variables on situations. These findings and the subsequent pruning will help to arrive at a manageable number of situations.

5 Method Fragments for Situation-Specific Methods for Service Identification

In order to design situation-specific methods for service identification, configurable method fragments that support this identification have to be provided. There are basically two possibilities to arrive at method fragments (Henderson-Sellers and Ralyté, 2010). On the one hand, these fragments can be extracted from existing methods by *method re-engineering*. Since most existing approaches do not use situational methods, it is unlikely that fragments are readily available. However, one might be able to extract implicitly existent fragments from these approaches. The inconsistent nomenclature concerning activities, roles, techniques and results is a major problem when fragments and elements are retrieved from existing methods. Identifying and consolidating these elements and assembling them to design fragments is the purpose of the following sub-sections. On the other hand, fragments can be designed from scratch in a so-called *ad-hoc construction*. The latter approach is only appropriate where no experience exists, i.e. where no fragments or elements can be retrieved from existing approaches. “New concepts, new ways of working, new ways of modelling must be taken into account and expressed” (Ralyté, 2004) (p. 5). This includes, e.g., new techniques or roles that can also be taken from literature but have not been used in service identification methods before.

As shown in section 3.2, a number of methods for service identification can be found in literature. Although they include many effective method elements and fragments, the lack of configurability is a major shortcoming. The following design of method fragments will consider both re-engineering of existing methods and ad-hoc construction. Ralyté (2004) identifies three ways to design method fragments from existing methods, namely product decomposition, process decomposition and exploration. These are complemented by the ad-hoc construction of methods as depicted in Figure 8.

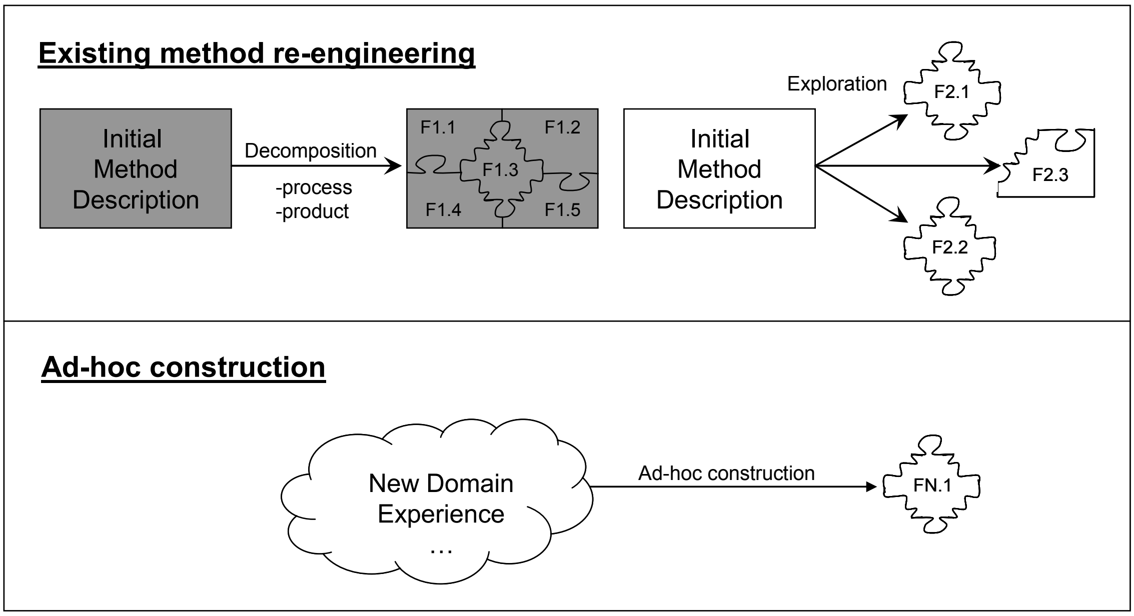

Figure 8. Approaches for Method Fragment Construction (following Ralyté, 2004, p. 4)

In the following, method re-engineering and ad-hoc construction will be used to identify method fragments. At first, process and product decomposition as well as exploration will be applied to existing methods because identifying already existing fragments is supposed to be easier than creating new ones from scratch. After identifying as many fragments as possible by re-engineering, ad-hoc construction is used to find additional fragments on the basis of the elements used in existing approaches complemented by newly introduced elements.

All method fragments presented herein are consistently structured along the following components:

Description: A meaningful description is important for the method engineer. It must enable him to judge if this fragment is applicable for his purposes. Therefore, it must explain what this fragment does and what it can be used for.
Input: Necessary inputs (such as documents) that might result from other fragments are listed.
Preconditions: Necessities that are not bound to certain results or documents (i.e. other than the aforementioned inputs) are shown.
Based on: The elements used for the construction of fragments were retrieved from related literature. References to this literature are given here.
Design: Detailed information on the process of re-engineering or ad-hoc construction, respectively, is presented.
Activity, Role, Technique and Result are used as explained in section 3.1.

The list of identified fragments does not claim to be complete and could be continued further. Along with the previously identified situations, method fragments designed in this section will be used in section 7 to engineer situational methods.

5.1 Existing Method Re-engineering

Most of the fragments that can be found in existing service identification approaches concentrate on the result of activities and are thus strongly product-driven. In these cases the process part – including roles and techniques – has to be conceptualized in order to obtain readily usable fragments. In cases where a fragment is identified by process model decomposition, the product part has to be elaborated since the processes are already available (Ralyté 2004) (p. 5). Exploration is used were product models can be put to different purposes. Oftentimes, relationships of their constituting elements are not explicitly shown, e.g. many activities are used in a method but it remains unclear a) which results of one activity are necessary inputs for another, b) if activities could be combined differently considering the first point and c) who performs which activities. The following fragments were subsequently extracted from existing approaches by decomposition or exploration, respectively.

Fragment 1: Overview of Existing Process Models
Description: If there are documented business processes, these should be used for further analyses to save time and money if possible. Documentation from previously conducted BPM projects is a valuable input. Usually, this fragment is used as a starting point for top-down approaches for service identification.
Input: Documented business processes
Preconditions: If no BPM projects have been conducted before, it is necessary to identify business processes before using this fragment.
Based on: Klose et al. (2007), Böhmann & Krcmar (2005), Kohlborn et al. (2009)
Design: In the first phase “preparation” of their approach, Klose et al. include the task “prepare existing process models” into their procedure model. Others also include hierarchical process models as results of their method. Since the authors describe activity, tech-

	niques and results, the fragment is identified by product decomposition. Only the role to perform the activity has to be added in order to complete the fragment.
Activity:	Preparation
Role:	Business analyst
Technique:	Document analysis of existing process models
Result:	Consolidated and complete set of hierarchical process models

Fragment 2: Asset Analysis

Description:	The existing IT infrastructure of an organization has to be analyzed from a technical perspective. Hardware, software and interfaces have to be identified and documented by the IT department. This may uncover redundancies of functionality, data and licenses. Apart from bottom-up approaches, this fragment is commonly used in hybrid approaches, too.
Input:	None.
Preconditions:	There has to be an existing IT infrastructure and of course, in virtually all cases there is some kind of IT already implemented. The only conceivable exemption is a start-up company that is built from scratch without any legacy systems. Thus, this fragment could be omitted in a service identification project within a start-up company.
Based on:	Böhmann & Krcmar (2005), Winkler (2007), Arsanjani et al. (2008), Kohlmann & Alt (2007), Kohlborn et al. (2009), Inaganti & Behara (2007)
Design:	A number of authors include an asset analysis in their approaches. In fact, it should be an integral part of almost any service identification method. In most approaches it is part of the preparation phase. Frequently, no roles are explicitly assigned and even techniques are not described very detailed. Thus, both elements had to be elaborated to describe the fragment comprehensively (product decomposition).
Activity:	Preparation
Role:	IT analyst, Enterprise Architecture Review Board
Technique:	Inventory analysis of physical hardware, i.e. count servers, clients, mainframes, routers, etc; interviews or surveys with managers and users to list all kinds of software that is used in different departments; find interfaces and their technical descriptions
Result:	Object model, network model, enterprise architecture model, UML diagrams, class diagrams, interface descriptions (e.g. XML code), asset documentation

Fragment 3: Identifying Stakeholders

Description:	First of all, knowing the stakeholders of a service is important from the business perspective. Creating value for stakeholders is one of the most important reasons to deploy a service-oriented architecture. Thus, identifying stakeholders should be a business-driven activity in the first place. The strategic dimension should be acknowledged by the management. In order to live up to the

	expectations, it is also important to consider internal stakeholders, i.e. to investigate who is using the system for which purpose.
Input:	Strategy documents.
Preconditions:	A communicated strategy that includes stakeholders must be given. The management's commitment to support this strategic task or at least a certain degree of awareness is necessary.
Based on:	Klose et al. (2007), Kohlborn et al. (2009), Inaganti & Behara (2007)
Design:	The previously mentioned approaches for service identification include a stakeholder analysis as part of their method. Neither more detailed techniques nor explicit roles are described which leads to a product decomposition. Thus, especially the procedural part of this fragment (i.e. "how" it should be done and by "whom") is elaborated.
Activity:	Preparation
Role:	Business analyst, management, consultant
Technique:	Analysis of strategy documents complemented by workshops and interviews with the management.
Result:	Stakeholder documentation including not only people but also their intentions and the way how they are involved in a business process.

Fragment 4: Identifying Activities

Description:	Employees who actually perform all necessary tasks in their every-day work have to be consulted to obtain a detailed insight into single steps of business processes. All activities that are necessary to fulfill a sub process have to be documented. This includes manual, semi-automated and automated steps. The documented business processes are broken down to work steps.
Input:	Documented business process(es)
Preconditions:	Employees involved in the operational work have to be available for workshops.
Based on:	Klose et al. (2007), Winkler (2007), Kohlborn et al. (2009), Inaganti & Behara (2007)
Design:	In her activity "identification of services", Winkler uses activity diagrams to break down business processes. Inaganti & Behara do the same without explicitly using this expression. Other authors use the term "decomposition of business processes" instead. The result "activity diagram" is exclusively mentioned in Winkler's approach and used in this fragment. Since techniques are commonly presented quite comprehensively, a process decomposition could be used for this fragment.
Activity:	Service identification
Role:	Business analyst, consultant, service owner
Technique:	In workshops with employees, business processes are decomposed to a fine-granular level so that activity diagrams can be developed by UML modeling.
Result:	Activity diagrams

Fragment 5: Compiling Service Candidates

Description:	Based on hierarchically decomposed process models which show all activities that have to be performed within a business process, a preliminary list of services – or so-called services candidates – is compiled. Every previously identified activity is checked whether it could be represented by a service or not. At this point, the business point of view, based on the work steps that have to be performed, is the focus while selecting service candidates. A re-design and a technical feasibility check will be conducted later.
Input:	Activity diagrams or BPMN diagrams
Preconditions:	None
Based on:	Klose et al. (2007), Winkler (2007), Kohlborn et al. (2009), Inaganti & Behara (2007)
Design:	Compiling a service candidate list is an activity that can be found in many approaches. Not all of them explicitly base this service list on activity diagrams as in Winkler's method. For a business process-oriented view, BPMN diagrams can also be used as input. This fragment uses existing activities and techniques for other purposes than originally intended which is typically an exploration.
Activity:	Service identification
Role:	Enterprise Architecture Review Board
Technique:	Identification of service candidates using work steps from activity diagrams or BPMN diagrams
Result:	Service candidate list

Fragment 6: Develop Key Performance Indicators

Description:	General statements of business goals are decomposed into actionable sub goals. The latter are further operationalized into key performance indicators (KPI) and metrics. This enables a measurement and monitoring of the success of an SOA implementation. One way to achieve an operationalization could be a Balanced Scorecard (Kaplan and Norton, 1992). At the same time, metrics allow for a benchmarking of services. This is important to support outsourcing decisions and the subsequent definition of service level agreements.
Input:	Business goals
Preconditions:	Based on previously identified business processes, the management must have defined business goals that should be achieved by these processes.
Based on:	Böhmann & Krcmar (2005), Arsanjani et al. (2008), Kohlborn et al. (2009)
Design:	The development of key performance indicators is recommended in the two mentioned service identification methods. Principles for the identification of KPIs can be taken from them but neither approach discusses necessary roles (product decomposition). Since KPIs are success measures, they clearly belong to the business domain and definition should be done by the management.

Activity:	Value analysis
Role:	Management
Technique:	Operationalization of business goals e.g. by using a Balanced Scorecard
Result:	List of key performance indicators and metrics

Fragment 7: Prioritization of Funding

Description:	By using both financial and non-financial measures, service candidates are evaluated. Based on their value for a company's business processes, funding decisions can be prioritized accordingly (Stewart and Chakraborty, 2010). Prioritization is used to decide which services (that have been confirmed by a feasibility check) are going to be implemented in which order.
Input:	Set of reusable services, feasibility check
Preconditions:	All desirable and feasible services have been identified before.
Based on:	Kohlborn et al. (2009), Arsanjani et al. (2008), Inaganti & Behara (2007)
Design:	In their approach to service identification, Kohlborn et al. use prioritization as separate phase that links the derivation of business services and software services. They provide three techniques that can be used to classify services. Funding decisions depend on the business value and should thus be made in the business department. Arsanjani et al. use prioritization in the identification phase to re-evaluate service candidates that had been discarded earlier. Well-described techniques and activities (and their sequence) provide for a process decomposition.
Activity:	Value analysis
Role:	Project manager
Technique:	Conduct value chain analysis or business network analysis; Analytical Hierarchy Process with financial/non-financial measures to classify business services (Hafeez, 2002)
Result:	Classified business services for funding decisions

5.2 Ad-hoc Construction

Not any fragment that is necessary for tailoring service identification methods based on the herein developed meta method can be re-engineered from existing methods. Thus, the second possibility to construct method fragments is ad-hoc construction, i.e. the design of new fragments from scratch (Ralyté, 2004). For this purpose, elements in existing methods are identified and further elements from related literature (not representing comprehensive service identification methods) are added. Afterwards, these elements are combined to newly designed fragments. The latter will complement the ones constructed by method re-engineering beforehand.

5.2.1 Identifying Elements

Method fragments are constituted by their elements as discussed in section 3.1. In the following, activities, roles, techniques and results found in different sources will be collected. The most important source is the literature on service identification approaches compared in section 3.2. Moreover, two case studies have been used to analyze respective elements. Börner et al. (2011b) analyze an SOA implementation in a small Australian data provider. Couzens (2009) looks at a service identification project in a large Australian insurance company. Some elements are taken from additional literature that deals e.g. with roles or techniques for service identification but does not present a comprehensive method (Alter et al., 2009; Börner et al., 2009; Kohlmann and Alt, 2009b; Weske, 2007). The following tables list all identified elements and their respective sources.

In Table 4, a number of ***activities*** as proposed by the analyzed approaches are consolidated.

Activities	**Encountered in**
Service identification, service analysis	(1), (3), (4), (6)
Service categorization, service design, module building	(1), (2), (3), (4)
Stakeholder definition	(1), (6)
Service exposure decisions	(4)
Component identification	(4)
Component specification	(4)
Preparation	(1), (5), (6)
Goal definition	(2)
Performance and design analysis	(2)
Develop SOA strategy	(6)
Establish business ontology	(6)
Value analysis, value chain analysis	(6), (7)
Legend: (1) Klose et al. (2007), (2) Böhmann & Krcmar (2005), (3) Winkler (2007), (4) Arsanjani et al. (2008), (5) Kohlmann & Alt (2007), (6) Kohlborn et al. (2009), (7) Inaganti & Behara (2007), (8) Börner et al. (2010), (9) Couzens (2009), (10) Alter et al. (2009), (11) Börner et al. (2009), (12) Kohlmann & Alt (2009b), (13) Weske (2007)	

Table 4. Activities of Method Fragments

The consolidation of activities taken from different methods is far from easy especially because the word activity is used for proceedings that differ significantly in scope. Some authors use activity for very detailed descriptions of how something has to be done. In the definition used herein this would rather be a technique and can thus be found in Table 6. Furthermore, one and the same activity might have different names in other approaches. Therefore, some lines contain more than one name for an activity because the names are taken from the respective approach. If these activities actually do the same thing, they are merged in one line of the table. Approaches such as (4) and (6) contain many activities. The fact that e.g. (5) can be found only once in Table 4 does not imply a smaller scope of this approach. The number of techniques e.g. exceeds the ones in (1) and (2) but the authors simply did not define activities that belong to these techniques. Subsequently, the sheer number of listings cannot be used to judge scope or completeness of an approach.

