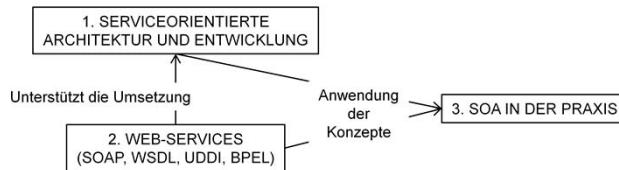


SERVICEORIENTIERTE ARCHITEKTUREN – Überblick über die Kurseinheiten



- (1) SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG behandeln übergeordnete Konzepte und die SOA-relevanten Standards zur Modellierung von Geschäftsprozessen und -services
- (2) WEB-SERVICES sind die heute bevorzugte Technologie zur Umsetzung von SOA-Konzepten
- (3) SOA IN DER PRAXIS ist ein (meist externer) Beitrag, der zeigt, wie Serviceorientierung praktisch angewendet werden kann

1 11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

Der WASA-Teil SERVICEORIENTIERTE ARCHITEKTUREN umfasst das gesamte Gebiet der Serviceorientierten Software-Anwendungen und als ein Bestandteil die Web-Services, die eine zentrale Technologie zur Implementierung solcher "fortgeschrittenen Web-Anwendungen" (-> Advanced Web Applications) darstellen.

Der Aufbau ist analog zu dem WASA-TEIL "WEB-ANWENDUNGEN", in denen anstelle der serviceorientierten (fortgeschrittenen) Konzepte die traditionellen Architektur- und Entwicklungskonzepte behandelt werden.

- (1) Es wird ein methodisches Vorgehen zur Entwicklung von serviceorientierten Systemen beschrieben, bei dem standardisierte Sprachen zur Modellierung von Geschäftsprozessen und -services gezielt genutzt werden.
- (2) Der Kern besteht aus Standards zur Kommunikation, Servicebeschreibung, Service-Registrierung und zur Komposition von Web-Services, wodurch eine Abbildung und Unterstützung von Geschäftsprozessen ermöglicht.
- (3) Diese Kurseinheit wird laufend gegen aktuell behandelte (wissenschaftliche bzw. projektbezogene) Themen ausgetauscht.

WASA

Web-Anwendungen und Service-orientierte Architekturen

SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG – Lernziele



(1) SERVICEORIENTIERUNG

Das Prinzip der Serviceorientierung und die damit verfolgten Ziele werden verstanden

(2) ARCHITEKTUR

Die Struktur und wesentlichen Bausteine einer serviceorientierten Architektur sind bekannt

(3) ENTWICKLUNG

Die Phasen des Entwicklungsprozesses für serviceorientierte Software werden verstanden und die in der Geschäftsanalyse und im Servicedesign auftretenden Artefakte können am Beispiel des KITCampusGuide nachvollzogen werden

(1) Dieses Lernziel stellt den serviceorientierten Ansatz in einen größeren geschäftsorientierten Kontext. Neben den wesentlichen Merkmalen der Serviceorientierung wird das in Quasar Enterprise verfolgte Vorgehen zur Einführung einer Serviceorientierung in einem Unternehmen skizziert.

(2) In diesem Lernziel werden die Grundlagen dafür gelegt, die Serviceorientierung zur Weiterentwicklung bestehender Softwarearchitekturen zu nutzen.

(3) Der serviceorientierte Entwicklungsprozess stellt eine evolutionäre Weiterentwicklung des traditionellen Prozesse dar und führt aufgrund der Ausrichtung auf das Geschäft neue Artefakte und dazugehörige Beschreibungssprachen ein. Diese Artefakte sind

- (i) in der Analyse: UML-basierte Geschäftsservices, BPMN-basierte Geschäftsprozesse
- (ii) im Design: SoaML-basierte Servicebeschreibungen

BPMN Business Process Model and Notation

SoaML Service oriented architecture Modeling Language

Hauptquellen

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

[EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.

[EK+08] Thomas Erl, Anish Karmarkar et al.: Web Service Contract Design and Versioning or SOA, Prentice Hall, 2008.

Services als Ergebnis einer evolutionären Entwicklung der Informationstechnologie

- (1) Seit Mitte des 20. Jahrhunderts ist der Trend hin zur Zerlegung und Verteilung in der Informationstechnologie klar erkennbar
- (2) Programme entwickelten sich von wenigen großen, inhärent komplexen Software-Bestandteilen zu zahlreichen kleinen, überschaubaren Codestücken
- (3) Ein Meilenstein in der IT-Entwicklung war die Ausbreitung des Internet Ende des 20. Jahrhunderts
- (4) Die Akzeptanz der serviceorientierten Architekturen (Service Oriented Architecture, SOA) kann als ein mit dem Internet vergleichbarer Entwicklungsschritt auf der Ebene der Software angesehen werden
- (5) Der SOA liegt die Idee eines Marktplatzes zugrunde
- (6) In einer SOA sind die serviceerbringenden Software-Komponenten lose gekoppelt

3

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[ORCHESTRA08:16]

(1) Dieser Trend gilt sowohl für die Hardware (immer kleinere und leistungsfähigere Rechner bei fallenden Preisen) als auch für die Software.

(2) Die Software hat einen ähnlichen Weg der Spezialisierung beschritten wie zuvor die insbesondere in größeren Organisationen arbeitenden Menschen.

(3) Das Internet bietet ein für jedermann zugängliches Rechnernetz, über das jegliche Form von Information erzeugt, gespeichert und ständig verändert werden kann.

(4) Die SOA setzt auf den zuvor entwickelten Modularisierungs- und OO-Paradigmen zur Zerlegung und Vereinfachung von Code auf. Eine Anwendung setzt sich in der SOA nicht aus einzelnen Codestücken zusammen, sondern aus atomaren Anwendungen, die spezifische Services erbringen. Als Vorstufen der SOA können gesehen werden [DJ+05:2]:

(i) Assembler, (ii) Remote Procedure Call (mit der Einführung von Netzwerken), (iii) CORBA (aufgrund neuer von RMI nicht erfüllter Anforderungen wie z.B. Typprüfung)

(5) Die Ideen sind:

- Bereitsteller -> Service (Funktionalität) -> Konsument
- Anfrage des Services resultiert in einer Antwort
- Anwendung kann in eine Interaktion von einer Menge von Services zerlegt werden
- Neben den Services ist auch die zugrundeliegende Infrastruktur relevant

(6) Das Prinzip der losen Kopplung ist in der SOA von zentraler Bedeutung und besagt, dass die Abhängigkeit zwischen einem Service und dem Gesamtsystem möglichst gering gehalten werden sollen. Das Ziel ist, einen Service möglichst einfach aus dem System herauslösen zu können, um ihn z.B. gegen einen anderen ersetzen zu können. Das wird durch die folgenden Konzepte erreicht:

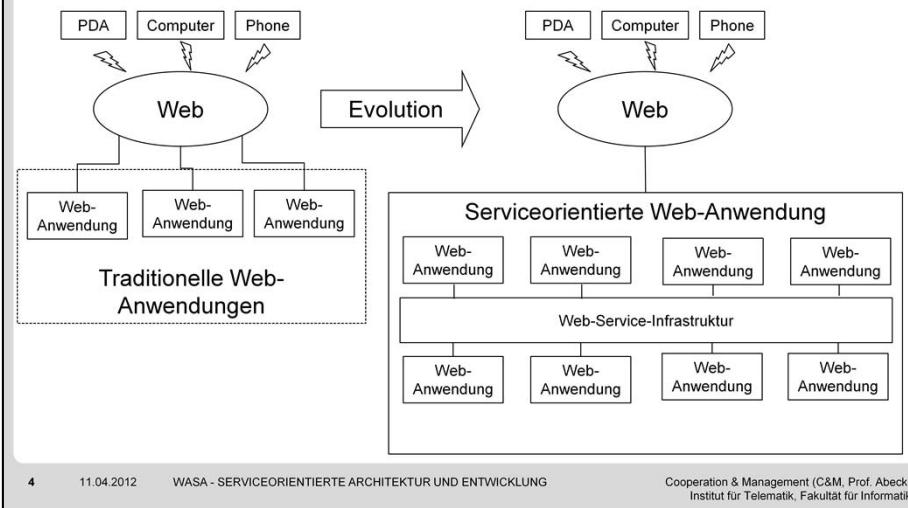
- Möglichst hohe Abgeschlossenheit (engl. self-contained)
- Standardisierte, klar formulierte Serviceschnittstellen, die keine Implementierungsdetails beinhalten (Black-Box-Sicht)
- Zustandslosigkeit eines Services
- Auffinden eines Services über einen Service Broker
- Asynchrone Kommunikation

[DJ+05] Wolfgang Dostal, Mario Jaeckle, Ingo Melzer, Barbara Zengler: Service-orientierte Architekturen mit Web Services, Elsevier Spektrum Akademischer Verlag, 2005.

[ORCHESTRA08] The ORCHESTRA Consortium: orchestra – an open architecture for risk management, www.eu-ORCHESTRA.org, 2008.

Evolution von Web-Anwendungen

- (1) Es lassen sich die folgenden zwei Arten von Web-Anwendungen unterscheiden



4

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[DM+07:114]

- (1) Die zwei Arten von Web-Anwendungen sind:
- (i) Traditionelle Web-Anwendungen (auch als Anwendungsinfrastrukturen bezeichnet [:113])
 - (ii) Serviceorientierte Web-Anwendungen (Serviceinfrastrukturen).

(Traditionelle Web-Anwendungen) Das Web ist das bevorzugte Medium bzw. die Plattform für die Kommunikation und Kollaboration zwischen Menschen und Anwendungen [:113].

(Serviceorientierte Web-Anwendung) Web-Systeme werden zusammengesetzt aus Services, die wohldefinierte Schnittstellen zur Anwendung-zu-Anwendung- und Benutzer-zu-Anwendung-Kommunikation ausstellen (engl. expose). Die Services werden als Fähigkeiten von Servicegebern bereitgestellt und lassen sich zu neuen Fähigkeiten komponieren [:115].

PDA Personal Digital Assistant

[DM+07] Naci Dai, Lawrence Mandel, Arthur Ryman: Eclipse Web Tools Platform: Developing Java Web Applications, Addison-Wesley, 2007.

Merkmale und Ziele der Serviceorientierung

- (1) Lose Kopplung
 - (1) Ziel ist das einfache Herauslösen und Ersetzen eines Services zur Laufzeit
 - (2) Dynamisches Binden wird durch das Serviceverzeichnis als zentralem Bestandteil des Servicemodells ermöglicht
- (2) Unterstützung von Geschäftsprozessen
 - (1) Services kapseln geschäftsrelevante Funktionalität
 - (2) Komposition von Services
 - (3) Prozessorientierte Anwendungsintegration
- (3) Verwendung von Standards
 - (1) Programmiersprachen- und Plattform-unabhängige Bereitstellung der Services und Servicekompositionen
 - (2) Das SOA-Paradigma wurde insbesondere durch die Web-Service-Standards vorangetrieben

(1) Kopplung ist eine strukturelle Eigenschaft eines Artefakts.

Im Fall einer Softwarearchitektur bezeichnet Kopplung den Grad, zu dem zwei Module unabhängig sind. Dies bezieht sich üblicherweise auf Module jeweils paarweise, wobei sich die Kopplung des Systems insgesamt aus der Kopplung der möglichen Paare ergibt [C&M-DA-Oe10].

(1.1) In einer SOA bezieht sich die lose Kopplung auf die auftretenden Services und deren Abhängigkeiten: Je geringer die Abhängigkeiten, d.h. je loser die Services gekoppelt sind, umso einfacher gestaltet sich die Umgestaltung der SOA durch Austausch oder Ergänzung von Services.

(1.2) In [DJ+05:9] wird lose Kopplung ausschließlich auf das dynamische Binden bezogen. Das ist zwar ein wesentlicher, aber sicherlich nicht der ausschließliche Aspekt der losen Kopplung. Weitere Aspekte, die zu einer losen Kopplung beitragen sind: (i) Zustandslosigkeit der Services, (ii) Asynchrone Kommunikation, d.h. kein Warten auf die Antwort, (iii) Inhaltliche Abgeschlossenheit, was zu einer höheren Wiederverwendbarkeit beiträgt.

(2) Ein wesentliches mit einer SOA verknüpftes Ziel besteht in der Ausrichtung der IT auf das Geschäft und die zu unterstützenden Geschäftsprozesse (sog. Business-IT Alignment).

(2.1) Aufgrund der Geschäftsorientierung sind Services und die von diesen bereitgestellten Operationen (z.B. im Vergleich zu Objektmethoden) eher grobgranular, d.h. umfassen eine komplexere und umfangreichere Funktionalität, die von einem Geschäftsanalysten verstanden wird.

(2.2) Kann als eine "imperative Programmierung im Großen" aufgefasst werden, wobei die einzelnen Ausführungsschritte die Serviceoperationen darstellen. Ziel ist die teilautomatisierte Abbildung der als Flussdiagramme vorliegenden Geschäftsprozesse auf Servicekompositionen, die auf der SOA ausführbar sind.

(2.3) Da die in der SOA komponierten Services meist von (bestehenden, engl. legacy) Anwendungen bereitgestellt werden, erfolgt eine Anwendungsintegration entlang der unterstützenden Geschäftsprozesse. SOA kann daher insbesondere als eine neue Form von Integrationsarchitektur angesehen werden, die die bestehenden Ansätze des "Enterprise Application Integration" (EAI) ergänzen oder sogar vollständig ersetzen wird.

(3.1) Die Umsetzung der SOA-Idee setzt eine maschinenlesbare Beschreibung der Services voraus.

(3.2) Die Grundlage für alle WS-* Standards liefert XML, eine standardisierte Datenbeschreibungssprache, aus der durch den Erweiterungsmechanismus beliebige Sprachen (u.a. Programmiersprachen) entwickelt werden können.

[C&M-DA-Oe10] Stefan Oehlert: Metriken zur Qualitätsanalyse dienstorientierter Architekturen, Diplomarbeit, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2010.

[DJ+05] Wolfgang Dostal, Mario Jaeckle, Ingo Melzer, Barbara Zengler: Service-orientierte Architekturen mit Web Services, Elsevier Spektrum Akademischer Verlag, 2005.

LZ SERVICEORIENTIERUNG – ÜA MERKMALE

- (1) Was zeichnet den serviceorientierten Ansatz gegenüber einem traditionellen (komponenten-/objektorientierten) Ansatz aus?
 - (1) Das Black-Box-Prinzip eines Services
 - (2) Der Geschäftsbezug der durch einen Service erbrachten Funktionalität
 - (3) Die Unterstützung von Geschäftsprozessen
 - (4) Die Forderung nach Wiederverwendung von Funktionalität
 - (5) Die Suche von Services in Verzeichnissen und deren Einbindung zur Laufzeit

Auswahl relevanter Arbeiten zum Thema der Serviceorientierung

- (1) The Prentice Hall Service-Oriented Computing Series von Thomas Erl
 - (1) Relativ abstrakt gehaltene Vorgaben zum Aufbau einer SOA
 - (2) Bücher u.a. zu Web Service Contracts und zu Servicedesign-Mustern
- (2) Service-oriented Analysis and Design (SOAD) von IBM
 - (1) Modellierung auf der Basis von UML bilden einen Schwerpunkt
 - (2) Führender Ansatz hinsichtlich einer werkzeuggestützten Modellierung und Entwicklung von serviceorientierter Software
- (3) Quasar Enterprise von sd&m
 - (1) Ein aus zahlreichen SOA-Projekten abgeleitetes methodisches Vorgehen
 - (2) Ein umfassendes Buch und zahlreiche Einzelpublikationen

(1) Die Serie besteht aus momentan fünf Büchern.

(1.1) Wenig konkrete Hilfestellungen zur Lösung der wichtigen SOA-Problemstellungen.

(1.2) [EK+08] gibt konkrete und hilfreiche Beschreibung zur Nutzung von Web-Services und der damit verbundenen XML-Technologie.

[Er08] beschreibt die strukturellen Anforderungen und Eigenschaften auf einem durchgängigen, aber eher informellen Abstraktionsniveau.

(2) Hierzu zählen insbesondere die (relativ früh begonnenen) Arbeiten von Arsanjani ([Ar04], [AG+08]).

(2.1) Beiträge zum Servicemodell und zum jetzigen SoaML. Die Modelle werden mit dem Rational-Modellierungswerkzeug erstellt [BK+05].

(2.2) Die Eignung der Werkzeuglösungen auch für große Projekte erscheint unklar. Das übliche (Spiel-) Beispiel ist der Bestellprozess (engl. purchase order), der die Fertigung (engl. production scheduling) und die Auslieferung (engl. shipping) umfasst.

(3) Das Konzept heißt Quasar Enterprise.

(3.1) Prozessorientierte Integration einer bestehenden Anwendungslandschaft mit einer Fokussierung von IT-organisatorischen Themen.

(3.2) Die nachfolgend beschriebenen Konzepte und Methoden orientieren sich weitestgehend an dieser Quelle.

SOAD	Service Oriented Analysis and Design (IBM)
SoaML	Service oriented archietcture Modeling Language

[AG+08] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy, K. Holley: SOMA: A method for developing service-oriented solutions, IBM Systems Journal, VOL 47, NO 3, 2008.

[Ar04] A. Arsanjani: Service-Oriented Modeling and Architecture, IBM developer works, 2004.

[BK+05] Alan W. Brown, Simon K. Johnston, Grant Larsen, Jim Palistrant: SOA Development Using the IBM Rational Software Development Platform: A Practical Guide, 2005.

[EH+08] G. Engels, A. Hess, B. Humm, O. Juwig, M. Lohmann, J.-P. Richter, M. Voß, J. Willkomm: A Method for Engineering a true SOA, 2008.

[EK+08] Thomas Erl, Anish Karmarkar et al.: Web Service Contract Design and Versioning of SOA, Prentice Hall, 2008.

[Er08] Thomas Erl: SOA Design Patterns, Prentice Hall, 2008.

Sichten auf die Serviceorientierung

- (1) Serviceorientierung aus Geschäftssicht
 - (1) Sicht der Unternehmensberater
 - (2) Servicedefinition: Geschäftsservices ohne IT-Bezug
- (2) Serviceorientierung aus IT-Sicht
 - (1) Sicht der SOA-Produkthersteller
 - (2) Servicedefinition: Softwaretechnischer Service, meist Web-Service
- (3) SOA hat das Ziel der Verbindung von Geschäft und IT
 - (1) Sicht der IT-Unternehmensarchitekten
 - (2) Servicedefinition: Mittel zur Ausrichtung der Anwendungslandschaft auf das Geschäft

8

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[EH+08:91]

- (1) In diesem Fall bilden die Services die Grundlage für die Strukturierung des Geschäfts.
 - (1.1) Den Unternehmensberater interessieren die von einem Unternehmen verfolgten Geschäftsziele und nicht die technische Unterstützung des Geschäfts im Vordergrund.
 - (1.2) Ein Geschäftsservice ist eine geschäftliche Leistung, zu der ein Servicegeber und ein Servicenehmer existiert.
- (2.1) Diese Sicht wird vom reinen Softwarearchitekt und Softwareentwickler eingenommen.

(3) SOA als die Brücke zwischen Geschäft und IT

- (3.1) Ein IT-Unternehmensarchitekt muss die Geschäftsarchitektur mit der IT-Architektur (Anwendungslandschaft) verbinden können.
- (3.2) Service als geschäftsorientiertes Integrationsartefakt

[EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.

Serviceorientierte Gestaltung des Geschäfts (gemäß Quasar Enterprise)



(1) Ziel

- (1) Flexible Anpassung der Unternehmensstrukturen und Geschäftsprozesse an sich schnell wandelnde Marktanforderungen

(2) Vorgehen

- (1) Strukturierung des Geschäfts in Geschäftsservices
 - (1) In sich abgeschlossene Bausteine geschäftlicher Leistungen
- (2) Variable Kombination der Geschäftsservices zu neuen Geschäftsprozessen
- (3) Die Serviceorientierung ist in einem Unternehmen zu entwickeln
 - (1) Gegenüber dem Kunden erbrachte Services
 - (2) Von Abteilungen innerhalb der Organisation bereitgestellte
 - (3) Von außen bezogene Services

9

11.04.2012 WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[EH+08:91]

Zu den hier betrachteten Unternehmen, für die die nachfolgende Methode genutzt werden kann, zählen z.B. Banken und Versicherungen, Groß- und Einzelhandel oder auch Bildungsanbieter wie das KIT.

(1.1) Insbesondere reicht es heute aufgrund der erhöhten Dynamik nicht, die Geschäftsprozesse nur einmal zu optimieren.

(2.1) Diese Bausteine werden als Geschäftsservices bezeichnet und im Folgenden noch genauer definiert.

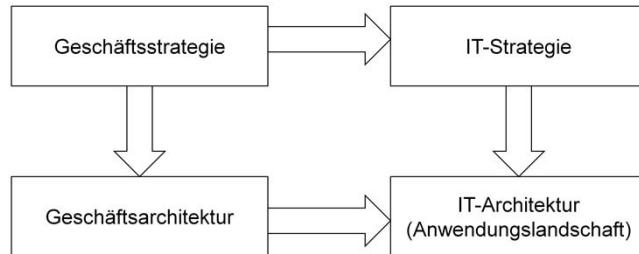
(2.2) Die Geschäftsservices werden entlang der Prozesse, die zu unterstützen sind, zusammengeschaltet bzw. komponiert.