Table 5 gives an overview of ***roles*** that could be extracted from the various methods. As discussed previously, many approaches do not consider different roles in the process of service identification at all.

Roles	**Encountered in**
Business point of view, consumer view, business analyst, business team	(1), (2), (4), (7)
IT perspective, provider view, IT analyst, IT team	(1), (4), (7)
Project manager	(7), (8)
Service Design Unit (SDU)	(10)
Enterprise Architecture Review Board, Service Excellence Center	(7), (10)
Service owner	(6)
Management	(6)
Consultants	(7)
Architects	(7)
Legend: (1) Klose et al. (2007), (2) Böhmann & Krcmar (2005), (3) Winkler (2007), (4) Arsanjani et al. (2008), (5) Kohlmann & Alt (2007), (6) Kohlborn et al. (2009), (7) Inaganti & Behara (2007), (8) Börner et al. (2010), (9) Couzens (2009), (10) Alter et al. (2009), (11) Börner et al. (2009), (12) Kohlmann & Alt (2009b), (13) Weske (2007)	

Table 5. Roles of Method Fragments

Despite their importance, roles only occur in five of the seven compared approaches. Since e.g. business-IT alignment is an important issue for SOA implementations in general and service identification in particular, it is worth considering roles such as Service Design Units (SDU) as proposed by Alter et al. (2009). Such a role is absent in the compared approaches but can be a valuable element of method fragments for service identification. Analogous to the activities, different terms are used for similar or equal roles. For instance, Klose et al. (2007) use the notion of *IT perspective* whereas Arsanjani et al. (2008) coin the term *provider view*. Both have the same meaning since they refer to technical aspects of SOA implementations and respective project members. The outstanding role of a project manager that has been encountered in one case study justifies the inclusion of this role in our table.

Consolidating all ***techniques*** utilized in literature is difficult since most approaches offer a plenty of techniques with sometimes overlapping components and scopes. However, there seem to be some typical and wide-spread techniques that are common to many approaches such as decomposition of business processes and asset analysis (even though sometimes called differently). The former reflects a top-down approach; the latter is used for bottom-up analyses. Hybrid approaches might use both techniques. Some lines of Table 6 include nomenclatures of techniques that sound differently from one another but have actually the same meaning.

Techniques	Encountered in
Decomposition of business processes (with BPM)	(1), (3), (6), (7)
Asset analysis, documentation of IT systems, existing system analysis, portfolio analysis, application analysis	(2), (3), (4), (5), (6), (7)
Goal service modeling, definition of preconditions and goals	(2), (4)
Domain decomposition, deduction through process activities (naming)	(4), (5), (6)
Subsystem analysis (creating object models)	(4), (7)
Component specification (data, rules, services, conf. profile, variations)	(4), (7)
Service specification, coarse/detailed service definition	(4), (5), (7)
Scope definition of the analysis and framework development, scope for service enablement, scope of modularization, area selection	(1), (2), (5), (6)
Document analysis of existing process models	(1)
Stakeholder analysis	(1), (6)
Analysis of customer integration, visibility and takeover potential	(1), (2), (6)
IT feasibility analysis, bottom-up verification	(1), (5)
Identification of relations to other services, identification of interactions	(5), (6)
Operationalization by Balanced Scorecard approach	(2), (4)
Definition of service compositions, design modules	(1), (2)
Modularization	(2)
Verification of reutilization/services, analysis of invocation frequency	(3), (5), (6), (7)
Workshops	(4)
Proof-of-concept prototype	(4)
Brainstorming	(6)
Create activity diagram	(3)
Refine activity diagram (normalization)	(3)
Identification of service candidates	(3), (7)
Interviews	(5), (6)
Analytical Hierarchy Process (AHP)	(6)
Define financial/non-financial measures (for value analysis)	(6)
Use case modeling	(7)
Service-to-component mapping	(7)
Analyzing legal constraints	(9)
Governance questionnaire	(10)
Strategy questionnaire	(11)
BPMN modeling	(13)
Legend: (1) Klose et al. (2007), (2) Böhmann & Krcmar (2005), (3) Winkler (2007), (4) Arsanjani et al. (2008), (5) Kohlmann & Alt (2007), (6) Kohlborn et al. (2009), (7) Inaganti & Behara (2007), (8) Börner et al. (2010), (9) Couzens (2009), (10) Alter et al. (2009), (11) Börner et al. (2009), (12) Kohlmann & Alt (2009b), (13) Weske (2007)	

Table 6. Techniques of Method Fragments

It is noteworthy that the scope of the listed techniques can differ considerably. Using a governance questionnaire is a straight forward and unambiguous procedure with limited scope and little room for interpretation. In contrast, the decomposition of business processes is much more complex.

Similar to the significant number of techniques, there are many ***results*** presented as part of the methods. These results (Table 7) are outputs of respective activities and can be inputs for the next activity. Thus, they are a crucial link between method fragments. Those fragments that depend on a certain input can only be used if the previously chosen fragment has produced the result necessary to carry on with further activities. The results themselves are quite different in nature and reach from technical interface descriptions to comprehensive SOA strategy documents or network models on an inter-organizational level.

Results (inputs&outputs)	Encountered in
Service (sub) goals	(4)
Key performance indicators, metrics	(2), (4)
Use cases	(1), (4), (7)
Object models, network models, enterprise architecture models	(2), (4), (5), (6)
Framework, service architecture	(1), (2)
Hierarchical process models	(1), (2), (6)
Modeling/naming conventions, design principles	(1), (4), (5), (6)
Specified basic services	(1)
Interface description	(1)
Takeover and visibility evaluation sheet	(1), (6)
Modularization matrix	(2)
Reference processes	(5)
Activity diagrams	(3)
SOA strategy	(6)
Stakeholder documentation	(1), (6)
Organizational charts	(6)
Business performance models	(6)
Classified/prioritized business services	(6)
Process flow/swim lane diagrams, interaction model	(6), (7)
Service specification template	(7)
Business rules	(7)
Service-component mapping	(7)
Service candidate list, list of initial services, service inventory	(1), (3), (6), (7)
Ownership list	(12)
BPMN model	(13)
Legend: (1) Klose et al. (2007), (2) Böhmann & Krcmar (2005), (3) Winkler (2007), (4) Arsanjani et al. (2008), (5) Kohlmann & Alt (2007), (6) Kohlborn et al. (2009), (7) Inaganti & Behara (2007), (8) Börner et al. (2010), (9) Couzens (2009), (10) Alter et al. (2009) , (11) Börner et al. (2009), (12) Kohlmann & Alt (2009b), (13) Weske (2007)	

Table 7. Results of Method Fragments

Since many results are necessary inputs for further fragments, it is important to make them accessible for project teams. Thus, Gonzalez-Perez (2007) proposes a work product pool approach where all results are stored in a central repository. Subsequently, these results can be automatically found and used for other fragments.

5.2.2 Designing Fragments

In this section, a number of fragments for the identification of services based on the elements identified previously will be designed. Fragments can have different scopes, i.e. the number of elements used in each fragment can be different. Sometimes, techniques are closely interwoven or cannot be applied separately so that they are best used within one fragment. Fragment 8 is an example for a rather large and comprehensive fragment. Furthermore, a fragment might produce more than one result or might need multiple roles to conduct activities. Since the following fragments are designed from scratch, elements are based on more than one existing approach or source of literature, respectively. Thus, the fact that they are constructed “ad-hoc” will not be mentioned in the design paragraph of each fragment.

Fragment 8: IT Governance Analysis

Description:	An organization can have manifold demands when it comes to implementing new IT infrastructures. Using agile methods for example could be one imperative. Technical restrictions, programming language, interfaces or naming conventions can impose restrictions on IT projects. Hence, these IT governance issues can be covered by a questionnaire and are incorporated in this fragment. Employees of the business and the IT department jointly forming a so-called Service Design Unit (SDU) form an important role to successfully design services based on this questionnaire.
Input:	Information on IT governance, IT strategy, SOA strategy
Preconditions:	Derived from an organization’s strategy, the IT strategy must be defined and documented before this fragment can be used. Often, conventions and principles regarding the implementation and development of IT are not explicated in readily available documents. Therefore, it might be necessary to interview (IT) managers in order to retrieve more detailed information.
Based on:	Klose et al. (2007), Kohlborn et al. (2009), Arsanjani et al. (2008), Kohlmann and Alt (2007), Alter et al. (2009), Börner et al. (2009)
Design:	Since the scope of this fragment is quite wide, elements have been selected from different approaches. Thus, the fragment in its comprehensive form could not be derived by re-engineering from any method. Domain decomposition for instance could be found in three existing methods. Special roles and techniques such as an SDU or a governance questionnaire, respectively, have been taken from related literature that does not present a comprehensive method for service identification but deals with governance aspects in general.
Activity:	Service design
Role:	Service Design Unit
Technique:	Governance questionnaire, domain decomposition, naming
Result:	Naming conventions, service ownership list, modeling conventions, design principles

Fragment 9: Identifying Business Processes

Description:	A company’s main business processes have to be identified. This is a rather strategic task. Based on stakeholder documentation, the management must decide which business processes are central for the company. These processes have to me modeled in an end-to-end perspective, i.e. the customer (internal or external) is both trigger of the process and receiver of its outcome.

Input: Stakeholder documentation.
Preconditions: Knowledge and experience of the management and staff from the business department must be captured. Thus, the availability of management and staff for interviews is essential.
Based on: Klose et al. (2007), Winkler (2007), Kohlborn et al. (2009), Inaganti & Behara (2007)
Design: Many methods for service identification take existing business process models for granted. However, in many cases there are no documented business processes at all. At least one of the major business processes has to be identified by the management as a basis for further service identification fragments to be used. Ideas from several existing methods are thus combined to design this fragment.
Activity: Preparation
Role: Business analyst, consultant, management
Technique: Interview management to identify business processes, workshops with staff
Result: At least one documented business process.

Fragment 10: Developing Reutilization Scenarios

Description: Based on the service candidate list and in absence of multiple use cases, possible scenarios for the reutilization of these service candidates are developed. This is an unstructured and creative process that can be supported by brainstorming techniques and groupware that enables collaborative approaches to develop scenarios. These ideas for new or re-designed business processes have to be developed in the business department. The development of reutilization scenarios underpins the reusability potential of services.
Input: Service candidate list
Preconditions: This fragment is frequently used if no comparison of existing use cases is possible, i.e. if there is only one or even no use case at all. However, it can be used even if multiple use cases and business processes are well-documented.
Based on: Inaganti & Behara (2007)
Design: Whereas an analysis of existing processes concerning the reusability of services is part of several approaches, development of reutilization scenarios is not mentioned in any method yet. Inaganti and Behara at least mention use cases. The technique of use case modeling is well-suited for this fragment.
Activity: Service design
Role: Business analyst
Technique: Brainstorming, use case modeling
Result: Documentation of reutilization scenarios

Fragment 11: Analyzing and Improving Reusability of Services (or Candidates)

Description: Frequently, a first round of analysis delivers a considerable number of potential services, so-called service candidates. A first step to identify reusable services is to look at how often services are

	invoked. Secondly, a more business process-oriented view can be used to increase this frequency by redesigning services. The former can be done by the IT department whereas the latter needs support from a Service Design Unit. Of course, this applies to both services and service candidates. The Service Excellence Center has an overview of existing services and can help to assess the necessity of new services.
Input:	List of service candidates or existing services, respectively, documentation of use cases and reutilization scenarios, documentation of existing assets
Preconditions:	Analyzing the frequency of service invocation is only possible if there are services in place already. Furthermore, there must be a measurement system in place that provides data for this analysis. A Service Design Unit must be installed in the organization.
Based on:	Winkler (2007), Kohlmann & Alt (2007), Kohlborn et al. (2009), Inaganti & Behara (2007), Alter et al. (2009), Börner et al. (2009)
Design:	In many approaches, the importance of reusability is reflected by an analysis of service reutilization. Thus, techniques such as an analysis of the invocation frequency can be found in literature. None of the existing approaches combines the analysis with an improvement of the service design based on use cases or reutilization scenarios. Involving an SDU as proposed in Alter et al. can increase the reusability of services significantly. Hence, a new fragment including this role is created from scratch.
Activity:	Service design
Role:	Service Design Unit, Service Excellence Center
Technique:	Verification of reutilization/services, analysis of invocation frequency (counting the number of invocations), create reutilization scenarios
Result:	Set of reusable services

Fragment 12: Feasibility Check

Description:	The service candidates from the business departments are checked whether they can be implemented on the existing technical platform. This fragment is usually used as an alternative to fragment 11 when a Service Design Unit is not existent. Reutilization scenarios from fragment 10 can be checked for feasibility by the IT department.
Input:	List of service candidates, documentation of existing assets, documentation of use cases or reutilization scenarios
Preconditions:	An asset analysis must have been conducted beforehand.
Based on:	Klose et al. (2007), Kohlmann & Alt (2007), Kohlborn et al. (2009), Arsanjani et al. (2008)
Design:	In Klose et al.’s method this step is performed in the service analysis phase by the IT department. Arsanjani et al. include a feasibility check without mentioning a role. However, they are the only ones including a technique in their description, namely a proof-of-concept prototype. Kohlmann & Alt’s remark that a business view is not sufficient implicitly hints at the role of the IT

	department. In Kohlborn et al.'s method it is part of the detailing phase. Since no existing method encompasses this fragment comprehensively it is developed by ad-hoc construction.
Activity:	Service design
Role:	IT analyst
Technique:	Analyze existing (documentation on) IT infrastructure to check technical feasibility of service candidates, proof-of-concept prototype
Result:	Set of reusable services

Fragment 13: Goal Service Modeling

Description:	Goal service modeling (GSM) is important to link a company's business strategy to technical services. Thus, it supports the alignment of strategy and IT. It can be used for "monitoring and managing the scope of further business-process modeling and existing asset analysis" (Arsanjani et al. 2008) (p. 384). It ties services e.g. to key performance indicators. The "middle-out view" necessitates both business and IT know-how.
Input:	Business processes, strategy documents
Preconditions:	Strategy awareness on the management's side is necessary. Abstract strategic goals like a company's vision have to be operationalized in order to have more concrete, achievable and timely fixed business goals.
Based on:	Böhmann & Krcmar (2005), Arsanjani et al. (2008), Alter et al. (2009)
Design:	This fragment is used in two existing approaches. Böhmann & Krcmar merely mention GSM without any further description. Arsanjani et al. incorporate GSM in their "identification" phase. They describe techniques in detail and even provide a table from an exemplary case as a result of this fragment. Application of techniques is not linked to roles. In order to achieve an IT-business-alignment a Service Design Unit jointly with the management should be responsible for GSM. This role is adapted from Alter et al.
Activity:	Service identification
Role:	Management, Service Design Unit
Technique:	Scorecard approach or strategy maps in workshops or discussions
Result:	Service candidate list

Fragment 14: Strategic Stakeholder Integration

Description:	By considering all stakeholders, such as customers, employees (users) and suppliers, business processes are analyzed concerning the integration of the former. The results show the implications on the respective services. Customer and employee interaction ("line of interaction") play an important role. An interaction model as proposed by Lovelock and Wirtz (2007) is used. Additionally, the "line of visibility" has to be considered which is a rather strategic decision. On the one hand, the business side or management has to decide what degree of stakeholder integration

is desirable. On the other hand, the IT department must assess if interaction and visibility can be implemented in services accordingly.

Input: Stakeholder documentation

Preconditions: This fragment is only useful if stakeholders other than the end consumer (for instance suppliers) should be integrated into the process. A strategic view on service orientation and management awareness is thus necessary.

Based on: Klose et al. (2007), Böhmann & Krcmar (2005), Kohlborn et al. (2009), Alter et al. (2009)

Design: Three approaches deal with stakeholder integration in a very similar manner. Kohlborn et al. directly refer to Klose et al.'s method. None of the approaches proposes roles for this activity. Since desirability and feasibility (strategically and technically) have to be evaluated, an SDU as proposed by Alter et al. is used for this fragment.

Activity: Service design

Role: Service Design Unit, management

Technique: Workshops to determine customer integration, visibility and takeover potential from both a business and an IT point of view; interaction model

Result: Swim lane diagrams, Evaluation sheet (takeover/visibility)

Fragment 15: Composition of Basic Services

Description: In a mature SOA organization, the service directory of existing services tends to grow. The business department should check on a regular basis if existing services can be (re-)composed to support business processes or even create new ones. Since the services that are to be composed exist already, an involvement of the IT department is not necessary. Particularly with SIMM levels of 4 or above this fragment should be used. By analyzing a directory of services and using creative techniques, new compositions of services can be developed.

Input: List of existing services

Preconditions: A company must have achieved a SIMM level that already provides for existing basic services that can be aggregated to composed or even process services.

Based on: Arsanjani et al. (2008), Klose et al. (2007), Kohlborn et al. (2009)

Design: Although composition is mentioned in several approaches (Arsanjani et al. use it in their specification phase, Kohlborn et al. in their identification phase), neither techniques nor roles are clearly assigned. Thus, this fragment is further elaborated by adding the missing elements ad-hoc.

Activity: Service design

Role: Business analyst, service owner

Technique: Analysis of service directory, workshop, brainstorming

Result: Business-driven composed or process services

Fragment 16: Breaking Down Business Processes

Description: According to fragment 4, employees who actually perform all necessary tasks in their every-day work have to be consulted to obtain a detailed insight into single steps of business processes. All activities that are necessary to fulfill a sub process have to be documented. This includes manual, semi-automated and automated steps. The documented business processes are broken down to work steps. This fragment differs from fragment 4 as it uses BPMN instead of UML (see *technique* and *result*).

Input: Documented business process(es)

Preconditions: Employees involved in the operational work have to be available for workshops.

Based on: Klose et al. (2007), Winkler (2007), Kohlborn et al. (2009), Inaganti & Behara (2007), Weske (2007)

Design: Many authors describe a break down of business processes using different notions. Whereas Winkler or Inaganti & Behara use activity diagrams, others describe a decomposition of business processes. The idea of BPMN models is advocated by many authors, e.g. Weske. Hence, the fragment is created from scratch using multiple sources.

Activity: Service identification

Role: Business analyst, consultant, service owner

Technique: In workshops with employees, business processes are decomposed to a fine-granular level so that BPMN diagrams can be developed by BPMN modeling.

Result: BPMN diagrams

6 Engineering of Situational Methods

So far, situations and method fragments that are both part of the meta method have been elaborated. The next step is engineering situational methods based on these ingredients, i.e. method fragments have to be combined to methods that are assigned to situations. As discussed earlier this is not part of the meta method presented herein (see Figure 6) but still important to arrive at a situational method for service identification, eventually.

Many authors agree that, in order to engineer situational methods, method fragments should be stored in a method base from where they can be retrieved (Saeki et al., 1993; Harmsen, 1997; Rolland et al., 1998; Deneckère et al., 2008). Several adaptation mechanisms can be used for this purpose. Section 3.1 briefly introduced two such mechanisms based on a literature review by Bucher et al. (2007) which will be complemented by a third one by Börner (2010) and discussed in more detail in the following. Table 8 provides an overview of the adaptation mechanisms and their characteristics.

	(1) Configuration according to Bucher et al. (2007)	**(2) Composition according to Bucher et al. (2007)**	**(3) Composition according to Börner (2010)**
Pre-configured base method	Yes	No	No
Configuration time	t_1	t_2	t_1

Table 8. Adaptation Mechanisms for Situational Methods

Method fragment configuration according to Bucher et al. (2007): This approach proposes a base method as starting point for further configurations. Fragments of the base method can be removed or substituted by others. New fragments can be added. Configuration packages and templates describe how the base method has to be altered according to certain situations (Karlsson and Agerfalk, 2007). These alterations to the base method have to be defined at time (t_1), i.e. ahead of the project. At project start (t_2); only the respective situation has to be identified to choose the right configuration template. Following Karlsson and Agerfalk (2004) the configuration process is divided into three phases:

- Defining configuration packages: A configuration package allows for a configuration of the base method with regard to one context factor, for example, company size.
- Combining configuration packages in configuration templates: Configuration templates represent the comprehensive configuration of a base method by considering many context factors at the same time. Hence, a combination of configuration packages leads to a configuration template that represents a method (comparable to those depicted in Figure 9) that fits to one or more situations.
- Selecting a configuration template that is adequate for the project situation: In a concrete project, the situation has to be identified and the suitable configuration of the base method has to be determined at time (t_2).

A disadvantage of this mechanism is that all changes to the base method have to be foreseen at time (t_1) when all methods (i.e. packages and templates) are designed. There is no room for flexibility at time (t_2) when the project is about to commence. At that time, only the situation is identified and a respective method selected. In order to use this mechanism, expert knowledge about which methods fragments suit certain situations must be available at time (t_1). At project time (t_2) it is not necessary to know which fragments are required because the identification of the situation at hand leads to an assignment of packages and templates that prescribe the application of respective fragments.

Method fragment composition according to Bucher et al. (2007): In the case of method fragment composition no base method is constructed in advance. In order to select and orchestrate fragments according to the needs of a software development project, method fragments can be combined and aggregated at the outset of a project at time (t_2). A free combination thus produces new constructional results. Of course, dependencies and relationships such as necessary preconditions (for example inputs and roles) have to be considered. Following Brinkkemper (1996) and Harmsen (1997), this process of method fragment composition can also be divided into three steps:

- Identifying situational characteristics: When the projects starts, context factors are analyzed in order to tailor the method to the project characteristics.
- Decomposing generic methods into method fragments: Generic methods are broken down into fragments. Dependencies and interrelationships of these fragments are described in construction principles taking the previously identified situational characteristics into account.

- Composing method fragments into a situational method: By applying previously defined construction principles that consider the relationships of fragments, method fragments are composed to a situational method. For this purpose, a set of construction principles would have to be specified at an earlier time to provide for a consistent composition.

This approach is widely used in method engineering literature (Brinkkemper et al., 1998; Baumöl, 2005; van Slooten and Hodes, 1996). Generally, by defining construction principles at time (t_1), the composition of fragments at project time (t_2) is restricted. However, this mechanism offers a higher degree of flexibility because the situational method is composed at (t_2) when project characteristics are known. Of course, expert knowledge about how to compose methods successfully in a given domain (for instance methods for service identification) has to be available at project time (t_2).

Method fragment composition according to Börner (2010): This third approach does not utilize a base method and is therefore also called “composition”. In contrast to the second mechanism, the composition of method fragments takes place at time (t_1). Keeping in mind possible situation characteristics, methods for all conceivable situations are engineered from method fragments at time (t_1). These methods are assigned to certain situations that are identified at project time (t_2) according to Figure 9. Again, three steps illustrate the proceeding of the approach.

- Composing methods from fragments: For all conceivable situations, methods are composed out of a pool of method fragments at time (t_1). This is done by experts who know the application domain.
- Identifying situations: At project time (t_2), the project manager identifies the situation at hand based on the context factors and an SOA implementation goal.
- Assigning methods to situations: After the situation is identified, a suitable method that has been composed at time (t_1) can be assigned to the situation according to the Situation Identification Matrix in Figure 9.

Subsequently, every situation is mapped to a method that is tailored to the needs of the respective situation. Figure 9 illustrates this mapping for service identification methods using context parameter value combinations and SOA implementation goals on the y- and x-axis, respectively.

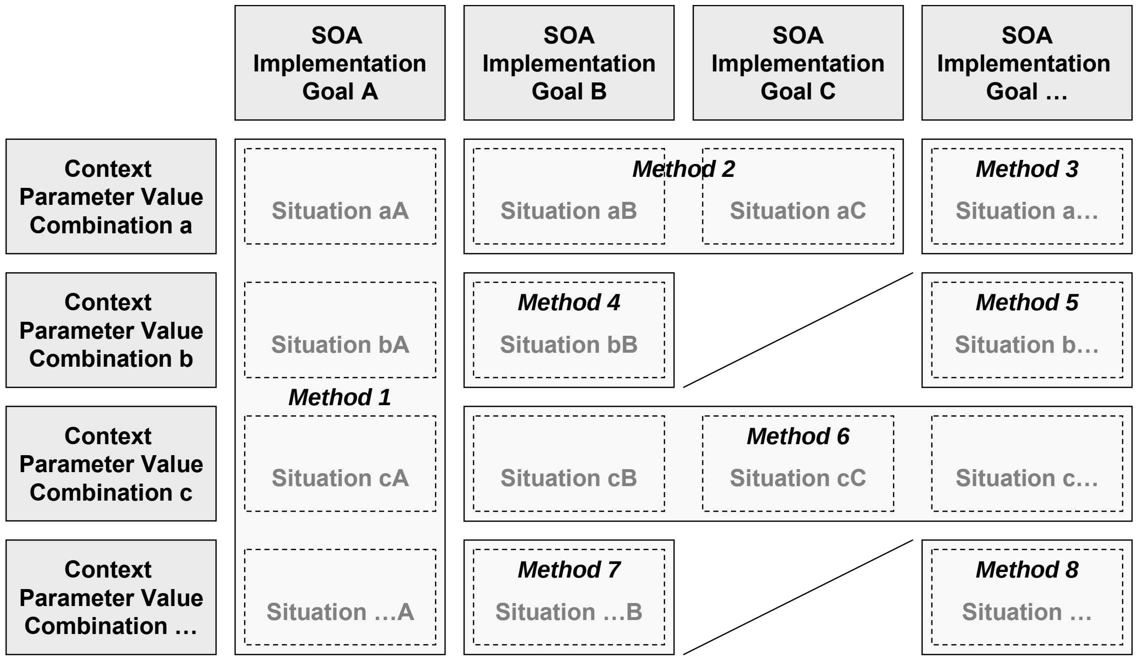

Figure 9. Situation Identification Matrix (Börner, 2010, p. 5)

The mapping of situations to methods is an n:1 relation, i.e. that every situation is assigned to exactly one method. However, methods may be used for several situations in case the preconditions of the latter are similar. As depicted in Figure 9, there are basically three types of methods. First, methods like number 3 or 4 apply to exactly one situation (1:1 relation), i.e. they support one given set of context factors (in the form of a CPVC) and one SOA implementation goal (SOAIG). Second, a so-called goal-specific method like method 1 covers all situations with SOAIG A irrespective of CPVCs. In this case the implementation goal is so dominant that any given CPVC does not influence the choice of a method. Third, methods like 2 or 6 are context combination-specific but span multiple implementation goals. In these cases only environmental factors determine the suitability of a method. The goal of an SOA implementation does not have a considerable effect on this selection.

The principle of assigning methods to situations is not restricted to this third adaptation mechanism. All following considerations can be extended to the first mechanism, too, since a prescribed alteration could be interpreted as a separate method as well.

Although methods that cover more than one situation are not extraordinary, they should come under special scrutiny. For instance, if two CPVCs are always covered by the same methods (only influenced by different SOAIGs) as shown in Figure 10, there is a possibility that this change in parameter values does not influence the situation. This might lead to the finding that, for example, the differentiation of large and small companies is irrelevant if all other context factors are left unchanged. It does not implicate though that this differentiation is irrelevant per se. Insights gained through the application of the matrix can be fed back to improve the meta method and reduce the number of situations before engineering situational methods.

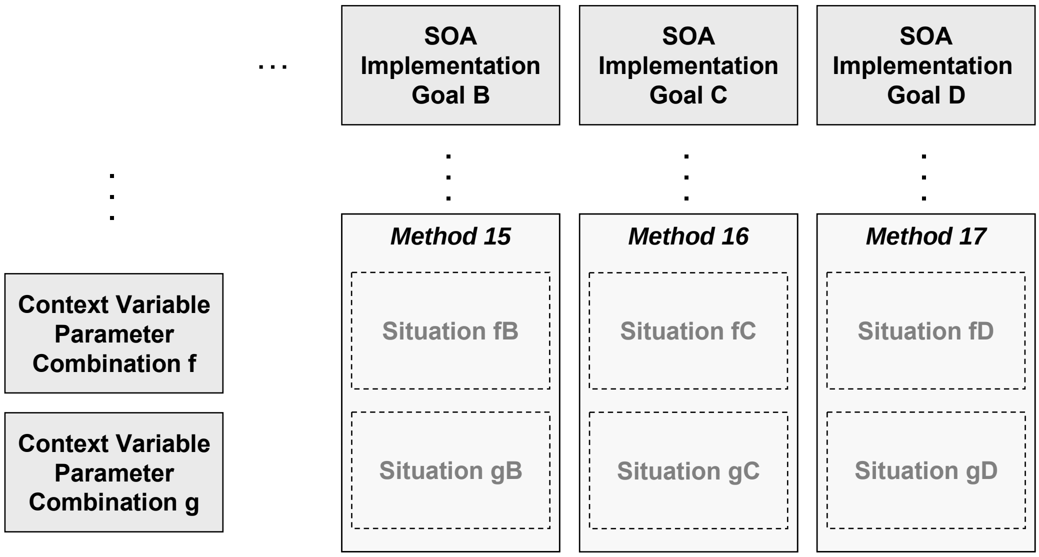

Figure 10. Methods Covering Similar Situations

Moreover, a method assigned to multiple situations might hint at correlations and relationships among context factors themselves and in combination with SOA implementation goals. These methods might span a number of situations in both the x and y dimension of the matrix. Based on a case study and a grounded theory approach, Börner, Goeken and Rabhi (2011b) develop a model that reveals such relationships. It is a further step in empirically validating the context factors that are used for the meta method.

Theoretically, there are many fragments – 16 alone have been elaborated in the previous section – that could be combined to thousands of slightly different methods. This would lead to the problem that after identifying a situation one method out of a million would have to be assigned to this situation which is almost impossible. Of course, not all combinations of fragments make sense. Dependencies, for instance concerning inputs and outputs or regarding roles, have to be taken into account. In order not to engineer a plethora of useless methods, the latter should be engineered with possible application situations in mind.

As shown in Table 8, utilization of a base method and composition time are the distinctive characteristics of adaptation mechanisms. Not using a base method that is later configured slightly in the course of a project but composing fragments instead, seems to be the more applicable approach. It provides flexibility and enables project managers to adapt the used method to changing circumstances even after the project start.

As far as composition time is concerned, the main difference between the mechanisms is the time when expert knowledge is required. For mechanism one and three expert knowledge about building methods for service identification has to be available at the time the method and its fragments are created (t_1). Mechanism two requires expert knowledge at the time of applying a method, i.e. when the project takes place (t_2). If a project leader is confident about all aspects of service identification, the latter approach seems appropriate to provide for a suitable configuration of fragments.

The meta method presented in this paper does not prescribe one or the other option. All of them can build upon the meta method developed in the previous sections. However, the exemplary application will facilitate the second adaptation mechanism, i.e. no base method for further configuration will be developed. The composition of fragments will take place at time (t_2), i.e. when the project starts.