(3) Das heißt, es sind die für die Erbringung des Geschäfts geeigneten Services zu ermitteln. Wünschenswert ist, dass diese Services so gestaltet werden, dass sie zu größeren Services kombiniert werden können.

[EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.

Geschäftsanalyse

- (1) Ermittlung der Geschäftsarchitektur des Unternehmens
 - (1) Umfasst u.a. Geschäftsservices, Geschäftsprozesse, Geschäftsobjekte, Organisationseinheiten
 - (2) Ableitung aus den geschäftlichen Zielen und Anforderungen
 - (3) Verwendung von Architekturleitlinien



10

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[EH+08:113]

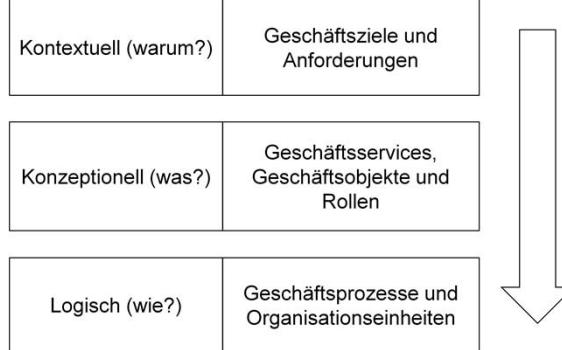
Diese Phase orientiert sich an dem in Quasar Enterprise verfolgten methodischen Vorgehen, das in der Geschäftsanalyse-Phase die Ermittlung der Geschäftsarchitektur zum Ziel hat.

- (1) Aus der Geschäftsarchitektur lassen sich von IT-Unternehmensarchitekten wesentliche Elemente der Anwendungslandschaft (IT-Architektur) ableiten.
- (1.2) Diese sind Teile der Geschäftsstrategie und liefern Hinweise darauf, was passende Geschäftsservices (sowie zugehörige Geschäftsobjekte und an den Services beteiligte Rollen) sind.
- (1.3) Architekturleitlinien (engl. architectural guidelines) sind Regeln, die aus dem Kontext der Geschäftsarchitektur abgeleitet sind und beim Design der IT-Architektur zu berücksichtigen sind.

(Pfeile) Illustrieren, welche Entwurfsergebnisse in die Erstellung der anderen Elemente einfließen.

[EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.

Aufstellen der Geschäftsarchitektur



- (1) Das Geschäft und die damit verfolgten Ziele liefern den Kontext für die Geschäftsarchitektur
- (2) Geschäftsservices beschreiben, was das Unternehmen leisten muss
- (3) Ein Geschäftsprozess erbringt einen Geschäftsservice

[EH+08:115]

- (1) Auf der kontextuellen Ebene werden aus den Geschäftszielen und Anforderungen (-> Geschäftsstrategie) die Architekturerlinien abgeleitet.
- (2) Die Geschäftsservices sind eine Art Bindeglied oder Dreh- und Angelpunkt zwischen Geschäft und IT:
 - Sie lassen sich unmittelbar aus der Geschäftsstrategie ableiten
 - Sie dienen zur Identifikation von Anwendungsservices (die Softwareservices der SOA).
- (3) Der Geschäftsprozess beantwortet die Frage, WIE der Geschäftsservice zu erbringen ist.

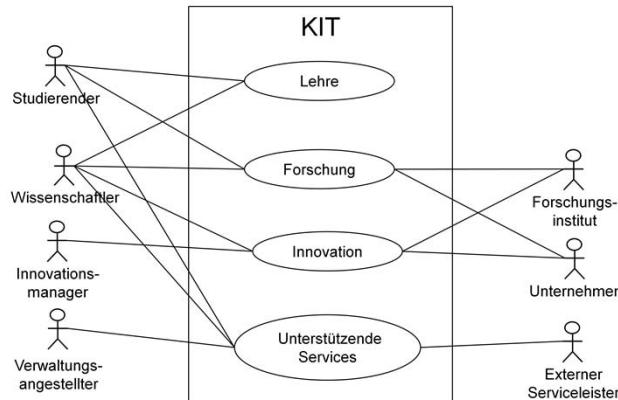
[EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.

- (1) Ausgangspunkt der Entwicklung einer SOA
- (2) Eine von einem Servicegeber erbrachte geschäftliche Leistung
- (3) Besondere Relevanz bei SOA-Problemen mit hoher geschäftlich-organisatorischer Dimension
- (4) Ein Geschäftsservice ist eine abgeschlossene, lose gekoppelte, funktionale Fähigkeit
- (5) Zur Modellierung von Geschäftsservices bietet sich das UML-Anwendungsdiagramm an
 - (1) Alternativ können Geschäftsanwendungsfälle genutzt werden

- (1) Die Geschäftsservices bilden die statische Struktur für die nachfolgenden Geschäftsprozesse.
- (2) Ein Geschäftsservice stellt eine geschäftliche Leistung dar, die ein Servicegeber einem Servicenehmer bereitstellt. Der Servicenehmer nutzt einen Geschäftsservice im Rahmen der Durchführung einer Geschäftsaktivität.
- (3) Das ist z.B. gegeben, wenn die Geschäftsservices eines ganzen Unternehmens nach einer bestimmten Systematik ermittelt werden sollen. Ein Beispiel einer solchen Systematik liefert z.B. [EH+08].
- (4) Eine grobgranulare Funktionalität die nach innen eine hohe Kohärenz und mit anderen Geschäftsservices eine geringe Kohärenz aufweist.
- (5) Dieses Vorgehen wird in [EH+08] genutzt.
- (5.1) Geschäftsanwendungsfälle werden in der auf RUP/SOMA (Rational Unified Process for Service-Oriented Modeling and Architecture) basierenden Vorgehen genutzt, das in [Jo04] beschrieben und in [C&M-Di-Ge11] weiter ausgeführt wurde.
- [C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.
- [EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.
- [Jo04] Simon Johnston: Rational UML Profile for Business Modeling, IBM Rational Software, 2004.

KCG: Geschäftsservices des KIT

- (1) Das Geschäft des Karlsruher Institut für Technologie (KIT) betrifft die Lehre, die Forschung und die Innovation
 - (1) KITCampusGuide (KCG) unterstützt einen Teil dieses Geschäfts



13 11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

- (1) Die vom KIT behandelten Inhalte betreffen im Schwerpunkt die Ingenieur- und Naturwissenschaften.

Das KIT tritt als ein Servicegeber gegenüber seinen Studierenden auf.

- (1.1) Die durch den KCG erbrachte Funktionalität lässt sich in die Gruppe der Unterstützenden Services einordnen.

In Anlehnung an das Vorgehen in [EH+08:22] wird ein UML-Anwendungsfalldiagramm dazu genutzt, die Services des KIT und die beteiligten Rollen zu skizzieren.

Die Anwendungsfälle repräsentieren in diesem Fall die Geschäftsservices. Die Rollen auf der linken Seite sind Mitglieder des KIT, während die Rollen auf der rechten Seite außerhalb des KIT stehen.

(Studierender, Forschung) Ein Studierender trägt durch gewisse der während seines Studiums erbrachten Leistungen (z.B. Masterarbeit) zur Forschung bei.

(Wissenschaftler, Lehre) In diesem Geschäftsservice tritt der Wissenschaftler als Dozent auf.

(Studierender, Unterstützende Services) Viele der Services werden in Form von sog. Selbstbedienungsservices online angeboten (hierzu zählt u.a. auch der durch den KCG bereitgestellte Service).

(Forschungsinstitut, Unternehmen) Ein Forschungsinstitut ist z.B. eine andere Universität oder eine (überwiegend) durch staatliche Gelder finanzierte Organisation. Diese Rollen könnten auch zusammengefasst werden zu einer Rolle "Externer Partner".

Die hier am Beispiel des KIT erfolgte Identifikation von Geschäftsservices ist der erste Schritt einer nachfolgend vorgestellten Methodik zur Identifikation und Verfeinerung von Geschäftsservices.

[EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.

Methode zur Identifikation und Verfeinerung von Geschäftsservices



- (1) Die Methode geht von den Geschäftszielen aus und besteht aus den folgenden Schritten
 - (1) Kerngeschäftsservices und unterstützende Geschäftsservices identifizieren
 - (2) Serviceausschnitt festlegen und funktional verfeinern
 - (3) Elementare Geschäftsservices fixieren
 - (4) Zugehörige Geschäftsobjekte bestimmen
- (2) Bei Vorliegen eines vollständigen Prozessmodells können die Geschäftsservices durch geeignete Gruppierung daraus abgeleitet werden

14

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[EH+08:127]

(1.1) Die in diesem Schritt identifizierten Geschäftsservices gehören zur obersten und damit zur Ebene 1.

(1.2) Auswahl derjenigen Geschäftsservices, zu denen eine funktionale Dekomposition vorgenommen wird.

(1.3) Diese Festlegung erfolgt anhand der Ziele und Rollen (weil elementar heißt, dass der Service von nur einer Rolle erbracht wird).

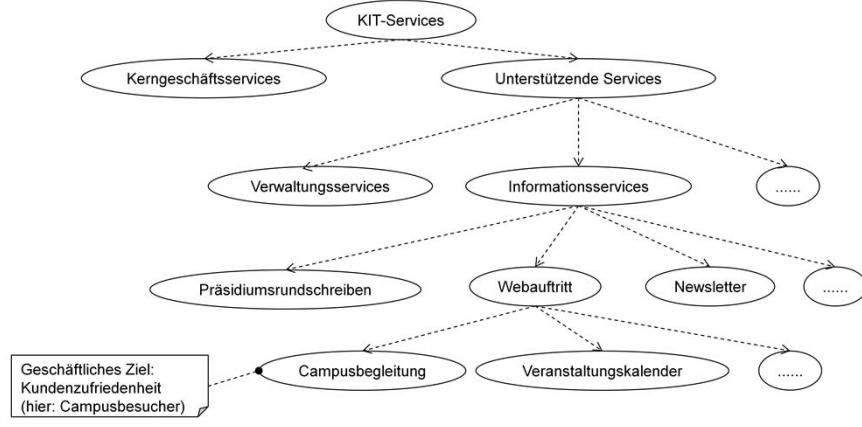
(1.4) Beschreibung der Interaktion zwischen benötigter bzw. gelieferter Geschäftsobjekte

(2) Die Methode wird angewendet, wenn noch keine solche Vorarbeit geleistet wurde. Die Geschäftsprozesse werden in diesem Fall dadurch ermittelt, dass beschrieben wird, WIE die Geschäftsservices realisiert werden.

[EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.

KCG: Serviceausschnitt festlegen

- (1) Die mit dem KCG vom KIT angebotenen Geschäftsservices gehören zu den Informationsservices, speziell zum Webauftritt



Der zweite Schritt der Methode zur Identifikation und Verfeinerung von Geschäftsservices soll am Beispiel des KITCampusGuide (KCG) aufgezeigt werden.