7 Application at the Example of efiport AG

The starting point for the service identification conducted at efiport was the company's product Campus & Learning Management System (CLM). Hence, it was a rather product-driven approach to service identification.

It is important to note two different intentions pursued by the author of this paper (in the following "investigator") on the one hand and managers of efiport on the other hand. The former is primarily interested in applying the meta method to a particular situation to demonstrate its applicability. Hence, an instantiation of the meta method, i.e. one method for service identification tailored to this situation, is the intended outcome. For efiport's management, solely a method for service identification does not yield any tangible results. Their interest is to apply the method in order to identify at least two or three actual services as a pilot for further SOA projects. Conflicts of interests can subsequently arise as the following example shows: While analyzing context factors, the investigator finds neither SOA nor BPM skills among the employees. For the configuration of the method this simply means that no fragments requiring these skills can be selected unless there is external support. The investigator is not interested in changing the situation (and indeed he should not) because the method should be engineered according to the circumstances at hand. However, efiport's management – hoping to identify pilot services for further projects – expects that the investigator contributes actively to the service identification by using his BPM skills and incorporating process know how from efiport's primary customer, the Frankfurt School of Finance & Management (FS). Acknowledging these diverging targets, this paper will follow its primary goal and thus develop a situation-specific method for the CLM case based on the given situation without intervention (see section 7.3). Only in a second step, by actively contributing to the project, a business process of the FS will be modeled to support the identification and yield concrete outcomes in the form of services for efiport (section 7.4).

7.1 Introduction of the Scenario

The company efiport is a software developer that is owned by the Frankfurt School Verlag which in turn belongs to the Frankfurt School of Finance & Management. It has developed the Campus & Learning Management System which is used at the FS and further academic institutions for administration purposes. Other companies such as banks use e.g. the product "clicnet" to manage staff training programs. All products are hosted on servers at efiport. Customers can access them via their web browsers. The FS, for instance, uses the full scope of functions offered by CLM whereas other customers use only parts of it. For this reason, functionalities of CLM will be explained based on examples of the FS.

The three main areas of CLM are *partner management*, *product management* and *event management*. In the partner management section all business partners of the FS are administered. This includes students, employees, lecturers, industry partners, etc. as well as organizations such as academic institutions and companies. It may include information such as address, contact details, grades for courses, mailing history, etc. This data is used for instance to make bookings in study programs or events, issue invoices or for marketing activities.

The product management can be used to manage study programs that are the "products" of academic institutions. Thus, all courses for academic programs such as bachelor and master can be found there. Additionally, all non-academic courses such as certification programs are included. The product management is the basis for the event management where single lectures that belong to a larger program or course can be administered. Both lecturers and students can be assigned to events, i.e. courses. Moreover, extra-curricular events of all kinds, e.g. conferences and receptions can be managed in this part of CLM. This includes the booking of rooms and lecture theatres. All event bookings are linked to a personal calendar for all users of CLM and respective calendar entries are generated automatically.

Developers of efiport are in the comfortable position of working with an almost homogenous IT landscape. An oracle database is the common ground for a client-server architecture. On top of it, a three layer architecture is provided containing a data extraction layer, a business logic layer and the graphical user interface (GUI) as depicted in Figure 11. The three layers are clearly divided so that changes in one layer have only a minimum impact on other layers.

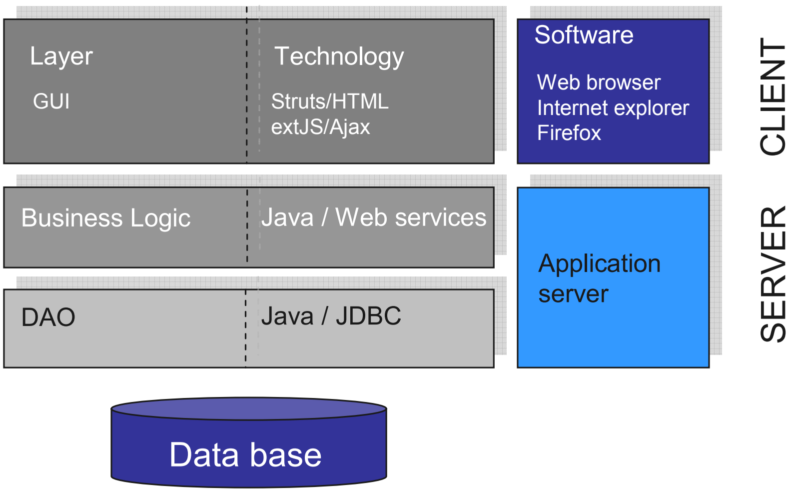

Figure 11. Client-Server Architecture of CLM (Source: efiport)

The system runs on Linux servers that are operated by efiport. CLM is a Java Platform Enterprise Edition (J2EE) application on distributed servers. The business logic is implemented in Java on the server. On the data extraction layer, Java Database Connectivity (JDBC) is used to access the underlying Oracle database.

The web-based GUI uses Java Server Pages and is presented in the user's web browser such as Microsoft's Internet Explorer or Mozilla's Firefox. Whenever a user clicks a button, the whole page is requested from the server via Hypertext Transfer Protocol (HTTP). At the moment, there are efforts to include Ajax and Java Script to enable a partial update of contents on the page so that not all of the data has to be requested from the server. Using Java Script will also make a drag & drop possible.

There are several interfaces to other software in the FS. The issuance of invoices in the partner management is linked to accounting software. Every night, new invoices are transferred as a batch to the accounting system. Furthermore, contents from the partner management are used for FS's website. The data is extracted by the content management system Magnolia and provided for the homepage.

7.2 Identifying the Situation

This section will elaborate on the situation in the case of efiport. Thus, context factors and the SOA implementation goal will be analyzed. For each context factor, the respective parameter value (CPV) will be identified. Based on the identified situation a method for service identification will be engineered in the next section.

With 16 employees, efiport can be classified as a small company without a doubt (**CPV 1.1**). A flat hierarchy inevitably involves all employees in many decisions that go beyond routine tasks. The degree of formalization (and thus documentation) of processes is extremely low. This is counterbalanced by intensive communication and a common understanding among both management and employees. Since efiport is an IT (service) provider, software functionalities are usually offered to external customers. The same holds true for CLM. Thus, legislation on data privacy, for example, has to be considered when personal data (such as FS's partner management) is stored in databases operated by efiport. There are no services available at efiport yet. Services developed on the basis of an identification process would be used by efiport's application development team in the first place. They would not be offered as services to third parties. However, customers would indirectly invoke services if they use CLM. Hence, services should be designed as if they were offered externally but still must adhere to internal policies (**CPV 2.3**).

Implementing a service-oriented architecture is not a focus at efiport at the moment. There have been efforts to use services before but they have not yielded any significant results. Since this is an academic project with limited scope, there is no real budget assigned for it (**CPV 3.2**). The management is aware that a maximum of three or four employees will spend a couple of hours to support this project. The project is considered a pilot. If it succeeds in identifying some services that can be implemented, there might be a follow up project to pursue the service-oriented paradigm more insistently. Subsequently, the approach to service identification is rather pragmatic due to limited resources. Although there was a small project on services before, there are no considerable SOA skills among efiport's developers. As far as Business Process Management is concerned, the skills would have to be provided on the customer's side, i.e. by the FS's staff. There are efforts to foster BPM in the FS but there is only one business process ("New Product Process") modeled so far and experience is scarce (**CPV 4.4**).

Since there is no service-oriented architecture in efiport and experience is limited to a small project, the SIMM level is 1 (**CPV 5.1**). The software and functionalities provided by efiport deal with sensitive customer data. Thus, handling of this data has to comply with data privacy laws. Although banks belong to efiport's customers, no industry-specific regulations (such as the "Kreditwesengesetz") have to be considered because in these cases CLM is only used to manage staff training activities (**CPV 6.1**). Only banks themselves would have to consider such regulation in case they are planning to out-source respective services. Since efiport is a small IT provider, the whole company could be considered as an IT department in the sense that the IT landscape and architecture is governed centrally (**CPV 7.1**). However, a comprehensive overview and documentation of this architecture does not exist. Besides CLM, efiport provides a couple of other products to customers including teaching content for e-learning and test generators for surveys or exams. Subsequently, it can be classified as a multi-product company (**CPV 8.2**).

The combination of context factors can be considered as an eight dimensional vector. For this project, the CPVC (efiport) = (1.1/2.3/3.2/4.4/5.1/6.1/7.1/8.2). As discussed in section 4.3, there are 1,152 possible CPVCs for the context factors identified. Figure 12 illustrates the pattern of the efiport project in a morphological box.

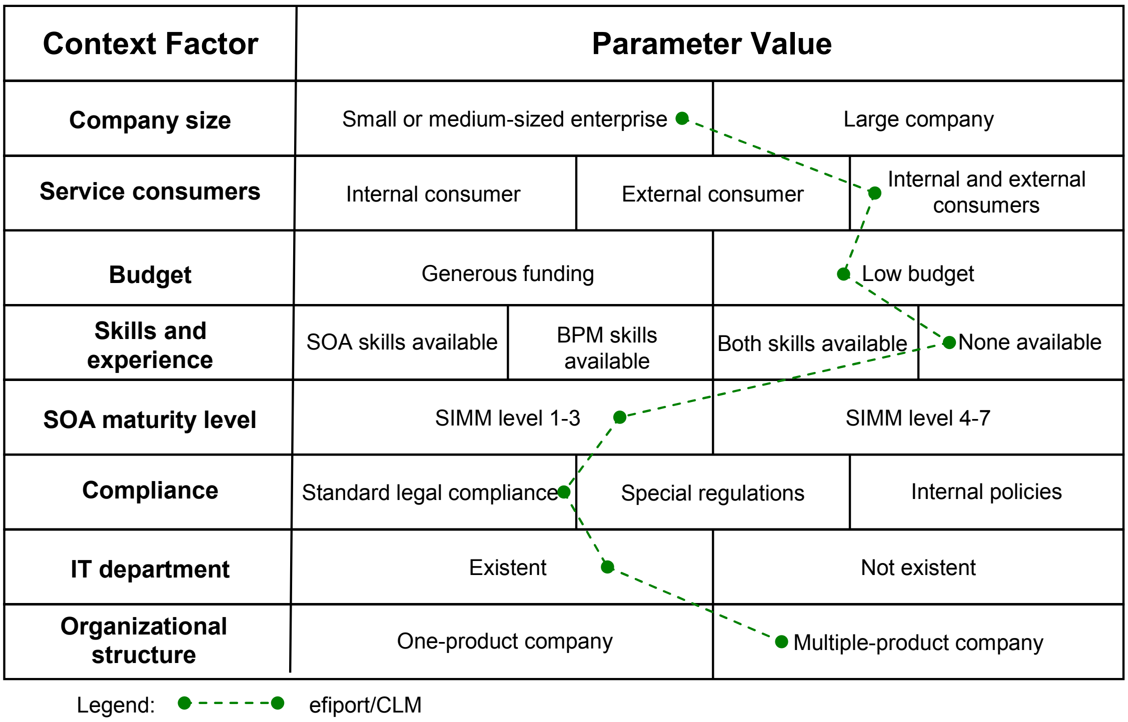

Figure 12. Context Factor Parameter Values for efiport/CLM

Finally, a situation depends on the SOA implementation goal that is pursued by establishing a service-oriented architecture. This goal is not unambiguous in efiport's case. The management's approach might be best described by "Let us identify some services and see what we can make out of it." The head of Application Development has more concrete ideas in mind. His aim is to replace functionalities in the existing client-server architecture by services. Some functionalities are used in multiple products (such as CLM and clicnet) and processes and are thus candidates for services.

For the purpose of this paper we assume that efiport's management is primarily interested in **standardization**. There had been previous efforts of implementing services in the course of a student project but there was no follow up so that the services are no longer used. If the concept of services is found to be appropriate for efiport, services with standardized interfaces could be implemented. On the one hand, this would improve reusability. On the other hand, the flexibility of combining services would increase significantly. In a next step, these standardized services could be provided directly to third parties. Especially companies that point out the importance of operational excellence and thus, process management could be targeted this way. The example of the FS shows the growing interest in business processes through the current appointment of a process manager.

7.3 Engineering a Situational Method

In order to engineer a situation-specific method for the efiport/CLM case, the second adaptation mechanism discussed in section 6 will be used, i.e. the method will be composed at project time (t_2) without using a pre-configured method. Particularly, the availability of information (documented or not) that is not necessarily linked to context factors can be considered that way. Fragment descriptions include the definition of inputs and other preconditions so that the availability of information clearly restricts the use of fragments. Table 9 gives an overview of the previously developed fragments that shall be assembled to a situational method in the following. If and how fragments will be used depends on necessary inputs and preconditions.

Fragments	Inputs	Preconditions
Fragment 1: Overview of Existing Process Models	Documented business processes.	If no BPM projects had been conducted before, it is necessary to identify business processes before using this fragment.
Fragment 2: Asset Analysis	None.	There has to be an existing IT infrastructure and of course, in virtually all cases there is some kind of IT already implemented. The only conceivable exemption is a start-up company that is built from scratch without any legacy systems. Thus, this fragment could be omitted in a service identification project within a start-up company.
Fragment 3: Identifying Stakeholders	Strategy documents.	A communicated strategy that includes stakeholders must be given. The management's commitment to support this strategic task or at least a certain degree of awareness is necessary.
Fragment 4: Identifying Activities	Documented business process(es).	Employees involved in the operational work have to be available for workshops.
Fragment 5: Compiling Service Candidates	Activity diagrams or BPMN diagrams.	None.
Fragment 6: Develop Key Performance Indicators	Business goals.	Based on previously identified business processes, the management must have defined business goals that should be achieved by these processes.
Fragment 7: Prioritization of Funding	Set of reusable services, feasibility check.	All desirable and feasible services have been identified before.
Fragment 8: IT Governance Analysis	Information on IT governance, IT strategy, SOA strategy.	Derived from an organization's strategy, the IT strategy must be defined and documented before this fragment can be used.

		Often, conventions and principles regarding the implementation and development of IT are not explicated in readily available documents. Therefore, it might be necessary to interview (IT) managers in order to retrieve more detailed information.
Fragment 9: Identifying Business Processes	Stakeholder documentation.	Knowledge and experience of the management and staff from the business department must be captured. Thus, the availability of management and staff for interviews is essential.
Fragment 10: Developing Reutilization Scenarios	Service candidate list.	This fragment is frequently used if no comparison of existing use cases is possible, i.e. if there is only one or even no use case at all. However, it can be used even if multiple use cases and business processes are well-documented.
Fragment 11: Analyzing and Improving Reusability of Services (or Candidates)	List of service candidates or existing services, respectively, documentation of use cases and reutilization scenarios, documentation of existing assets.	Analyzing the frequency of service invocation is only possible if there are services in place already. Furthermore, there must be a measurement system in place that provides data for this analysis. A Service Design Unit must be installed in the organization.
Fragment 12: Feasibility Check	List of service candidates, documentation of existing assets, documentation of use cases or reutilization scenarios.	An asset analysis must have been conducted beforehand.
Fragment 13: Goal Service Modeling	Business processes, strategy documents.	Strategy awareness on the management's side is necessary. Abstract strategic goals like a company's vision have to be operationalized in order to have more concrete, achievable and timely fixed business goals.
Fragment 14: Strategic Stakeholder Integration	Stakeholder documentation.	This fragment is only useful if stakeholders other than the end consumer (for instance suppliers) should be integrated into the process. A strategic view on service orientation and management awareness is thus necessary.
Fragment 15: Composition of Basic Services	List of existing services.	A company must have achieved a SIMM level that already provides for existing basic services that can be aggregated to composed or even process services.
Fragment 16: Breaking Down Business Processes	Documented business process(es).	Employees involved in the operational work have to be available for workshops.

Table 9. Fragments for Service Identification

Due to the low degree of formalization (because of the small company size) found in efiport, fragments based on the analysis of documents are less suitable. There is a significant amount of tacit knowledge among employees that needs to be explicated. Hence, questionnaires, workshops or interviews are better suited to retrieve information (e.g. fragment 16). Although customer interaction of services is likely in the case of CLM, the integration of customers is not the primary concern. Fragments that consider strategic aspects of customer integration and illustrate a line of interaction e.g. through swim lane diagrams are consequently omitted here. The small scope of the project without a real budget prohibits the establishment of new organizational units such as a Service Design Unit and therefore rules out fragments such as 8, 11, 13 and 14.

There is only little documentation on business processes that can be used for a detailed analysis of work tasks that could be supported by services. A lack of BPM skills at both efiport and its customer FS makes an application of fragments (using Business Process Management techniques) difficult. The same is true for the limited SOA skills. However, many activities, for instance the analysis of existing IT infrastructure, can be performed without special SOA skills. Only the suitability of the outcomes for a service-oriented architecture might be compromised. Available skills in this field are thus not absolutely necessary but would certainly improve the results of certain fragments and support further steps. Many fragments can only be applied in companies that have reached a certain SIMM level. The composition of basic services to composed or process services for example assumes existing services but cannot be used in a monolithic or silo application landscape. Subsequently, fragments like number 15 are inadequate in the case of efiport.

Fragments that cover compliance to industry- or country-specific regulations can be omitted. Since the IT architecture is governed centrally, it is not necessary to organize workshops or interviews with department heads in order to get an overview of isolated software and hardware distributed throughout the company. If there is any documentation of the existing architecture, it is stored centrally and available from the head of Application Development. Fragments using the role *IT analyst* can be used. The fact that efiport offers multiple products to its customers raises the question if there are services that could be invoked by users of different products. Since there are no services established, an invocation of certain functionalities could be measured as a proxy if this is technically possible. Fragment 11 is not useful in this case because there is only one business process and thus, a reusability potential across processes cannot be identified.

Since standardization of services is the primary goal of service identification in efiport's case, fragments dealing e.g. with strategic aspects like stakeholder integration (fragment 14) are less important.

Based on the meta method, a situation-specific method for service identification in the efiport case would be engineered as follows:

- Since there was no BPM project in advance, there are no business process models available which makes fragment 1 inapplicable in this project. For a start, at least one business process should be analyzed in depth. For the identification of services in efiport's case it makes sense to analyze a customer's business process to learn how, when and by whom functionalities in a product like CLM are used. Thus, FS's management could be interviewed to identify a business process (fragment 9).
- Afterwards, the process has to be broken down to sub processes and activities. It is absolutely necessary to involve employees at this stage in order to guarantee that all work steps are taken into account. Fragment 16 will deliver BPMN diagrams as a result of this break down.

- These are the input for fragment 5 in which potential services (so-called service candidates) are elaborated and compiled in a list.
- Fragment 11 builds on this service candidate list. Unfortunately, this fragment cannot be used in efiport's case because there are neither services (and thus invocation frequencies) nor multiple use cases that could be analyzed. Moreover, the role SDU is not available in this case. Instead, fragment 10 followed by fragment 12 can be used as alternatives.
- At some point before applying fragment 12, an asset analysis (fragment 2) must have been carried out. The existing IT infrastructure is important if you want to assess the feasibility of services. Although the "as-is situation" might lead to a number of technical restrictions, it should not be considered unchangeable. Thus, a feasibility check could imply different consequences. On the one hand, certain services might be ruled out because the infrastructure is inappropriate. On the other hand, changing the underlying architecture could enable the implementation of one or more services.
- Fragments concerned with financial measures and value analyses, e.g. number 6 and 7, do not play a role here because of the goal of this project and its limited scope.

Figure 13 illustrates the method fragments that were chosen for the efiport case. Furthermore, all intermediary results – being output and input for respective fragments – are shown.

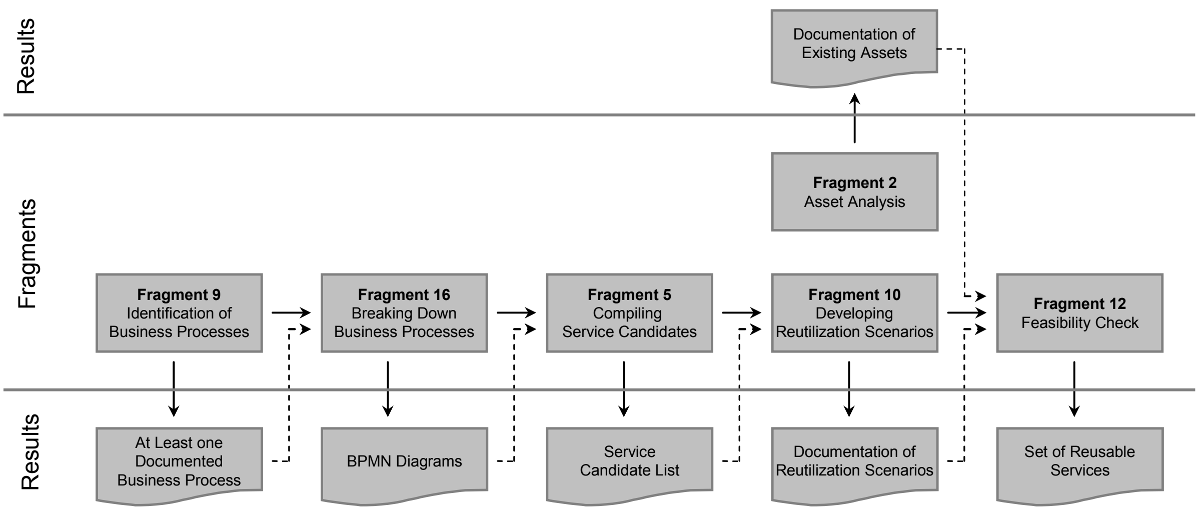

Figure 13. Method Configuration for the efiport Case

The selection and sequence of fragments are tailored to the efiport case. Together, they constitute the situational method that suits the project at hand. The engineering process of this method has been guided by the previously developed meta method. Thus, this sub section yields a concrete method that is used for the purpose of demonstration as demanded by Peffers et al. (2008).

7.4 Above and Beyond Method Configuration

As discussed on previous occasions, the configuration of a situational method is the focus of this paper. The service identification that is described in the following shall show that the situational method for efiport can be used to actually identify services.

While talking to efiport's management for the first time, it became clear quite quickly that there are no documented business processes available. The CLM software was developed and is updated based on requirements from users rather than on business processes. In order to identify business processes that involve CLM and to examine how it is used, the investigator was referred to the process manager and head of internal auditing at the Frankfurt School of Finance & Management. Since the idea of process management is fairly new at the FS as well, there is no comprehensive overview of business processes. As a prototype, a single process called "New Product Process" was under scrutiny. According to Figure 13, the fragment "Identification of Business Processes" was used in order to identify and document at least one business process.

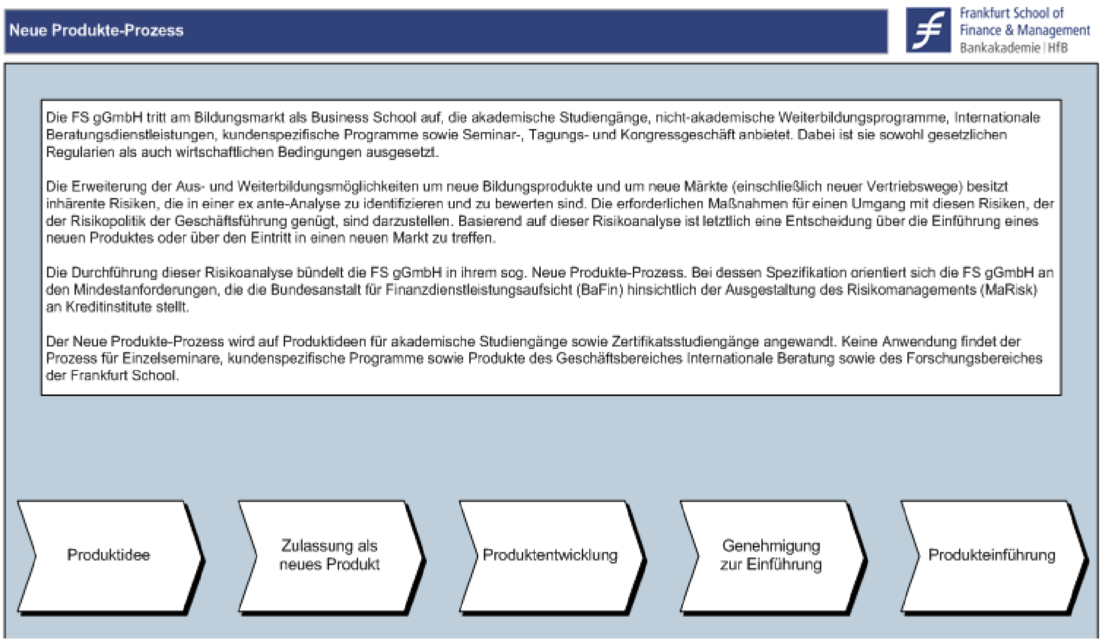

Figure 14. "New Product Process" on Top Level

The analysis, break down and documentation of this process was done with the modeling tool "ViFlow". Figure 14 is a screenshot that could be taken from any user's web browser. It shows the top level of the "New Product Process" that is a blueprint for both academic and non-academic programs at the FS. This process is subdivided in five sub processes from "product idea" to "product implementation".

The next step in the service identification process is fragment 16 "Breaking Down Business Processes". By breaking down the top level business process to sub processes and further to work steps, several BPMN diagrams were created. Part of the sub process "product implementation" is the activity "Definition in CLM". Figure 15 illustrates how BPMN can be used to depict the activities that are necessary to define a new product in CLM. BPMN includes roles (i.e. departments within the FS and efiport) and corresponding responsibilities represented by horizontal swim lanes. Connectors like "AND", "OR" and "XOR" (exclusive or) can be used to describe optional or parallel activities.

As a first step of this activity, a study program brochure for the new product (academic or non-academic) is developed by the marketing division of the FS. In any case, the Application Development of efiport has to define the product from the IT side (AND connector). On the business side, the division "Firmenprogramme & Services" (FP&S) is responsible for the definition of non-academic courses, whereas "Konzeption & Programmentwicklung" (KPE) has to define academic programs. This is depicted by an XOR connector in Figure 15. Once the product is defined, single lectures have to be assigned. Again, the Application Development is involved in any case and FP&S takes care of non-academic courses. If the new product is an academic program though, definition of lectures is a coordinated process between KPE and the division "Studienbetreuung" (SB). Only after all necessary roles contributed their part, the product is completely defined and the activity terminates.

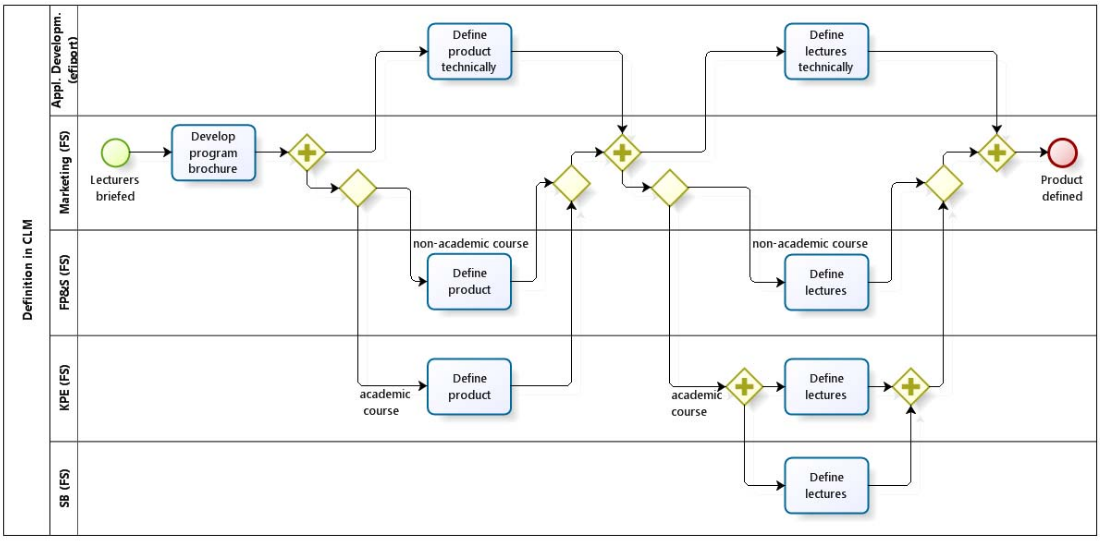

Figure 15. The Activity "Definition in CLM" in BPMN

Within the analyzed "New Product Process", there are 41 activities on the respective level of granularity that could be depicted in BPMN as exemplarily shown in Figure 15. The resulting BPMN diagrams were used as input for fragment 5 "Compiling Service Candidates". The detail level of BPMN diagrams is granular enough to separate automated and semi-automated from manual work steps. For each work step, relevant IT support could be identified and subsequently, a service candidate list has been compiled. Table 10 shows an excerpt of this list.

Activity	**Work Step**	**Explanation**	**Supporting IT System**	**Service Candidate**
Specification of details	Develop course and module description	Since it is a new product, no existing descriptions can be used. Once developed, the description can be uploaded to the intranet.	CLM product management	Upload course and module description. Administer description.
	Define marketing strategy	Manually	-	-
	Plan budget and media appearance	Manually	Magnolia Content Management System	-
	Acquire lecturers	Contact information retrieved from partner database	CLM partner management	Propose lecturer based on professional experience
Instruction of lecturers	Develop lecturer's manual	Manually	-	-
	Conduct meeting with all lecturers	Room bookings have to be made, a new event has to be created and booked in the users' calendars. Lecturers must be contacted.	CLM event management, CLM partner management, users' calendars, room booking	Check room bookings when event is created in event management;
Definition in CLM	Develop study program brochure	Manually	-	-
	Define product (technically)	Manually	CLM product management	-
	Define lectures	Room bookings have to be made, new events has to be created. Dates must be booked in lecturers' and students' calendars	CLM event management, room bookings, users' calendars	Check room bookings when event is created in event management;
Prepare assessment center (AC)	Develop AC concept	Manually	-	-
	Develop tests for AC	Check existing tests	Shared Workspace	-
	Fix dates for AC	Room bookings have to be made, new events has to be created. Assessors have to be acquired.	CLM event management, room bookings, CLM partner management	Check room bookings when event is created in event management; Proposes assessors based on professional experience.

Table 10. Excerpt of Service Candidate List

Based on this list of service candidates, FS's process manager together with the investigator tried to find out if there were identical or similar work steps and which service candidates could be used for various work steps. One quite obvious service candidate can e.g. support the creation of new events by automatically checking existing room bookings. This service could be invoked as part of the work steps "Conduct meeting with all lecturers", "Define lectures" and "Fix dates for AC". A less obvious opportunity to provide a reusable service is the automated proposal of lecturers and assessors. The former are needed to teach courses, the latter to assess applicants in assessment centers. If the database included information on a business partner's field of experience (e.g. "mergers and acquisitions") and their job experience (e.g. "more than ten years"), both could be identified by a simple query. Potential lecturers or assessors could subsequently be proposed to the organizer by a service. A proposal list would be based on the former's professional experience that must be provided in CLM's partner management. Services could be incorporated in an additional layer between sub processes and IT systems. Figure 16 exemplarily shows how the activities "Prepare Assessment Center" and "Specification of Details" use IT systems and could invoke services.

In an intermediary step, services can be introduced in an additional layer (enterprise service bus) and continue to invoke the existing applications. Later, services can replace these applications by providing the functionalities themselves. If this is not possible due to technical restrictions, legacy systems will stay in place and services will be used as wrapper to provide a standardized interface.