(1) Die Informationsservices sind - wie auch die Verwaltungsservices oder die Lehre-bezogenen, Forschungs-bezogenen und Innovations-bezogenen Services - Geschäftsservices der Ebene 1.

(Kerngeschäftsservices) Können weiter in die Bereiche Lehre-bezogene, Forschungs-bezogene und Innovations-bezogene Services aufgeteilt werden.

(Verwaltungsservices) Hierzu gehören u.a. Rechnungswesen, Berichtswesen, Personalwesen.

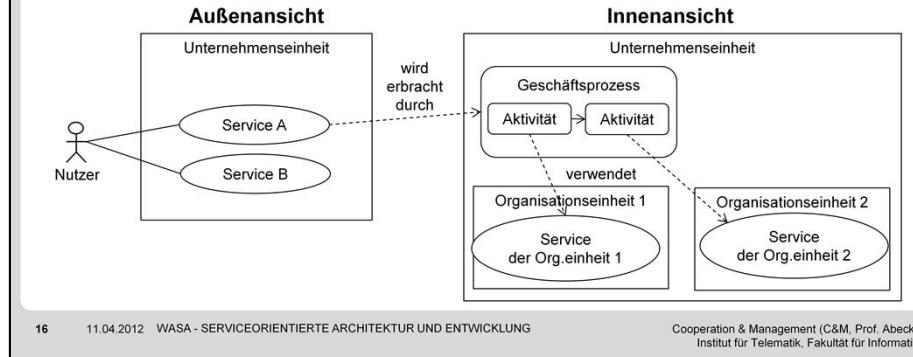
(Unterstützende Services) Ein Beispiel einer weiteren Gruppe von unterstützenden Services sind die Beratungsservices (z.B. zur Abwicklung von EU-Projekten -> solche Services liegen an der Grenze zu den Kerngeschäftsservices).

(Präsidiumsrundschreiben) Beispiel eines Services, der nicht vollständig elektronisch erbracht wird (weil die Rundschreiben teilweise auf Papier per Hauspost verteilt werden).

(Geschäftliches Ziel) Neben der Einordnung des betrachteten Serviceausschnitts werden in diesem Schritt der Methode auf dieser Detailebene die geschäftlichen Ziele angegeben, die mit dem Serviceausschnitt verknüpft werden. Im Falle der Campusbegleitung kann als ein Ziel formuliert werden: Es muss sichergestellt werden, dass ein den Service in Anspruch nehmender Besucher des Campus sich auf dem Gelände gut orientieren kann und mit der (virtuellen) Begleitung zufrieden ist.

Zusammenhang zwischen Geschäftsservice und Geschäftsprozess

- (1) Ein Geschäftsservice ist geschäftliche Leistung, die ein Servicegeber gegenüber Servicenehmern erbringt
- (2) Ein Geschäftsprozess ist eine Folge von Aktivitäten zur Erreichung eines geplanten Arbeitsergebnisses in einem Unternehmen
- (3) Geschäftsservices werden durch Geschäftsprozesse realisiert



[EH+08:96]

(1) Ein Geschäftsservice (engl. business service) ist ein Element geschäftlichen Verhaltens [EH+08:96].

- Servicegeber: Einheit des Unternehmens, also eine Abteilung oder eine einzelne Stelle.
- Servicenehmer: Kunde oder externer Partner oder Einheit des Unternehmens

Es liegt ein Vertrag (engl. contract) zugrunde, der die vom Servicenehmer bei Nutzung des Services durchzuführenden Aktionen beschreibt

(2) Ein Geschäftsprozess (engl. business process) dient direkt oder indirekt einer Leistung für einen Kunden oder den Markt [:96].

Die Folge von Geschäftsprozessaktivitäten, kurz Aktivitäten, ist funktions- und stellenübergreifend.

(3) Geschäftsservices beschreiben die Außenansicht an der Schnittstelle zum Servicenehmer.

Geschäftsprozesse beschreiben die Innenansicht des Servicebringers.

[EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.

- (1) Der KITCampusGuide (KCG) dient zur Bereitstellung des unterstützenden Geschäftsservices "Campusbegleitung"
 - (1) Nutzer des Geschäftsservices sind alle potentiellen Besucher des KIT-Campus
- (2) Der Campus-Begleitservice gehört zu den Informationsservices
 - (1) Das Ziel ist, den Personen auf dem Campus eine bessere Orientierung zu ermöglichen
- (3) Der den Service realisierende Geschäftsprozess nutzt die Services der folgenden Abteilungen
 - (1) Personalverwaltung: Personeninformationen abfragen
 - (2) Gebäudeverwaltung: Standort abfragen
 - (3) Externe Servicegeber: Karte abfragen, Route berechnen

Es wird hier eine ausschließlich geschäfts- und organisationsorientierte Sicht auf den KITCampusGuide (KCG) eingenommen.

(1) Der KCG unterstützt nicht unmittelbar das Kerngeschäft des KIT (Lehre, Forschung, Innovation) und trägt daher zur Erbringung eines unterstützenden Geschäftsservices, dem Campus-Begleitservice, bei.

Der Name suggeriert, dass der Service einen Begleitassistenten für den Nutzer darstellen soll, der ihm den Weg zeigen soll und Informationen zu Gebäuden und anderen Standorten auf dem Campus geben soll.

(1.1) Der Nutzerkreis wird über die Festlegung der Autorisierung des Begleitervices zu bestimmen sein.

(2) Insgesamt soll der Aufenthalt auf dem Campus für die Besucher attraktiver gestaltet werden, womit das KIT mit dem Geschäftsservice gewisse Marketing-Ziele (Geschäftsziel Steigerung der Besucherzufriedenheit) verfolgt.

(3) Eine detaillierte Beschreibung des Geschäftsprozesses in BPMN wird in [C&M-DI-Ge11] gegeben.

(3.1) (3.2) (3.3) Das Beispiel ist bewusst klein gehalten, was zwangsläufig dazu führt, dass die vom Campus-Begleitservice in Anspruch genommenen Services eher technisch und feingranular ausfallen.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

(1) Geschäftsservices ...

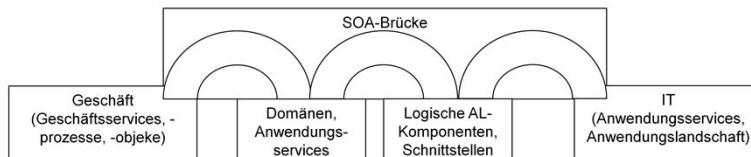
- (1) ... sind grundsätzlich durch IT zu unterstützen
- (2) ... werden vom IT-Unternehmensarchitekten ermittelt
- (3) ... werden durch einen Geschäftsprozess realisiert
- (4) ... können zur Durchführung von Geschäftsprozessaktivitäten genutzt werden
- (5) ... sollten als Eigenschaften aufweisen
 - (1) feingranular
 - (2) geschäftsrelevant
 - (3) hohe Abhängigkeiten zu anderen Geschäftsservices
 - (4) lose Kopplung
 - (5) komponierbar, um Geschäftsprozesse flexibel unterstützen zu können

LZ SERVICEORIENTIERUNG ÜA BEISPIEL KIT UND KCG



- (1) Was ist ein Beispiel eines Kerngeschäftsservice des KIT und welche Rollen sind daran beteiligt?
- (2) Wie lässt sich die durch den KITCampusGuide erbrachte Funktionalität in den Geschäftsservice-Katalog einordnen?
- (3) Der Zusammenhang zwischen Geschäftsservice und Geschäftsprozess ist am Beispiel des Services "Campusbegleitung" zu erklären

- (1) Geschäftsservices eines Unternehmens bestimmen die von deren Anwendungslandschaft (AL) bereitzustellenden Funktionen
- (2) An der Erbringung eines Geschäftsservices sind häufig mehrere Anwendungen beteiligt
 - (1) Punkt-zu-Punkt-Integration von Anwendungen erfüllt nicht die Gestaltungsziele Effektivität, Effizienz und Agilität
- (3) Mit einer serviceorientierten Architektur (SOA) wird eine Abstraktionsebene zwischen Geschäft und IT eingeführt
 - (1) Einfachere Erreichung aller Gestaltungsziele von ALs



20

11.04.2012 WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)

Institut für Telematik, Fakultät für Informatik

[EH+08:98]

Bislang lag der Schwerpunkt in dieser Kurseinheit auf der Betrachtung der Serviceorientierung aus Geschäftssicht. Jetzt wird der Übergang vom Geschäft zur IT behandelt und nachfolgend stehen dann die (software-) technischen Aspekte der Serviceorientierung im Mittelpunkt.

(1) Der Grund hierfür ist, dass nahezu alle Geschäftsprozesse durch IT unterstützt werden.

(2) In diesem Fall muss ein Unternehmen die Anwendungen integrieren.

(2.1) Durch individuell und ad-hoc gestaltete Punkt-zu-Punkt-Verbindungen werden Anwendungslandschaften eng gekoppelt und die fachlichen Funktionen werden zu gering gekapselt.

(3) Die fachlich orientierte Sichtweise der SOA ergänzt die eher technisch geprägten Technologien der Enterprise Application Integration (EAI), wodurch die bereits durch EAI im Vergleich zur Punkt-zu-Punkt-Integration verbesserte Effizienz und Agilität noch weiter gesteigert wird [:100].

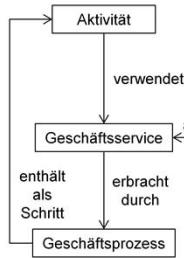
Die Abstraktionsschicht enthält neue, technologieunabhängige Artefakte, wie z.B. Domänen und Anwendungsservices [:101] und schließen die Lücke zwischen Geschäft und Systemen der Anwendungslandschaft.

AL	Anwendungslandschaft
EAI	Enterprise Application Integration

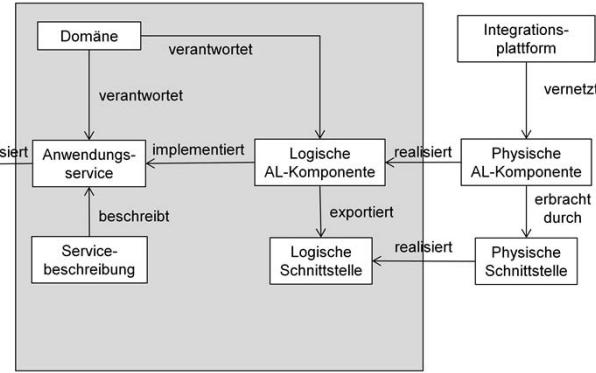
[EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.

SOA-Artefakte und ihre Abhängigkeiten

Geschäft



IT



- (1) Die Artefakte liefern ein abstraktes Begriffsgebäude, das unabhängig ist von konkreten Standards, Technologien oder spezifischen Implementierungsdetails

21

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[EH+08:102]

(1) Der für die Artefakte gewählte Abstraktionsgrad ist vergleichbar mit dem der OASIS-Spezifikation "Reference Model for Service Oriented Architecture 1.0".

(Anwendungsservice) Werden von logischen AL-Komponenten über ihre logischen Schnittstellen bereitgestellt.

(Domäne) Der spezifische Verantwortungsbereich (z.B. ein Geschäftsbereich oder eine geschäftliche Aufgabe), der einen Anwendungsservice verwaltet bzw. verantwortet.

(Servicebeschreibung) Öffentlich zugängliche Informationen (Name, Adresse, Ein-/Ausgabeparameter der Operationen), die ein Servicenutzer zur Verwendung des Services benötigt. Diese rein syntaktische Information kann durch semantische Angaben ergänzt werden [:103].

(Physische AL-Komponenten) Implementierte, im Rechenzentrum betriebene Softwaresysteme, wie z.B. ein SAP- oder ein Siebel-System.

(Integrationsplattform) Vernetzt physische AL-Komponenten und bietet ihnen eine Ablaufumgebung; eine Ausprägung ist ein Enterprise Service Bus (ESB).

ESB Enterprise Service Bus

OASIS Organization for the Advancement of Structured Integrated Standards

[EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.

Richtig oder falsch

- (1) Eine Servicebeschreibung bezieht sich auf einen Geschäftsservice
- (2) Eine Domäne ist hier als ein Verantwortungsbereich aufzufassen
- (3) Anwendungsservices werden durch logische Anwendungslandschafts-Komponenten implementiert
- (4) Eine Integrationsplattform besteht aus logischen Anwendungslandschaftskomponenten

Begriffsdefinitionen: Architektur und serviceorientierte Architektur

- (1) Architektur (eines Systems)
 - (1) Eine Menge von Regeln zur Definition der Struktur eines Systems und der Zusammenhänge zwischen seinen Teilen (ISO)
 - (2) Die fundamentale Organisation eines Systems in Form von dessen Komponenten, deren Beziehungen zueinander und der Prinzipien, die den Design und die Evolution des Systems leiten (IEEE)
- (2) Serviceorientierte Architektur (Service-oriented Architecture SOA)
 - (1) Ein IT-Architekturansatz, der die Erzeugung von Geschäftsprozessen von funktionalen Einheiten, die als Services definiert sind, unterstützt (Zhang)
 - (2) Ein architektureller Stil zum Aufbau und zur Zusammenstellung von Anwendungssystemen (Colan)

(1) Es folgen zwei Definitionen der ISO und IEEE, wobei die IEEE-Definition feiner ist, da sie Designprinzipien mit einbezieht.

(1.1) Aus dem Standard "ISO/IEC 10746-2": An architecture (of a system) is a set of rules to define the structure of a system and the interrelationships between its parts.

(1.2) Aus dem Standard "IEEE 1471": An architecture is the fundamental organization of a system embodied in its components, their relationship to each other and the environment, and the principles guiding its design and evolution.

(2.1) A Service-oriented Architecture (SOA) is an “information technology (IT) architectural approach that supports the creation of business processes from functional units defined as services” [Zh05].

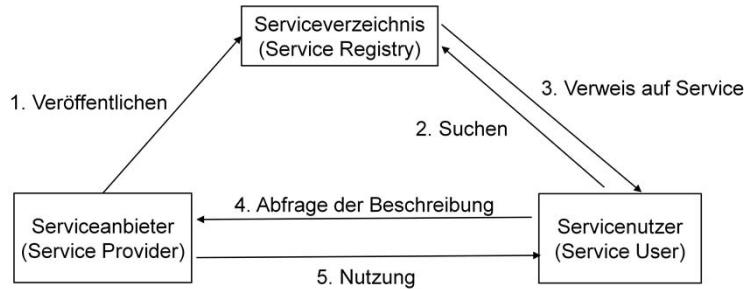
(2.2) SOA is the architectural style of building applications based on services, and deals with the arrangement of services in application systems ([Co05]).

[Co05] Mark Colan: SOA and Web Services, presentation slides, 2005, <http://ibm.com/developerworks/speakers/colan>, IBM Software Group.

[Zh05] Zhuopeng Zhang, Ruimin Liu, Hongji Yang: Service Identification and Packaging in Service Oriented Reengineering, Software Engineering and Knowledge Engineering (SEKE), 2005.

Servicemodell als der Kern einer SOA

- (1) SOA ist ein abstraktes Konzept einer Softwarearchitektur
- (2) Ein neuer Aspekt einer SOA ist dadurch gegeben, dass benötigte Funktionalität in Form von Services zur Laufzeit eingebunden werden kann
- (3) Die Basis für die dynamische Bindung ist das Servicemodell



24

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[DJ+05]

(1) SOA ist im Gegensatz zum Remote Procedure Call (RPC) oder zur Remote Method Invocation (RMI) keine Technik, sondern eine Abstraktion, bei der bestimmte technische Aspekte der Wirklichkeit unterdrückt werden [:8].

(2) Beim RPC bzw. RMI waren die Aufrufe, durch die die Kopplung der Anwendungen realisiert wurden, hart in die Anwendung einkodiert [:8].

(3) Das Servicemodell (das "magische Dreieck einer SOA" [:12]) führt drei verschiedene Rollen ein, die jeweils eng mit dem Service verknüpft sind.

(Serviceanbieter) Stellt Services bereit, was neben der Entwicklung der Servicefunktionalität auch den Betrieb und die Wartung der Services umfasst. Die Güte des Betriebs und der Wartung wird durch den "Quality of Service" (QoS) ausgedrückt (z.B. Verfügbarkeit und Antwortzeit).

(1. Veröffentlichen) Durch das Veröffentlichen eines Services in einem Serviceverzeichnis stellt der Serviceanbieter sicher, dass der Service von einem potentiellen Servicenutzer gefunden werden kann.

(Serviceverzeichnis) Ist vergleichbar mit den "gelben Seiten" und unterstützt das Finden der von einem potentiellen Servicenutzer benötigten Services. Es gibt nicht das eine Verzeichnis. Vielmehr kann ein Unternehmen z.B. getrennte Verzeichnisse für intern und extern genutzte Services vorsehen oder es können Verzeichnisse für einzelne Branchen existieren. Ein Verknüpfung der Serviceverzeichnisse kann man sich ähnlich wie beim Domain Name Service (DNS) vorstellen, bei dem Anfragen transparent an andere DNS-Verzeichnisse weitergeleitet werden [:15].

(Servicenutzer) Mit Ausnahme der Veröffentlichen-Aktion hängen alle im Servicemodell auftretenden Aktionen mit dem Servicenutzer zusammen. Damit der Servicenutzer einen Service finden kann, damit der Servicebeschreibung etwas anfangen kann und den Serviceaufrufen und nutzen kann, müssen entsprechende Standards existieren [:16].

DNS	Domain Name Service
QoS	Quality of Service
RMI	Remote Method Invocation
RPC	Remote Procedure Call

[DJ+05] Wolfgang Dostal, Mario Jaeckle, Ingo Melzer, Barbara Zengler: Service-orientierte Architekturen mit Web Services, Elsevier Spektrum Akademischer Verlag, 2005.

Kategorien von Anwendungsservices

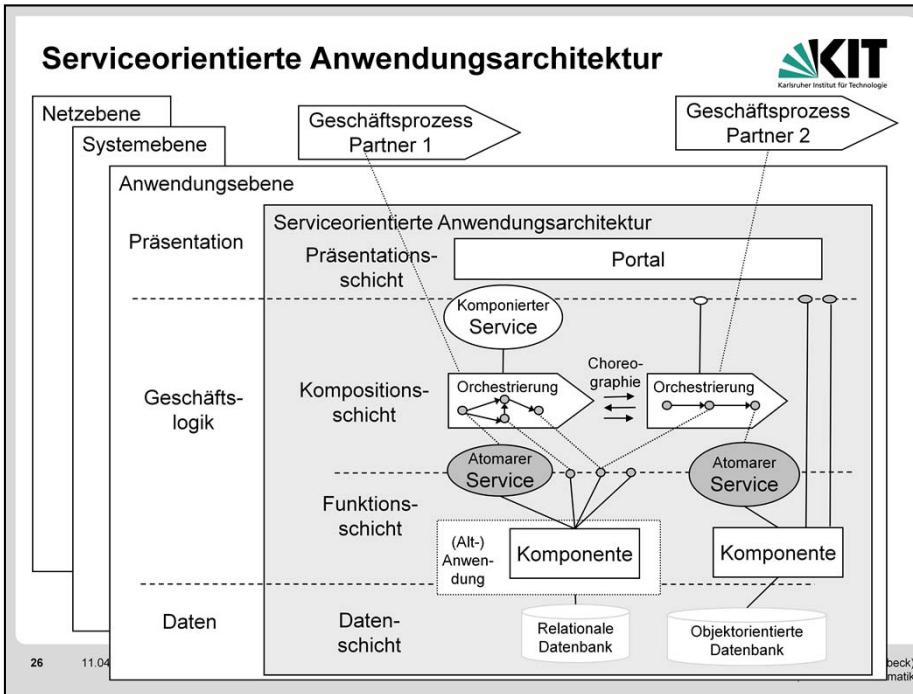
- (1) Utility-Service
 - (1) Generische Querschnittsaufgaben, keine Geschäftslogik
 - (2) Hoher Wiederverwendungsgrad
- (2) Entity-Service
 - (1) Geschäftszentrischer agnostischer Servicetyp
 - (2) Hoher Wiederverwendungsgrad
- (3) Task-Service
 - (1) Geschäftszentrischer nicht-agnostischer Service
 - (2) Benötigt für einen spezifischen Geschäftsprozess

In der Literatur bestehen verschiedene (im Kern z.T. ähnliche und vergleichbare) Ansätze, die im Servicebestand auftretenden Anwendungsservices gemäß charakteristischer Eigenschaften zu klassifizieren. Die hier unterschiedenen drei Servicekategorien sind aus [Er08].

- (1) Utility-Service: Generische Verarbeitungslogik, die Querschnittsaufgaben (z.B. Logging, Benachrichtigung) erfüllt und keine Geschäftslogik darstellt [:164]. Diese Services stehen in engem Zusammenhang zu der Integrationsplattform (ESB) als Teil der gesamten Unternehmensarchitektur.
- (2) Entity-Service: Geschäftszentrischer agnostischer Servicetyp, der elementare und hochgradig wiederverwendbare Funktionen auf Geschäftsobjekten/-einheiten (engl. business entity) bereitstellt. Diese Funktionen werden häufig auch als CRUDS-Operationen (Create, Read, Update, Delete, Search) bezeichnet.
- (3) Task-Service: Geschäftszentrischer nicht-agnostischer Service, zu dessen Erbringung häufig andere (insbesondere Entity) Services erforderlich sind und der für einen spezifischen Geschäftsprozess benötigt wird.