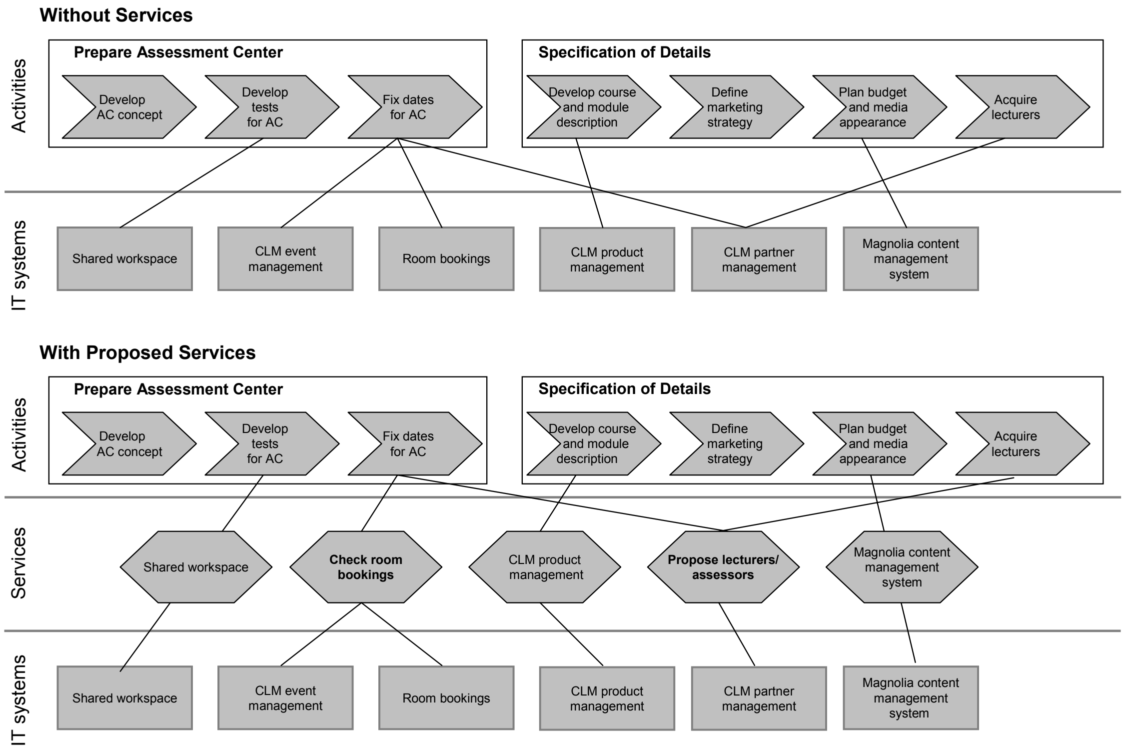

Figure 16. Adding a Service Layer

The services depicted in the lower part of Figure 16 offer two opportunities. Firstly, the service "Check room bookings" is able to integrate two database accesses, i.e. every process or user only needs to invoke a single service instead of placing two requests in different IT systems. Secondly, the claim of reusability can be fulfilled e.g. by the service "Propose lecturers/assessors" that is used for two activities from different processes. If Figure took all processes in Table 10 into account, the service "Check room bookings" would be invoked by three different processes which shows its potential for reusability.

These reutilization scenarios are one input for fragment 12 "Feasibility Check" coming from the business side. The second input to this fragment, i.e. a documentation of existing assets, is the result of an asset analysis (fragment 2) that must have been conducted on the IT side previously. So before the feasibility check was performed, efiport's head of Application Development provided multiple sorts of information about efiport's IT infrastructure that was presented in detail in section 7.1.

Finally, the feasibility check described in fragment 12 can be conducted since all necessary inputs are available. From the business side, two exemplary service candidates, namely "Check room bookings" and "Propose lecturers/assessors" (see Figure 16) were identified. An asset analysis was conducted to get an overview of existing technology. There are only little restrictions on the software development part since solely the relational database and the firewall are commercial products. Apart from that, software was either developed by efiport itself or is open source. As far as the two above mentioned candidates are concerned, no limitations were found during the feasibility check. Thus, both services could be included in the "Set of Reusable Services" that is the result of fragment 12.

Focus of this paper was the construction of methods for service identification and an exemplary application to the case of efiport. However, this sub section moved beyond method construction and used one instantiation of a method to identify services in CLM that can be implemented as part of a future service-oriented architecture. Thus, it could be shown that a service identification method constructed by applying the meta method is actually able to yield services.

8 Limitations

The construction of the meta method belongs to the domain of design science. Guidelines of design science research were utilized as far as possible. However, there are a number of limitations concerning scope, generalizability and validation of the results.

The identified context factors are based on an extensive literature research. Moreover, their significance was supported by two case studies where qualitative scientific methods such as grounded theory were used. However, desk research was the primary research method which might restrict the findings. Probably, there are more potential factors that could influence a situation and it is important to find the decisive ones. Multiple case studies could yield a comprehensive list of these factors. Research methods such as factor analysis could help to understand the impact of single factors on the resulting method. An investigation into relationships and interdependencies between context factors is subject to ongoing research. Börner et al. (2011b) present a model showing these relationships based on several hypotheses.

In order to identify situations, parameter values were used for each context factor. As discussed in section 4.1, these parameter values can be either deductive or inductive in nature. Whereas deductive parameter values can be derived logically, the derivation of inductive CPVs can be challenged for three reasons. Firstly, is the CPV an appropriate operationalization of the context factor? The number of employees certainly reflects the size of a company but the turnover in a certain period could be used, too. Secondly, how many classes of a context factor are adequate? There might be good reasons distinguishing between small and large enterprises or maybe include medium size as a third class. Thirdly, how are these classes divided? Is a company with 100 employees still a small company or is it medium-sized already? All decisions on CPVs are not unambiguous.

Their justification is based on literature and experience as good as possible and could be improved in further case studies while applying the meta model developed herein. A discussion about adequate criteria (such as number of employees vs. turnover) is justified but should not be overestimated. In the example of efiport the result would have been a “small company” in both cases. The same is true for the total number of employees used to separate small from large companies. For most companies the classification stays the same, no matter if 250 or 300 employees are the threshold.

The development of method fragments is theoretically supported by two fragment construction approaches, namely re-engineering and ad-hoc construction. Since at least the former uses parts of existing methods, a certain degree of reliability can be assumed. Still, there is no guarantee that there are more, other or better fragments that could be used to configure situational methods for service identification. In order to improve this reliability, ongoing research by Looso, Börner and Goeken proposes a new research method called Grounded Method Engineering (GME). GME uses techniques from grounded theory to derive elements that can be used for method engineering. In the same way, fragments for situational method engineering could be developed. Further research could, for instance, build on expert interviews. Moreover, experience from the application of the meta method could be used to improve the method by adjusting elements, fragments or configuration mechanisms.

As illustrated in Figure 6, this paper delivers the basis for situational methods in service identification projects, i.e. specific situations and fragments. The applicability of fragments depends on certain inputs, i.e. fragments cannot be combined arbitrarily but are bound to preconditions. However, a comprehensive set of configuration rules for all conceivable situations that can occur in service identification projects is out of the scope of this paper and consequently left to further research. An investigation into how far one method can support several situations (see also section 6) would be interesting.

In theory, service identification projects (and the methods proposed in literature) consider the entity “organization” or “company” as starting point for service identification. The structure of companies can differ considerably which is one reason why this paper advocates the engineering of situational methods. For instance, the existence of an IT department influences the applicability of certain fragments. In real-life projects, however, company borders can be fuzzy and a definition of what is an IT department or not can be unclear. In efiport’s case, the company is closely linked to the Frankfurt School of Finance & Management. Actually, until ten years ago it used to be FS’s IT department. At the same time, FS is still efiport’s main customer but third parties become increasingly important. This leads to the problem that parts of the company itself are still a de facto IT department of FS. Hence, relevant business processes for service identification are to be found on the customer’s side. This example shows that differentiations like “internal/external consumer” or “IT department existent or not” seem to be straight forward in theory but can be quite fuzzy in real-life projects.

The identification of services in the efiport case was strongly driven from a product point of view. Thus, not the entity “efiport” was the basis for the investigation but rather its product CLM. This is only one example for conflicts of interests that were encountered while conducting the project with efiport. Another problem was that the investigator's aim was to identify the situation at efiport "as-is" and subsequently choose adequate fragments and omit inadequate ones to configure a suitable method for service identification. The management of efiport was interested in concrete results in the form of services (see section 7). It was important to separate both aims from each other as good as possible in order to develop the method for service identification with the best fit for efiport's case without the investigator himself interfering into the project.

Besides developing a meta method, this paper shows the applicability of the latter at the example of efiport/CLM. Of course, this applicability check does not claim representativeness nor is it easily generalizable. Further evaluation of the meta method through case studies, action research or statistical methods is left to future research.

9 Conclusion

This paper showed how concepts of situational method engineering can be transferred to the domain of service identification. For this purpose, context factors and SOA implementation goals were elaborated to identify situations. Furthermore, method fragments for service identification were developed and their assignment to situations was discussed. Jointly, this provides the basis for a configurability of methods for service identification that is absent from existing methods so far. Hence, the herein created meta method is able to guide a situation-specific configuration of methods for service identification. The applicability of the meta method was demonstrated for the case of an IT service provider.

The meta method developed in this paper can be used with all three adaptation mechanisms presented in section 6. However, the application at efiport showed that – given a certain level of know how among project staff – method fragment composition according to Bucher et al. (2007) seems to be the most promising approach for situation-specific engineering of service identification methods. The flexibility to adapt the method and use other fragments is given throughout the project as long as preconditions of fragment use are met and necessary inputs, roles, etc. are available. The project manager can react to changing circumstances by composing fragments differently even after a project has started.

Future research could investigate in how far the idea of the work product pool approach by Gonzalez-Perez (2007) can be transferred from method results to complete method fragments for service identification methods. If method fragments were accessible in a central repository, method engineers could easily search for and configure existing fragments. Even an automated configuration based on the parameters of a situation could be implemented in software using Method Engineering Language (Agerfalk et al., 2007; Brinkkemper et al., 2001).

Applying the meta method in further projects and case studies is essential to gain further insights in the reasonable configuration of fragments. Experiences from the method-in-action should be fed back to the construction of methods and the design of fragments (Karlsson and Agerfalk, 2007). "This feedback is typically done continuously throughout the project" (p.28). Since situations, fragments and the method engineering process described herein are based on little empirical evidence so far, this feedback is even more important. Further experience could also be used to assign weights to context factors and make the assignment of methods to situations more reliable.

Finally, it is assumed that engineering a method for service identification guided by this meta method leads to a better identification of services because the project context is considered adequately. Project managers do not have to worry about inappropriate method fragments that do not create any value in a project at hand. Instead, after determining the concrete situation, they can concentrate on identifying and designing services in the best possible way under given circumstances.

References

Agerfalk, P. J., Brinkkemper, S., Gonzalez-Perez, C., Henderson-Sellers, B., Karlsson, F., Kelly, S. and Ralyté, J. (2007), Modularization Constructs in Method Engineering: Towards Common Ground?, in: Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.), *Situational Method Engineering: Fundamentals and Experiences*, Springer, Boston, 359-368.

Allen, P. (2006), Service Orientation: Winning Strategies and Best Practices, Cambridge University Press, Cambridge.

Alter, S., Börner, R. and Goeken, M. (2009), Operationalisierung der IT-Governance-Kernbereiche für die Identifizierung und Gestaltung von Services, *Informatik 2009*, Lübeck, Germany, 3660-3674.

Anderson, D., Howell-Barber, H., Hill, J., Javed, N., Lawler, J. and Li, Z. (2005), A Study of Web Services Projects in the Financial Services Industry, *Information Systems Management*, 22, 1, 66-76.

Archer, L. B. (1984), Systematic method for designers, in: Nigel Cross (Ed.), *Developments in Design Methodology*, John Wiley, London, 57-82.

Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S. and Holley, K. (2008), SOMA: A method for developing service-oriented solutions, *IBM Systems Journal*, 47, 3, 377-396.

Arsanjani, A. and Holley, K. (2006), The Service Integration Maturity Model: Achieving Flexibility in the Transformation to SOA, *IEEE International Conference on Services Computing, Chicago*, IL, 515.

Avison, D. E. and Fitzgerald, G. (1995), Information Systems Development: Methodologies, Techniques and Tools, McGraw-Hill, New York, NY.

Avison, D. E. and Wood-Harper, A. (1991), Information systems development research: An exploration of ideas in practice, *The Computer Journal*, 34, 2, 98-112.

Aydin, M. N. (2007), Examining Key Notions for Method Adaptation, in: Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.), *Situational Method Engineering: Fundamentals and Experiences*, Springer, Boston, 49-63.

Bachman, J. (2005), SOA Maturity Model, *Progress Sonic Corporation Homepage*, [http://www.sonicsoftware.com/solutions/service_oriented_architecture/soa_maturity_model/index.ssp], Accessed: 2009-09-25.

Becker, A., Buxmann, P. and Widjaja, T. (2009), Value Potential and Challenges of Service-Oriented Architectures - A User and Vendor Perspective, *17th European Conference on Information Systems (ECIS)*, Verona, Italy.

Becker, J., Knackstedt, R., Holten, R., Hansmann, H. and Neumann, S. (2001), Konstruktion von Methodiken: Vorschläge für eine begriffliche Grundlegung und domänenspezifische Anwendungsbeispiele, *Arbeitsberichte des Instituts für Wirtschaftsinformatik No. 77*, Westfälische Wilhelms-Universität Münster, Münster, Germany.

Becker, J., Knackstedt, R., Pfeiffer, D. and Janiesch, C. (2007), Configurative Method Engineering, *13th Americas Conference on Information Systems (AMCIS)*, Keystone, Colo., Paper 56.

Becker, J., Kugeler, M. and Rosemann, M. (2004), Process Management. A Guide for the Design of Business Processes, Springer, Berlin.

Becker, J., Seidel, S., Pfeiffer, D. and Janiesch, C. (2006), Evolutionary Method Engineering - A Case Study in Meta Modeling, in: Saartje Brockmans, Jürgen Jung and York Sure (Eds.), *2nd International Workshop on Meta-Modelling*, GI, Karlsruhe, Germany, 85-107.

Benbasat, I., Goldstein, D. K. and Mead, M. (1987), The case research strategy in studies of information systems, *MIS Quarterly*, 11, 269-386.

Beverungen, D., Knackstedt, R. and Müller, O. (2008), Entwicklung Serviceorientierter Architekturen zur Integration von Produktion und Dienstleistung, *Wirtschaftsinformatik*, 50, 3, 220-234.

Bieberstein, N., Bose, S., Walker, L. and Lynch, A. (2005), Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals, *IBM Systems Journal*, 44, 4, 691-708.

Böhmann, T. and Krcmar, H. (2005), Modularisierung: Grundlagen und Anwendung bei IT-Dienstleistungen, in: Thomas Herrmann, Uwe Kleinbeck and Helmut Krcmar (Eds.), *Konzepte für das Service Engineering. Modularisierung, Prozessgestaltung und Produktivitätsmanagement*, Physica, Heidelberg, 45-83.

Börner, R. (2010), Applying Situational Method Engineering to the Development of Service Identification Methods, *16th Americas Conference on Information Systems (AMCIS)*, Lima, Peru, Paper 18.

Börner, R. and Goeken, M. (2009a), Identification of Business Services - Literature Review and Lessons Learned, *15th Americas Conference on Information Systems (AMCIS)*, San Francisco, Calif., Paper 162.

Börner, R. and Goeken, M. (2009b), Methods for Service Identification, in: Daniel Moldt, Juan C. Augusto and U. Ultes-Nitsche (Eds.), *7th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS)*, Milano, Italy, 76-84.