CRUDS Create, Read, Update, Delete, Search

[Er08] Thomas Erl: SOA Design Patterns, Prentice Hall, 2008.



(Netzebene, Systemebene, Anwendungsebene) Die SOA ist als eine Softwarearchitektur auf der Anwendungsebene angesiedelt. Die in der SOA auftretenden Services kapseln gewisse Anwendungsfunktionalität, die in Form von (Anwendungs-)Komponenten implementiert ist.

(Daten, Geschäftslogik und Präsentation) Bei der komponentenbasierten Programmierung einer verteilten Anwendung werden drei logische Schichten – Daten, Geschäftslogik und Präsentation – unterschieden.

(Kompositionsschicht, Funktionsschicht) Diese drei Aspekte finden sich auch in der evolutionären Weiterentwicklung der serviceorientierten Anwendungsarchitektur wieder, wobei die mittlere Geschäftslogiksschicht in eine die Komponenten beinhaltenden Funktionsschicht und eine Kompositionsschicht unterteilt wird, in der die Servicekomposition erfolgt.

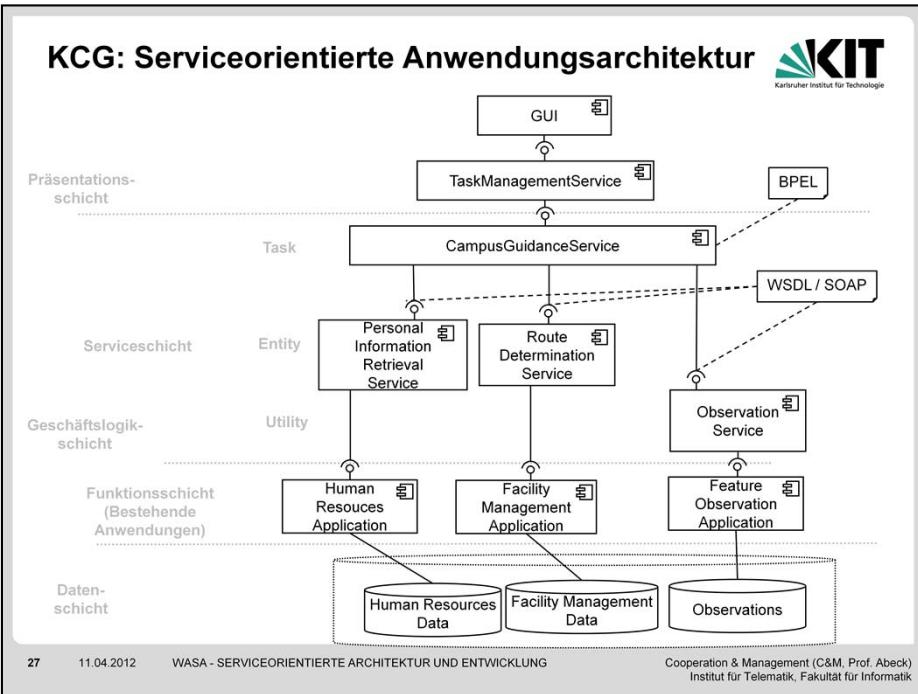
(Orchestrierung, Chorographie) Die Zusammenstellung von Services zur Unterstützung des Geschäftsprozesses eines Prozesspartners wird als Orchestrierung bezeichnet, während die Chorographie die Interaktion zwischen zwei (orchestrierten) Prozessen beschreibt.

(Atomarer Service, Komponierter Service) Services, die von einer Komponente erbracht werden, heißen atomar. Solche atomaren Services können entweder direkt von der Präsentationsschicht genutzt werden oder zur Erbringung komplexerer Services, den komponierten Services, verwendet werden.

((Alt-)Anwendung, Komponente) Bei einer Komponente kann es sich um ein beliebiges Stück ablauffähigen Code handeln. Im Idealfall handelt es sich um eine mit moderner Komponententechnologie (z.B. JEE) und objektorientierten Techniken entwickelten Softwarekomponenten. In der Praxis treten häufig bestehenden Anwendungen (Alt-Anwendungen, Legacy Applications) an dieser Stelle auf, die um entsprechende Serviceschnittstellen erweitert wurden, um auf deren Funktionalität zugreifen zu können. Auf diese Weise lassen sich solche Anwendungen elegant in eine moderne und erweiterbare Gesamt-IT-Architektur integrieren.

JEE

Java Enterprise Edition



Die Schichtenarchitektur einer serviceorientierten Anwendung wird am Beispiel eines Ausschnitts aus der serviceorientierten KCG-Architektur aufgezeigt. Da es sich hierbei um eine spezielle Ausprägung einer Softwarearchitektur handelt, bietet sich das UML-Komponentendiagramm als graphische Notation an.

(Datenschicht, Geschäftslogikschicht, Präsentationsschicht) Die klassischen drei logischen Schichten einer Anwendungssoftwarearchitektur, die auch in einer SOA auftreten.

(Funktionsschicht, Serviceschicht) Die Geschäftslogikschicht zerfällt in einer SOA in diese zwei Teilschichten.

(Task Entity, Utility) Typen von Services, die sich im Wesentlichen aufgrund der Kriterien "Wiederverwendbarkeitsgrad" und "Geschäftsbezug" unterscheiden.

Aus der Perspektive der Softwarearchitektur handelt es sich bei den Services um eine spezielle Art von Komponenten.

(TaskManagementService) Der Service realisiert die Auftragsverwaltung und kommuniziert mit der GUI (bzw. mit einer Mensch-Maschine-Schnittstelle, Human Machine Interface, HMI) und dem Task-Service "CampusGuidanceService".

(CampusGuidanceService, BPEL) Ein Task-Service, der den Prozess zur Begleitung des Besuchers auf dem Campus unterstützt. Der Service wird durch Nutzung verschiedener Entity- und Utility-Services erbracht, die mittels der Business Process Execution Language (BPEL) komponiert werden.

(PersonalInformationRetrievalService) Dient dazu, Informationen über KIT-Mitglieder (Studierende, Mitarbeiter) zu ermitteln.

(RouteDeterminationService) Es wird im Beispiel davon ausgegangen, dass das Facility Management über die Fähigkeit verfügt, Routen zu ermitteln.

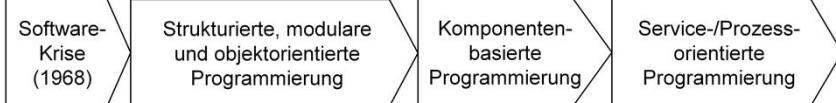
(Observation Service) Dieser Utility-Service ermöglicht die Beobachtung gewisser Gegenstände (engl. feature) auf dem Campus. Ein Beispiel ist die Beobachtung, ob in einem Raum Personen anwesend sind (z.B. über eine Kamera oder andere Sensoren).

Hinweis: Inwieweit dieser Service tatsächlich Querschnittsfunktionalität erbringt und daher als Utility-Service und nicht als Entity-Service eingeordnet ist, hängt von der jeweiligen Situation ab und stellt eine Designentscheidung dar.

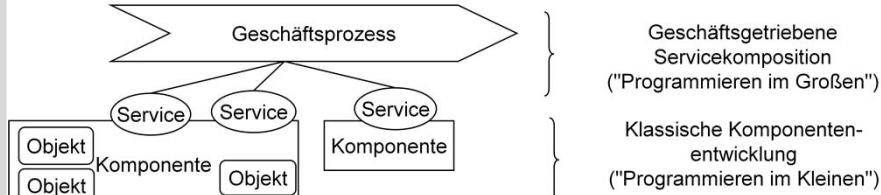
BPEL	Business Process Execution Language
HMI	Human-Machine Interface

- (1) Welche drei Rollen werden im SOA-Servicemodell ("SOA-Dreieck") unterschieden?
- (2) Welche Aufgaben übernimmt das Service-Verzeichnis?
- (3) Welche drei Kategorien von Anwendungsservices lassen sich unterscheiden
- (4) Sind Task-Services im Vergleich zu Entity-Services weniger agnostisch und schlechter wiederverwendbar?
Die Antwort ist am Beispiel der serviceorientierten Architektur zum KITCampusGuide zu verdeutlichen

Evolution des Programmier-Stils



- (1) Aus der Sicht der Software-Entwicklung stellt das SOA-Paradigma ein evolutionäres Programmierkonzept dar
- (2) Programmieren im Großen zur flexiblen Unterstützung von Geschäftsprozessen



Parallel zur Evolution der Softwarearchitektur wurden die Programmiertechniken ausgeprägt, die den Aufbau solcher Architekturen unterstützt haben.

(Softwarekrise, Strukturierte, modulare und objektorientierte Programmierung) Ausgehend von der Erkenntnis, dass monolithische, nicht strukturierte Programme ab einer gewissen Größe nicht mehr überschaubar und wartbar sind, wurden Programmiersprachen entwickelt, die eine Kapselung von Funktionalität in Form von Unterprogrammen, Prozeduren und Modulen ermöglichen. Mit der Entwicklung der Objektorientierung Mitte der 80er Jahre wurde ein Strukturierungskonzept für Programme entwickelt, das den heutigen Stand der Technik in der Programmierung bildet.

(Komponentenbasierte Programmierung) Im Vergleich zu einem Objekt ist eine Komponente eine in einer meist verteilten Laufzeitumgebung auslieferbare (engl. deployable) Einheit. Außerdem ist eine Komponente grobgranularer und nicht wie Objekte auf eine bestimmte Programmiersprache eingeschränkt. Objektorientierung und Komponentenbasierung ergänzen sich insofern, als dass Komponenten meist mehrere Objekte bündeln.

Die komponentenbasierte Programmierung ist der erste Entwicklungsschritt zum Aufbau verteilter Softwaresysteme.

(Service-/Prozess-orientierte Programmierung) Setzt auf der komponentenbasierten Programmierung auf und setzt diese in evolutionärer Form fort. Die mit dieser Art der Ziele (lose Kopplung, dynamische Binden, Geschäftsprozessunterstützung) verfolgten Ziele wurden bereits erläutert.

Mit den Web-Services steht ein allgemein akzeptierter Standard zur Verfügung, mit dem die SOA-Konzepte plattformunabhängig umgesetzt werden können. Die Web-Services haben im SOA-Umfeld einen Akzeptanzgrad erreicht, der dem CORBA-Standard im Bereich der komponentenbasierten Programmierung verweht blieb.

(2) Die Idee zu einer Unterscheidung zwischen einem Programmieren im Großen und einem Programmieren im Kleinen wurde schon 1976 von Deremer und Kron vorgeschlagen [DJ+05:17].