Börner, R., Goeken, M., Kohlborn, T. and Korthaus, A. (2011a), Context Factors for Situational Service Identification Methods, *6th International Conference on Internet and Web Applications and Services (ICIW)*, Xpert Publishing Services, St. Maarten, The Netherlands Antilles, forthcoming.

Börner, R., Goeken, M. and Rabhi, F. (2011b), SOA Development and Service Identification - A Case Study on Method Use, Context and Success Factors, Unpublished Working Paper, Frankfurt/Main.

Börner, R., Looso, S. and Goeken, M. (2009), Towards an Operationalisation of Governance and Strategy for Service Identification and Design, *13th IEEE International EDOC Conference*, Auckland, New Zealand, 180-188.

Brinkkemper, S. (1996), Method engineering: engineering of information systems development methods and tools, *Information & Software Technology*, 38, 4, 275-280.

Brinkkemper, S., Saeki, M. and Harmsen, A. F. (2001), A method engineering language for the description of systems development methods, *13th International Conference on Advanced Information Systems Engineering (CAiSE)*, Springer, Interlaken, Switzerland.

Brinkkemper, S., Saeki, M. and Harmsen, F. (1999), Meta-Modelling Based Assembly Techniques for Situational Method Engineering, *Information Systems*, 24, 3, 209-228.

Brooks, F. P. (1987), Essence and Accidents of Software Engineering, *IEEE Computer*, 20, 4, 10-19.

Bucher, T. and Klesse, M. (2006), Contextual Method Engineering, St. Gallen, Switzerland.

Bucher, T., Klesse, M., Kurpjuweit, S. and Winter, R. (2007), Situational Method Engineering - On the Differentiation of "Context" and "Project Type", in: Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.), *Situational Method Engineering: Fundamentals and Experiences*, Springer, Boston, 33-48.

Chau, P. Y. K. (1995), Factors used in the selection of packaged software in small businesses: views of owners and managers, *Information and Management*, 29, 2, 71-78.

Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S. and Rackham, G. (2005), Impact of service orientation at the business level, *IBM Systems Journal*, 44, 4, 653-668.

Cossentino, M., Gaglio, S., Henderson-Sellers, B. and Seidita, V. (2006), A metamodelling-based approach for method fragment comparison, *11th International Workshop on Exploring Modeling Methods in Systems Analysis and Design at CAiSE*, Luxemburg.

Couzens, J. A. (2009), Implementing an Enterprise System at Suncorp Using Agile Development, *20th Australian Software Engineering Conference*, Gold Coast, Australia.

Cronholma, S. and Ågerfalk, P. J. (1999), On the Concept of Method in Information Systems Development, *22nd Information Systems Research Seminar in Scandinavia*, 229-236.

Davis, G. (1982), Strategies for information requirements determination, *IBM Systems Journal*, 21, 1, 4-30.

DeLone, W. H. (1988), Determinants Of Success For Computer Usage In Small Business, *MIS Quarterly*, 12, 1, 50-61.

Deneckère, R., Iacovelli, A., Kornyshova, E. and Souveyet, C. (2008), From Method Fragments to Method Services, *13th International Conference on Exploring Modelling Methods for Systems Analysis and Design*, Montpellier, France, 81-97.

Dubin, R. (1978), Theory Building, Free Press, London.

Eekels, J. and Roozenburg, N. F. M. (1991), A methodological comparison of the structures of scientific research and engineering design: Their similarities and differences, *Design Studies*, 12, 4, 197-203.

Erl, T. (2004), Service-Oriented Architecture - A Field Guide to Integrating XML and Web Services, Prentice Hall, Upper Saddle River, NJ.

Erradi, A., Kulkarni, N. and Maheshwari, P. (2007), Service Design Process for Reusable Services, in: Bernd J. Krämer, Kwei-Jay Lin and Priya Narasimhan (Eds.), *Service-Oriented Computing - ICSOC 2007*, Springer, Berlin Heidelberg, 606-617.

Fitzgerald, B., Russo, N. L. and O'Kane, T. (2003), Software Development: Method Tailoring at Motorola, *Communications of the ACM*, 46, 4, 65-70.

Galliers, R. D., Merali, Y. and Spearing, L. (1994), Coping with Information Technology? How British Executives Perceive the Key Information Systems Management Issues in the Mid-1990s, *Journal of Information Technology*, 9, 3, 223-238.

Gehlert, A., Schermann, M., Pohl, K. and Krcmar, H. (2009), Towards A Research Method For Theory-Driven Design Research, *Wirtschaftsinformatik 2009*, Vienna, Austria, 441-450.

Goeken, M. (2006), Entwicklung von Data-Warehouse-Systemen. Anforderungsmanagement, Modellierung, Implementierung, Deutscher Universitäts-Verlag, Wiesbaden.

Goldkuhl, G., Lind, M. and Seigerroth, U. (1998), Method integration: the need for a learning perspective, *IEEE Proceedings - Software*, 145, 113-118.

Gonzalez-Perez, C. (2007), Supporting Situational Method Engineering with ISO/IEC 24744 and the Work Product Pool Approach, in: Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.), *Situational Method Engineering: Fundamentals and Experiences*, Springer, Boston, 7-18.

Gregor, S. and Jones, D. (2007), The Anatomy of a Design Theory, *Journal of the Association for Information Systems*, 8, 5, 312-335.

Greiffenberg, S. (2003), Methoden als Theorien der Wirtschaftsinformatik, in: Wolfgang Uhr, Werner Esswein and Eric Schoop (Eds.), *6th International Conference Wirtschaftsinformatik*, Dresden, Germany, 947–967.

Gutzwiller, T. A. (1994), Das CC-RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen, Physica, Heidelberg.

Hafeez, K. (2002), Determining Key Capabilities of a Firm Using Analytic Hierarchy Process, *International Journal of Production Economics*, 76, 1, 39-51.

Harmsen, A. F. (1997), Situational Method Engineering, Moret Ernst & Young, Amsterdam.

Harmsen, F., Brinkkemper, S. and Oei, H. (1994), Situational Method Engineering for Information System Project Approaches, in: A.A. Verrijn-Stuart and T.W. Olle (Eds.), *Methods and Associated Tools for the Information Systems Life Cycle*, Elsevier Science B.V., Amsterdam, 169-194.

Henderson-Sellers, B. and Ralyté, J. (2010), Situational Method Engineering: State-of-the-Art Review, *Journal of Universal Computer Science*, 16, 3, 424-478.

Henderson, J. C. and Venkatraman, N. (1993), Strategic alignment: leveraging information technology for transforming organizations, *IBM Systems Journal*, 32, 1, 4-16.

Heuskel, D. (1999), Wettbewerb jenseits von Industriegrenzen: Aufbruch zu neuen Wachstumsstrategien, Campus, Frankfurt/Main.

Heutschi, R. (2007), Serviceorientierte Architektur, Springer, Berlin Heidelberg.

Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004), Design Science In Information Systems Research, *MIS Quarterly*, 28, 1, 75-105.

Heym, M. (1993), Methoden-Engineering - Spezifikation und Integration von Entwicklungsmethoden für Informationssysteme, Institut für Wirtschaftsinformatik, Universität St. Gallen, St. Gallen.

Hofstede, A. H. M. t. and Verhoef, T. F. (1997), On the feasibility of situational method engineering, *Information Systems*, 22, 401-422.

Inaganti, S. and Behara, G. K. (2007), Service Identification - BPM and SOA Handshake, *BPTrends*, 3, 1-12.

Josuttis, N. (2008), SOA in der Praxis - System-Design für verteilte Geschäftsprozesse, dpunkt.verlag, Heidelberg.

Kaplan, R. S. and Norton, D. P. (1992), The Balanced Scorecard - Measures that Drive Performance, *Harvard Business Review*, 70, 1, 71-79.

Karlsson, F. (2002), Meta-Method for Method Configuration - A Rational Unified Process Case, Linköping University, Faculty of Arts and Sciences, Linköping.

Karlsson, F. and Agerfalk, P. J. (2004), Method Configuration - Adapting to Situational Characteristics while Creating Reusable Assets, *Information & Software Technology*, 46, 9, 619-633.

Karlsson, F. and Agerfalk, P. J. (2007), Multi-Grounded Action Research in Method Engineering: The MMC Case, in: Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.), *Situational Method Engineering: Fundamentals and Experiences*, Springer, Boston, 19-32.

Kashanchi, R. and Toland, J. (2008), Investigating the Social Dimension of Alignment: Focusing on Communication and Knowledge Sharing, *19th Australasian Conference on Information Systems (ACIS)*, Christchurch, New Zealand, 462-472.

Kleinaltenkamp, M. (2000), Blueprinting – Grundlage des Managements von Dienstleistungsunternehmen, in: Herbert Woratschek (Ed.), *Neue Aspekte des Dienstleistungsmarketing - Ansatzpunkte für Forschung und Praxis*, Gabler, Wiesbaden, 3-28.

Klose, K., Knackstedt, R. and Beverungen, D. (2007), Identification of Services - A Stakeholder-Based Approach to SOA Development and Its Application in the Area of Production Planning, *15th European Conference on Information Systems (ECIS)*, St. Gallen, Switzerland, 1802-1814.

Kohlborn, T., Korthaus, A., Chan, T. and Rosemann, M. (2009a), Identification and Analysis of Business and Software Services - A Consolidated Approach, *IEEE Transactions on Services Computing*, 2, 1, 50-64.

Kohlborn, T., Korthaus, A., Chan, T. and Rosemann, M. (2009b), Service Analysis - A Critical Assessment of the State of the Art, *17th European Conference on Information Systems (ECIS)*, Verona, Italy.

Kohlmann, F. (2007), Service Modelling: A Hybrid Approach In Decomposed Financial Value Chains, *2nd International Workshop on Enterprise Modelling and Information Systems Architecture (EMISA)*, St. Goar, Germany, 205-218.

Kohlmann, F. and Alt, R. (2007), Business-Driven Service Modeling - A Methodological Approach from the Finance Industry, in: Witold Abramowicz and Leszek Maciaszek (Eds.), *Business Process & Service Computing: First International Working Conference on Business Process and Services Computing*, Gesellschaft für Informatik e.V., Leipzig, Germany, 180-193.

Kohlmann, F. and Alt, R. (2009a), Aligning Service Maps - A Methodological Approach From the Financial Industry, *42nd Hawaii International Conference on System Sciences (HICSS)*, Hawaii, 1-10.

Kohlmann, F. and Alt, R. (2009b), Deducing Service Ownerships in Financial Networks, *15th Americas Conference on Information Systems (AMCIS)*, San Francisco, CA, Paper 518.

Kohlmann, F., Börner, R. and Alt, R. (2010), Framework for the Design of Service Maps, *16th Americas Conference on Information Systems (AMCIS)*, Lima, Peru, Paper 39.

Lai, V. S. (1994), A Survey Of Rural Small Business Computer Use: Success Factors And Decision Support, *Information & Management*, 26, 6, 297-304.

Laukkanen, S., Sarpola, S. and Hallikainen, P. (2007), Enterprise size matters: objectives and constraints of ERP adoption, *Journal of Enterprise Information Management*, 20, 3, 319-334.

Legner, C. and Heutschi, R. (2007), SOA Adoption in Practice - Findings From Early SOA Implementations, in: Hubert Österle, Joachim Schelp and Robert Winter (Eds.), *15th European Conference on Information Systems (ECIS)*, St. Gallen, Switzerland, 1643-1654.

Leyer, M. and Moormann, J. (2010), Facilitating operational control of business services: A method for analysing and structuring customer integration, *21st Australasian Conference on Information Systems (ACIS)*, Brisbane, Australia, Paper 42.

Lovelock, C. H. and Wirtz, J. (2007), Service Marketing: People, Technology, Strategy, Pearson/Prentice Hall, Upper Saddle River, N.J.

Luthria, H. and Rabhi, F. A. (2009), Building the Business Case for SOA: A Study of the Business Drivers for Technology Infrastructure Supporting Financial Service Institutions, in: Dennis Kundisch, Daniel J. Veit, Tim Weitzel and Christoph Weinhardt (Eds.), *Enterprise Applications and Services in the Finance Industry*, Springer, 94-107.

Mahfouz, A., Barroca, L., Laney, R. and Nuseibeh, B. (2010), Requirements-Driven Design of Service-Oriented Interactions, *IEEE Software*, 27, 6, 25-32.

Moody, K. W. (2003), New Meaning to IT Alignment, *Information Systems Management*, Fall 2003, 30-35.

Myers, M. D. (2009), Qualitative Research in Business & Management, Sage Publications, London.

Nunamaker, J. F., Chen, M. and Purdin, T. D. M. (1991), Systems development in information systems research, *Journal of Management Information Systems*, 7, 3, 89-106.

Offermann, P. (2009), Eine Methode zur Konzeption betrieblicher Software mit einer serviceorientierten Architektur, Gito, Berlin.

Palvia, P., Mao, E., Salam, A. F. and Soliman, K. S. (2003), Management Information Systems Research: What´s There in a Methodology?, *Communications of the AIS*, 11, 1, 289-309.

Papazoglou, M. P. (2003), Service-Oriented Computing: Concepts, Characteristics and Directions, *4th International Conference on Web Information Systems Engineering*, IEEE Computer Society, Rome, Italy, 3-12.

Peffers, K., Tuunanen, T., Rothenberger, M. A. and Chatterjee, S. (2008), A Design Science Research Methodology for Information Systems Research, *Journal of Management Information Systems*, 24, 3, 45-77.

Pitts, R. A. and Lei, D. (2005), Strategic Management, South-Western, Mason, Ohio.

Prahalad, C. K. and Hamel, G. (1990), The Core Competence of the Corporation, *Harvard Business Review*, 68, 3, 79-91.

Punter, T. and Lemmen, K. (1996), The MEMA-model: towards a new approach for Method Engineering, *Information & Software Technology*, 38, 4, 295-305.

Ralyté, J. (2004), Towards Situational Methods for Information Systems Development: Engineering Reusable Method Chunks, *International Conference on Information Systems Development (ISD'04)*, Vilnius Technika, Vilnius, Lithuania, 271-282.

Ralyté, J. and Rolland, C. (2001), An Approach for Method Engineering, in: H.S. Kunii, S. Jajodia and A. Solvberg (Eds.), *20th International Conference on Conceptual Modelling*, Springer, Yokohama, Japan, 471-484.

Raymond, L. (1985), Organizational Characteristics And MIS Success In The Context Of Small Business, *MIS Quarterly*, 9, 1, 37-52.

Rolland, C., Plihon, V. and Ralyté, J. (1998), Specifying the Reuse Context of Scenario Method Chunks, in: B. Pernici and C. Thanos (Eds.), *10th International Conference on Advanced Information Systems Engineering (CAiSE)*, Springer, Pisa, Italy, 191-218.

Rolland, C. and Prakash, N. (1996), A proposal for context-specific method engineering, in: S. Brinkkemper, K. Lytinnen and R.J. Welke (Eds.), *IFIP International Federation for Information Processing TC8, Working Conference on Method Engineering*, Chapman&Hall, London, UK, 191-207.

Rosemann, M. and Vessey, I. (2008), Toward Improving the Relevance of Information Systems Research to Practice: The Role of Applicability Checks, *MIS Quarterly*, 32, 1, 1-22.

Saeki, M., Iguchi, K., Wen-yin, K. and Shinohara, M. (1993), A Meta-model for Representing Software Specification & Design Methods, *IFIP WG8.1 Conference on Information Systems Development Process*, Come, Italy, 149-166.

Schelp, J. and Aier, S. (2009), SOA and EA - Sustainable Contributions for Increasing Corporate Agility, in: Ralph H. Sprague (Ed.), *42nd Hawaii International Conference on System Sciences (HICSS)*, IEEE Computer Society, Hawaii, 1-8.

Schermerhorn, J. R. (1993), Management for Productivity, Wiley, New York, NY.

Sedera, D. (2008), Does Size Matter? Enterprise System Performance in Small, Medium and Large Organizations, *2nd Workshop on 3rd Generation Enterprise Resource Planning Systems*, Copenhagen, Denmark.

Sehmi, A. and Schwegler, B. (2006), Service-oriented modeling for connected systems - part 1, *The Architecture Journal*, 7, 33-41.

Shostack, G. L. (1982), How to Design a Service, *European Journal of Marketing*, 16, 1, 49-63.

Simon, H. A. (1996), The Sciences of the Artificial, MIT Press, Cambridge, MA London.

Stebbins, R. A. (2001), Exploratory Research in the Social Sciences, Sage Publications, Thousand Oaks.

Stewart, G. and Chakraborty, A. (2010), Service Identification through Value Chain Analysis and Prioritization, *16th Americas Conference on Information Systems (AMCIS)*, Lima, Peru, Paper 70.

Sunyaev, A., Hansen, M. and Krcmar, H. (2008), Method Engineering: A Formal Approach, *17th International Conference on Information Systems Development*, Paphos, Cyprus.

Terlouw, J., Terlouw, L. and Jansen, S. (2009), An Assessment Method for Selecting an SOA Delivery Strategy: Determining Influencing Factors and Their Value Weights, *4th International Workshop on Business/IT Alignment & Interoperability (BUSITAL)*, Amsterdam, The Netherlands.

Teubner, R. A. (1999), Organisations- und Informationssystemgestaltung - Theoretische Grundlagen und integrierte Methoden, DUV/Gabler, Wiesbaden.

van Slooten, K. and Hodes, B. (1996), Characterizing IS Development Projects, in: S. Brinkkemper, K. Lytinnen and R.J. Welke (Eds.), *IFIP International Federation for Information Processing TC8, Working Conference on Method Engineering*, Chapman&Hall, London, UK, 29-44.

Vergidis, K., Turner, C. J. and Tiwari, A. (2008), Business process perspectives. Theoretical developments vs. real-world practice, *International Journal of Production Economics*, 114, 1, 91-104.

Walls, J., Widmeyer, G. and El Sawy, O. (1992), Building an information system design theory for vigilant EIS, *Information Systems Research*, 3, 1, 36-59.

Welsh, J. A. and White, J. F. (1981), A small business is not a little big business, *Harvard Business Review*, 59, 4, 18-32.
Weske, M. (2007), Business Process Management - Concepts, Languages, Architectures, Springer, Berlin Heidelberg New York.
Winkler, V. (2007), Identifikation und Gestaltung von Services. Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich, *Wirtschaftsinformatik*, 49, 4, 257-266.
Zeithaml, V. A. and Bitner, M. J. (1996), Services Marketing, McGraw Hill, New York.
Zelewski, S. (2008), Theoretische Fundierung der Wirtschaftsinformatik, in: Reinhard Jung and Thomas Myrach (Eds.), *Quo vadis Wirtschaftsinformatik?*, Gabler, Wiesbaden, 163-190.

6.4 Fazit

Der konsolidierte Blick auf alle Artikel zeigt, wie Konzepte des Situational Method Engineering erfolgreich auf die Konstruktion von Serviceidentifikationsmethoden übertragen werden können. Ein besonderer Schwerpunkt lag dabei auf einer empirisch gestützten Ableitung von relevanten Kontextfaktoren. Aber auch der Entwicklung von Methodenfragmenten durch Techniken wie Dekomposition und Exploration wurde besondere Aufmerksamkeit geschenkt. Die Art und Weise der Konfiguration (so genannte Anpassungsmechanismen) auf Basis verschiedener Situationen wurde ebenfalls diskutiert.

Die Anwendung konnte beispielhaft für den Fall der efiport AG demonstriert werden, indem eine konkrete Methode zur Serviceidentifikation entwickelt wurde. Darüber hinaus konnte die konstruierte Methode dazu verwendet werden, tatsächliche Services, die im Falle einer Umsetzung des Paradigmas serviceorientierter Architekturen in efiport implementiert werden sollen, zu identifizieren.

Die vorgestellte Meta-Methode bietet an vielen Stellen Anknüpfungspunkte für weitere Forschungsvorhaben. Vielfältige Einschränkungen und weiterer Forschungsbedarf wurden in den einzelnen Artikeln schon mehrfach beschrieben. Der Aufbau eines Pools von validen Methodenfragmenten ist dabei eine wichtige Grundlage. Die größte Herausforderung für zukünftige Weiterentwicklungen wird darin bestehen, in Abhängigkeit von identifizierten Situationen konkrete Regeln für die Konfiguration von Methodenfragmenten zu formulieren. Diese könnten als Algorithmus implementiert werden. Dazu ist deutlich mehr Erfahrung im Umgang mit solchen Meta-Methoden in Serviceidentifikationsprojekten notwendig.

7. Literaturverzeichnis

Allen, P. (2006): *Service Orientation: Winning Strategies and Best Practices*, Cambridge: Cambridge University Press.

Arni-Bloch, N./Ralyté, J. (2008): *Service-Oriented Information Systems Engineering*, Konferenzpapier, 20th Conference on Advanced Information Systems Engineering (CAiSE), Montpellier, France.

Arsanjani, A./Ghosh, S./Allam, A./Abdollah, T./Ganapathy, S./Holley, K. (2008): SOMA: A method for developing service-oriented solutions, in: *IBM Systems Journal*, *47* (3), S. 377-396.

Atkins, C./Sampson, J. (2002): *Critical Appraisal Guidelines for SingleCase Study Research*, Konferenzpapier, European Conference on Information Systems, Gdansk, Poland.

Aydin, M.N. (2007): Examining Key Notions for Method Adaptation, in: Ralyté, J./Brinkkemper, S./Henderson-Sellers, B. (Hrsg.), *Situational Method Engineering: Fundamentals and Experiences*. Boston: Springer, S. 49-63.

Becker, J./Kugeler, M./Rosemann, M. (2008): *Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung*, 6., überarbeitete und erweiterte Aufl., Berlin Heidelberg: Springer.

Benbasat, I./Goldstein, D.K./Mead, M. (1987): The case research strategy in studies of information systems, in: *MIS Quarterly* (11), S. 269-386.

Bieberstein, N./Bose, S./Fiammante, M./Jones, K./Shah, R. (2006): *Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap*, Upper Saddle River, NJ: Prentice Hall.

Böhmann, T./Krcmar, H. (2005): Modularisierung: Grundlagen und Anwendung bei IT-Dienstleistungen, in: Herrmann, T./Kleinbeck, U./Krcmar, H. (Hrsg.), *Konzepte für das Service Engineering. Modularisierung, Prozessgestaltung und Produktivitätsmanagement*. Heidelberg: Physica, S. 45-83.

Brinkkemper, S. (1996): Method engineering: engineering of information systems development methods and tools, in: *Information & Software Technology*, *38* (4), S. 275-280.

Brooks, F.P. (1987): Essence and Accidents of Software Engineering, in: *IEEE Computer*, *20* (4), S. 10-19.

Brooksbank, R. (1991): Defining the small business: a new classification of company size, in: *Entrepreneurship and Regional Development*, *3* (1), S. 17-32.

Bucher, T./Klesse, M./Kurpjuweit, S./Winter, R. (2007): Situational Method Engineering - On the Differentiation of "Context" and "Project Type", in: Ralyté, J./Brinkkemper, S./Henderson-Sellers, B. (Hrsg.), *Situational Method Engineering: Fundamentals and Experiences*. Boston: Springer, S. 33-48.

Darke, P./Shanks, G./Broadbent, M. (1998): Successfully completing case study research: combining rigour, relevance and pragmatism, in: *Information Systems Journal*, *8* (4), S. 273-289.

Durst, M./Daum, M. (2007): Erfolgsfaktoren serviceorientierter Architekturen, in: *HMD - Praxis der Wirtschaftsinformatik*, *44* (253), S. 18-27.

Eckert, J./Repp, N./Schuller, D./Kiewning, L./Steinmetz, R. (2009): Implications of Service-oriented Architectures in the German Banking Industry - A Case Study, in: *efinancelab quarterly* (2), S. 4-5.

Erl, T. (2004): *Service-Oriented Architecture - A Field Guide to Integrating XML and Web Services*, Upper Saddle River, NJ: Prentice Hall.

Fettke, P. (2006): State of the Art des State of the Art, in: *Wirtschaftsinformatik*, *48* (4), S. 257–266.

Fitzgerald, B./Russo, N.L./O'Kane, T. (2003): Software Development: Method Tailoring at Motorola, in: *Communications of the ACM, 46* (4), S. 65-70.

Greiffenberg, S. (2003): Methoden als Theorien der Wirtschaftsinformatik. In Uhr, W./Esswein, W./Schoop, E. (Eds.), *6th International Conference Wirtschaftsinformatik* (pp. 947–967). Dresden, Germany.

Hagen, C. (2003): Integrationsarchitektur der Credit Suisse, in: Aier, S./Schoenherr, M. (Hrsg.), *Enterprise Application Integration - Flexibilisierung komplexer Unternehmensarchitekturen*. 1. Aufl., Berlin: Gito, S. 61-83.

Harmsen, F./Brinkkemper, S./Oei, H. (1994): Situational Method Engineering for Information System Project Approaches, in: Verrijn-Stuart, A.A./Olle, T.W. (Hrsg.), *Methods and Associated Tools for the Information Systems Life Cycle*. Amsterdam: Elsevier Science B.V., S. 169-194.

Hevner, A.R./March, S.T./Park, J./Ram, S. (2004): Design Science In Information Systems Research, in: *MIS Quarterly*, *28* (1), S. 75-105.

Inaganti, S./Behara, G.K. (2007): Service Identification - BPM and SOA Handshake, in: *BPTrends, 3*, S. 1-12.

Ivanov, K./Stähler, D. (2005): Prozessmanagement als Voraussetzung für SOA, in: *OBJEKTspektrum*, *12* (6), S. 60-64.

Janssen, M./Kuk, G. (2007): Comparing Coordination Arrangements Enabled by Web Services and Web Service Orchestration Technology, *The 15th European Conference on Information Systems* (pp. 1718-1728).

Jarke, M. (2009): Perspectives in the Interplay Between Business and Information Systems Engineering and Computer Science, in: *Business & Information Systems Engineering*, *1* (1), S. 70-74.

Josuttis, N. (2008): *SOA in der Praxis - System-Design für verteilte Geschäftsprozesse*, 1. Aufl., Heidelberg: dpunkt.verlag.

Klein, H.H./Myers, M.D. (1999): A set of principles for conducting and evaluating interpretive field studies in information systems, in: *MIS Quarterly*, *23* (1), S. 67-94.

Klose, K./Knackstedt, R./Beverungen, D. (2007): *Identification of Services - A Stakeholder-Based Approach to SOA Development and Its Application in the Area of Production Planning*, Konferenzpapier, 15th European Conference on Information Systems (ECIS), St. Gallen, Switzerland.

Kohlmann, F./Alt, R. (2007): *Business-Driven Service Modeling - A Methodological Approach from the Finance Industry*, Konferenzpapier, Business Process & Service Computing: First International Working Conference on Business Process and Services Computing, Leipzig, Germany.

Lee, A.S. (1989): A scientific methodology for MIS case studies, in: *MIS Quarterly*, *13* (1), S. 33-50.

Legner, C./Heutschi, R. (2007): *SOA Adoption in Practice - Findings From Early SOA Implementations*, Konferenzpapier, 15th European Conference on Information Systems (ECIS), St. Gallen, Switzerland.

Lovelock, C.H./Wirtz, J. (2007): *Service Marketing: People, Technology, Strategy*, 6th Aufl., Upper Saddle River, N.J.: Pearson/Prentice Hall.

Luthria, H./Rabhi, F.A. (2009): Building the Business Case for SOA: A Study of the Business Drivers for Technology Infrastructure Supporting Financial Service Institutions, in:

Kundisch, D./Veit, D.J./Weitzel, T./Weinhardt, C. (Hrsg.), *Enterprise Applications and Services in the Finance Industry*. Springer, S. 94-107.

Mingers, J. (2001): Combining IS Research Methods: Towards a Pluralist Methodology, in: *Information Systems Research*, *12* (3), S. 240-259.

Nadhan, E.G. (2004): Seven Steps to a Service-Oriented Evolution, in: *Business Integration Journal* (1), S. 41-44.

Papazoglou, M.P. (2003): *Service-Oriented Computing: Concepts, Characteristics and Directions*, Konferenzpapier, 4th International Conference on Web Information Systems Engineering, Rome, Italy.

Peffers, K./Tuunanen, T./Rothenberger, M.A./Chatterjee, S. (2008): A Design Science Research Methodology for Information Systems Research, in: *Journal of Management Information Systems*, *24* (3), S. 45-77.

Rabhi, F.A./Yu, H./Dabous, F.T./Wu, S.Y. (2006): A service-oriented architecture for financial business processes: A case study in trading strategy simulation, in: *Information Systems and e-Business Management*, *5* (2), S. 185-200.

Ralyté, J. (2004): *Towards Situational Methods for Information Systems Development: Engineering Reusable Method Chunks*, Konferenzpapier, International Conference on Information Systems Development (ISD'04), Vilnius, Lithuania.

Ralyté, J./Rolland, C. (2001): *An Approach for Method Engineering*, Konferenzpapier, 20th International Conference on Conceptual Modelling, Yokohama, Japan.

Reinheimer, S./Lang, F./Purucker, J./Brügmann, H. (2007): 10 Antworten zu SOA, in: *HMD - Praxis der Wirtschaftsinformatik*, *44* (253), S. 7-17.

Simon, H.A. (1996): *The Sciences of the Artificial*, 3. Aufl., Cambridge, MA London: MIT Press.

Starke, G./Tilkov, S. (2007): *SOA-Expertenwissen*, 1. Aufl., Heidelberg: dpunkt.verlag.

Stebbins, R.A. (2001): *Exploratory Research in the Social Sciences*, Thousand Oaks: Sage Publications.

Strauss, A./Corbin, J.M. (1990): *Basics of qualitative research: Grounded theory procedures and techniques*, Thousand Oaks, CA: Sage Publications.

Thomas, O./Leyking, K./Dreifus, F. (2007): Prozessmodellierung im Kontext serviceorientierter Architekturen, in: *HMD - Praxis der Wirtschaftsinformatik*, *44* (253), S. 37-46.

Walsham, G. (1995): Interpretive case studies in IS research. Nature and Method, in: *European Journal of Information Systems* (4), S. 74-81.

Weske, M. (2007): *Business Process Management - Concepts, Languages, Architectures*, Berlin Heidelberg New York: Springer.

Wilde, T./Hess, T. (2007): Forschungsmethoden der Wirtschaftsinformatik - Eine empirische Untersuchung, in: *Wirtschaftsinformatik*, *49* (4), S. 280-287.

Winkler, V. (2007): Identifikation und Gestaltung von Services. Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich, in: *Wirtschaftsinformatik*, *49* (4), S. 257-266.

Yin, R.K. (2003): *Case Study Research - Design and Methods*, 3 Aufl., Thousand Oaks London New Delhi: SAGE Publications.

Zacharias, R. (2005): Serviceorientierung: Der OO-König ist tot, es lebe der SOA-König, in: *OBJEKTspektrum*, *12* (2), S. 43-52.

Eidesstattliche Versicherung

Ich versichere an Eides Statt, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Für alle Inhalte, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Quellen gleich welcher Art entnommen sind, habe ich die fremde Urheberschaft kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form weder im Inland noch im Ausland als Prüfungsarbeit eingereicht worden. Sie ist auch noch nicht veröffentlicht.

Frankfurt am Main, 04. Mai 2012

René Börner