[DJ+05] Wolfgang Dostal, Mario Jaeckle, Ingo Melzer, Barbara Zengler: Service-orientierte Architekturen mit Web Services, Elsevier Spektrum Akademischer Verlag, 2005.

Relevante Vorarbeiten zur serviceorientierten Softwareentwicklung



- (1) Strukturierte Softwareentwicklungsprozesse des Software Engineering
 - (1) Ein Beispiel ist das in Bruegge & Dutoit beschriebene Vorgehen
- (2) Service Oriented Architecture Framework (SOAF) von Erradi et al.
- (3) Serviceorientierte Analyse und Design von Erl
- (4) Quasar Enterprise von Engels et al. bzw. sd&m
- (5) Rational Unified Process / Service-oriented Modeling and Architecture (RUP/SOMA) von IBM

30

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

Seit dem Jahr 2005 werden Konzepte zur strukturierten Entwicklung von serviceorientierten Softwaresystemen in der Wissenschaft und in der Industrie vorangetrieben. Die wichtigsten Arbeiten werden hier im Überblick vorgestellt.

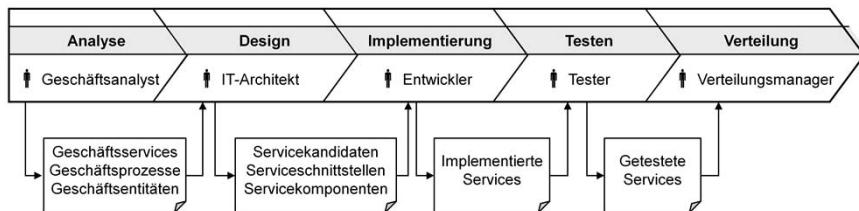
- (1) Die Tatsache, dass das klassische Software Engineering die Grundlage der serviceorientierten Softwareentwicklung ist, unterstreicht den evolutionären Charakter dieses neuen Ansatzes.
- (2) SOAF ist ein Beispiel für eine Arbeit aus dem wissenschaftlichen Umfeld [EA+06].
- (3) Thomas Erl stellt eine umfassende Sammlung an Büchern zu SOA bereit, die an der Schnittstelle zwischen Wissenschaft und Praxis einzuordnen sind.
- (4) Prozessorientierte Integration einer bestehenden Anwendungslandschaft mit einer Fokussierung von IT-organisatorischen Themen. Die zuvor beschriebenen Konzepte und Methoden orientieren sich weitestgehend an dieser Quelle.
- (5) Die Modellierung auf der Basis von UML bilden einen Schwerpunkt. IBM ist mit dem SOMA-Ansatz führend in Bezug auf Werkzeugunterstützung für das Engineering von serviceorientierten Lösungen.

[BD04] Bernd Bruegge, Allen H. Dutoit: Object-Oriented Software Engineering Using UML, Patterns and Java, Pearson Prentice Hall, 2004.

[EA+06] Abdelkarim Erradi, Sriram Anand, Naveen Kulkarni: SOAF – An Architectural Framework for Service Definition and Realization, 2006.

[EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.

Phasen des serviceorientierten Softwareentwicklungsprozesses



- (1) In der Geschäftsanalyse werden die von gewissen internen oder externen Serviceleistern bereitzustellenden Geschäftsservices aus den Anforderungen abgeleitet
- (2) In den nachfolgenden Phasen werden die Geschäftsservices überführt in Software-Artefakte als Teil der serviceorientierten Softwarearchitektur

[C&M-DI-Ge11:53]

Das im folgenden vorgestellte Vorgehen zur Entwicklung von Services orientiert sich an den in der Softwaretechnik bereits etablierten Prozessen [BD04] und der in der Literatur verbreiteten Systematik für serviceorientierte Architekturen (z.B. [EH+08]).

(1) Während beim traditionellen Softwareentwicklungsprozess das zu entwickelnde System den Ausgangspunkt der Anforderungsanalyse bildet, steht beim serviceorientierten Softwareentwicklungsprozess das Geschäft und die zu dessen Erbringung benötigte Funktionalität in Form von Geschäftsservices im Vordergrund. Im Zusammenhang mit einem Geschäftsservice ist zu klären, von wem dieser erbracht werden soll.

(Geschäftsentitäten) Diese werden im Rahmen der Analyse durch die Erstellung eines Domänenmodells identifiziert, in dem auch die Abhängigkeiten zwischen den Entitäten festgehalten werden [:54].

(2) Bei der Überführung der Geschäftsservices in das Design von Anwendungsservices kommt die SOA als die für serviceorientierte Anwendungssysteme relevante Softwarearchitektur ins Spiel.

(Design) Im Rahmen des Designs werden alle Aspekte des Services spezifiziert, die zur Implementierung dieses Services benötigt werden. Diese sind:

- (i) Signatur der Serviceschnittstelle und darin auftretende Datentypen
- (ii) Servicekomponente, durch die der Service erbracht wird

(Implementierung) Überführung der zunächst unabhängig von konkreten Technologien erstellten Serviceentwürfe in technologiespezifische ausführbare Softwareartefakte.

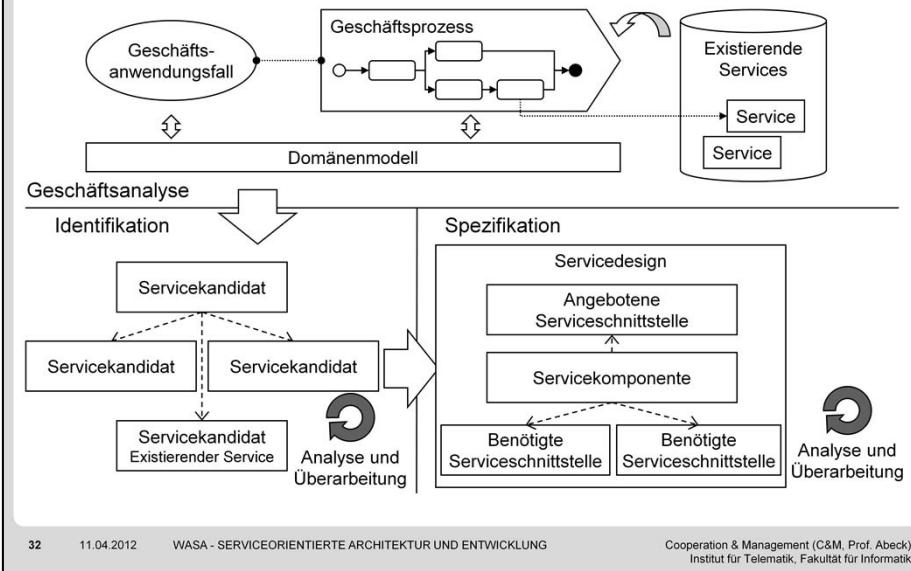
(Designprozess) Nachfolgend wird der die ersten zwei Phasen behandelnde Designprozess eines serviceorientierten Softwaresystems näher beschrieben.

[BD04] Bernd Bruegge, Allen H. Dutoit: Object-Oriented Software Engineering Using UML, Patterns and Java, Pearson Prentice Hall, 2004.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

[EH+08] Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten, dpunkt.verlag, 2008.

Designprozess



[C&M-DI-Ge11:55]

Nach der Designphase muss das zu entwickelnde System in einer kohärenten und wohldefinierten Repräsentation vorliegen. Diese Repräsentation sind im serviceorientierten Fall detaillierte, aber noch technologieunabhängigen Servicespezifikationen.

(Geschäftsanalyse) Den Ausgangspunkt hierzu liefern die in der vorgelagerten Geschäftsanalysephase ermittelten Anforderungen, die in Form von Geschäftsanwendungsfällen und Geschäftsprozessen vorliegen.

(Domänenmodell) Dient dazu, einheitliche Begrifflichkeiten bei der Erstellung der Analyseartefakte zu verwenden.

(Identifikation) Aus den Artefakten der Geschäftsanalyse werden Service- und Operationskandidaten systematisch abgeleitet.

Der Begriff des Kandidaten drückt aus, dass das Artefakt ein möglicher Bestandteil des zu entwickelnden Systems, aber erst dann implementiert wird, wenn er im Rahmen der nachfolgenden Spezifikation tatsächlich als notwendig erachtet wird.

(Spezifikation) Die Servicekandidaten werden ergänzt um eine vollständige Spezifikation der Serviceschnittstelle, die aus der angebotenen (engl. provided) Schnittstelle und den benötigten (engl. required) Schnittstellen besteht.

(Analyse- und Überarbeitung) Hierbei handelt es sich um eine iterative Überarbeitung der Entwürfe, die zur Sicherstellung von gewünschten Qualitätseigenschaften, wie z.B. loser Kopplung oder Auffindbarkeit dient.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

In den Entwicklungsschritten einsetzbare standardisierte Technologien

- (1) Analyse des Geschäfts durch Geschäftsservices (UML) und Geschäftsprozesse (BPMN)
- (2) Ableitung der Servicekandidaten aus dem Geschäftsprozess mit Systembezug (SoaML, Sprachelement "Capabilities")
- (3) Bewertung der Servicekandidaten im Hinblick auf Serviceeigenschaften und Festlegung der (umzusetzenden) Services im initialen Servicedesign (SoaML, Sprachelement "Service Interfaces")
- (4) Änderung des Servicedesigns aufgrund bestehender Services, die wiederverwendet werden können (SoaML)
- (5) Erstellung der das Servicedesign umsetzenden serviceorientierten Komponentenarchitektur (SCA)
- (6) Implementierung der SCA mittels geeigneter Technologien (Web-Services, Java)

(1) Ein Geschäftsprozess ohne Systembezug umfasst nur die (menschlichen) Geschäftspartner (Organisationseinheiten, Rollen), die jeweils durch einen Organisations-Pool repräsentiert werden. Ein Systembezug wird in einem solchen Geschäftsprozess dadurch hergestellt, indem dieser durch einen oder mehrere System-Pools ergänzt wird, um die Interaktion von organisatorischen Rollen mit Systemfunktionen zu modellieren.

(2) Es lassen sich zwei Arten von Servicekandidaten unterscheiden:

- (i) Die vom Geschäftsprozess abgeleiteten Kandidaten (engl. business process capabilities). Diese beschreiben die vom Geschäft geforderten (engl. required) Fähigkeiten.
- (ii) Die von der Systemunterstützung abgeleiteten Kandidaten (engl. application domain capabilities). Diese beschreiben die in den beteiligten Anwendungsdomänen bereitgestellten (engl. provided) Fähigkeiten.

(3) Serviceeigenschaften sind u.a. die lose Kopplung und die Wiederverwendbarkeit.

Durch die Bewertung lassen sich eventuelle Defizite aufdecken und durch Änderungen z.B. in der Strukturierung oder Namensgebung beseitigen.

Es ist zu überlegen, ob die Änderungen bereits auf der Ebene der Servicekandidaten oder erst auf der Ebene der Services im Servicedesign durchgeführt werden sollen.

(4) Je größer die Menge der bestehenden und wiederverwendbaren Services ist, umso stärker werden diese bereits den initialen Servicedesign (vorhergehender Schritt) beeinflussen.

(5) Die Umsetzung eines SoaML-Modells in eine initiale SCA sollte möglichst durch ein Werkzeug unterstützt werden.

(6) Die Wahl der einzusetzenden Technologien wird u.a. dadurch beeinflusst, in welchen Technologien die wiederverwendeten Services (Schritt 5) implementiert wurden.

Richtig oder falsch

- (1) Durch die serviceorientierte Programmierung werden die objektorientierte und die komponentenbasierte Programmierung obsolet
- (2) Der Übergang vom Geschäft zur IT (Softwarearchitektur) erfolgt im serviceorientierten Softwareentwicklungsprozess in der Phase des Servicedesigns
- (3) Mit der BPMN lassen sich Geschäftsservices modellieren
- (4) Aus Geschäftsprozessen lassen sich Servicekandidaten ableiten
- (5) Servicekandidaten lassen sich mittels der SoaML modellieren
- (6) Ein Servicekandidat enthält alle Informationen, um diesen Service zu implementieren

Domänenmodell

- (1) Das Domänenmodell dient zur einheitlichen Verwendung von Begriffen in den nachfolgend erstellten Artefakten
 - (1) Dient zur Beschreibung der Konzepte der Domäne und der Beziehungen zwischen den Konzepten
 - (2) Die Spezifikation einer solchen Konzeptualisierung wird als Ontologie bezeichnet
- (2) Zu dieser Formalisierung kann die die Ontology Web Language (OWL) genutzt werden
 - (1) Ein Grund ist die Unterstützung der semantische Annotation von Web-Services
 - (2) Alternativ könnte auch UML bzw. das Ontology Definition Metamodell (ODM) genutzt werden
- (3) Die durch das Domänenmodell eingeführten Konzepte bestimmen die Bezeichnung der Aktivitäten und Nachrichtenelemente

[C&M-Di-Ge11:57]

Das Domänenmodell ist gemäß [C&M-Di-Ge11] das erste im Rahmen der Geschäftsanalysephase (und damit das allererste im Rahmen des gesamten serviceorientierten Softwareentwicklungsprozesses) erstellte formale Artefakt.

(1) Das betrifft insbesondere die Artefakte der Geschäftsanalyse, also die Geschäftsanwendungsfälle und Geschäftsprozesse. Da die Artefakte der Designphase systematisch aus den Analyseartefakten abgeleitet werden, wirkt sich die durch das Domänenmodell erreichte Begriffsvereinheitlichung auch auf die weiteren Artefakte vorteilhaft aus.

(1.1) Die Konzepte beschreiben zwei Arten von Entitäten der Geschäftswelt:

- (i) materielle Entitäten (z.B. ein Campusbesucher, eine Campuskarte oder ein Raum) oder
- (ii) immaterielle Entitäten (z.B. eine Routenplanung oder die Beauftragung, eine Route zu planen)

(1.2) Der Begriff der Ontologie geht auf eine frühe Arbeit von Gruber zurück.

(2) OWL ist eine vom W3C standardisierte XML-basierte Sprache, die sich insbesondere im Bereich des Semantic Web etabliert hat.

(2.1) Diese Unterstützung leistet der Standard mit der Bezeichnung "Semantic Annotation for WSDL and XML Schema" (SAWSDL). Hierin wird gefordert, dass das Domänenmodell XML-basiert vorliegen muss.

(2.2) Grundsätzlich kann UML ohne jegliche Erweiterung zur Domänenmodellierung genutzt werden, was aber aufgrund einer fehlenden Überführung in OWL nicht praktikabel ist.

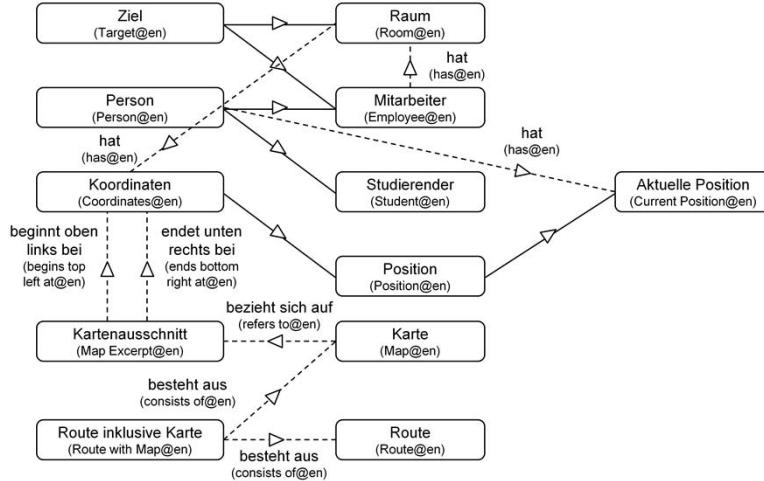
ODM ist ein UML-Profil, das eine automatisierte Überführung in OWL unterstützt.

(2.3) Hierdurch wird gerade die gewünschte einheitliche Benennung erzielt.

(3) Die Aktivitäten treten innerhalb der Geschäftsprozesse und Geschäftsanwendungsfälle auf, während die Nachrichten zwischen den Prozessteilnehmern ausgetauscht werden [:60].

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

KCG: Domänenmodell



(1) Zu der graphischen Darstellung besteht ein die Inhalte vollständig wiedergebendes XML-Dokument

[C&M-DI-Ge11:155]

Die Darstellung des Domänenmodells, das dem KITCampusGuide (KCG) zugrunde liegt, orientiert sich am Werkzeug OntoGraf, das Teil des weit verbreiteten Ontologie-Werkzeugs Protégé (entwickelt vom Stanford Center for Biomedical Informatics Research) ist. In dieser Darstellung repräsentiert

- (i) ein abgerundetes Rechteck eine Klasse, durch die ein einzelnes Konzept einer Domäne beschrieben wird
- (ii) eine gerichtete Kante eine Beziehung zwischen den Konzepten.

Die Vererbungsbeziehung ("is-a") wird durch eine durchgezogene Linie beschrieben, wobei der Pfeil auf die UnterkLASSE zeigt (im Gegensatz zum Vererbungspfeil in einer UML-Assoziation).

Beispiel: Raum "ia-a" Ziel, Mitarbeiter "is-a" Ziel, Mitarbeiter "is-a" Person

Die anderen Arten von Beziehungen werden durch Bezeichnungen ausgedrückt, die an die gestrichelten gerichtete Kanten geschrieben werden.

Beispiele:

- (i) besteht aus: entspricht der Komposition in UML (gefüllte Raute); Route inklusive Karte "besteht aus" Karte (und) Route
- (ii) hat: entspricht der (einfachen) Assoziation; Raum "hat" Koordinaten

(@en) Es können zusätzliche Bezeichnungen (engl. label) zu den Klassen oder Objekteigenschaften (engl. object properties) hinzugefügt werden.

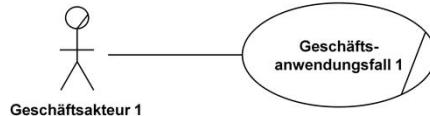
Durch das Kürzel "@en" ist vermerkt, dass es sich bei dem Label um den englischen Begriff handelt, wodurch eine Mehrsprachigkeit im Domänenmodell erreicht wird.

- (1) Das XML-Dokument ist die OWL-Beschreibung dieses graphisch beschriebenen Domänenmodells.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

Geschäftsanwendungsfälle

- (1) Inhalt eines Geschäftsanwendungsfalls
 - (1) Ausschnitt des Geschäfts, der durch IT unterstützt werden soll
 - (2) Beschreibt einen Geschäftsprozess von einem externen, einen Mehrwert erbringenden Standpunkt
- (2) Vorgehen bei der Analyse des Geschäfts
 - (1) Identifikation der Geschäftsakteure
 - (2) Identifikation der Geschäftsanwendungsfälle aufgrund
 - (1) des Mehrwerts, den ein Geschäftsakteur hierdurch erhält
 - (2) der Aufgabe, die ein Geschäftsakteur durchführt



37

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[C&M-Di-Ge11:60]

Die Geschäftsanwendungsfälle werden von RUP/SOMA im Rahmen der Geschäftsanalyse verwendet.

- (1.1) Diese IT-Unterstützung erfolgt auf der Grundlage einer serviceorientierten Architektur.
- (1.2) Mit einem Geschäftsanwendungsfall (genauer einer Instanz davon) ist eine Folge von Geschäftsaktionen verbunden, die für einen Geschäftsakteur ein beobachtbares, für ihn wertvolles Resultat liefert.

Hinsichtlich dieses Aspekts besteht eine Analogie zum Systemanwendungsfall, da durch Nutzung des Systems gemäß dem Anwendungsfall für den Nutzer ebenfalls ein Mehrwert entstehen soll.

- (2.1) Kandidaten für Geschäftsakteure sind Personen, Gruppen, Organisationen oder Maschinen, die mit dem Geschäft interagieren [:61].

Die Benennung der Geschäftsakteure sollte Aufschluss über seine Rolle im Geschäft geben und konsistent mit dem Domänenmodell sein (ggf. ist hierzu das Domänenmodell anzupassen oder zu erweitern).

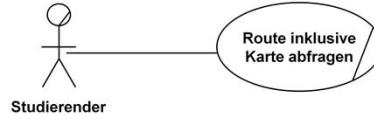
- (2.2) Die beiden Ansätze reflektieren genau die beiden Aspekte, die ein Geschäftsanwendungsfall aufweist:

(2.2.1) Die Außensicht: Was bringt dieser?

(2.2.2) Die Innensicht: Was wird geleistet?

Semantik: "Geschäftsakteur 1" ist ein externer Teilnehmer, der den "Geschäftsanwendungsfall 1" auslöst und damit einen Ablauf innerhalb des betrachteten Geschäfts anstößt.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.



- (1) Es wird der Studierende als der im Mittelpunkt stehende Geschäftsakteur des Geschäfts zum KITCampusGuide identifiziert
- (2) Mittels seines Mobiltelefons stößt er den Geschäftsanwendungsfall an
 - (1) Das Resultat liefert ihm einen Mehrwert
- (3) Die im Diagramm verwendeten Bezeichnungen sind aus dem Domänenmodell übernommen

[C&M-DI-Ge11:156]

- (1) Alternativ können noch weitere Personen wie Mitarbeiter oder auch Gäste des KIT als Geschäftsakteure in Frage kommen.
- (2) Mit dem Geschäftsanwendungsfall ist ein Geschäftsprozess verbunden, der nachfolgend aufgezeigt wird.
 - (2.1) Das ist neben dem eher funktionalen Aspekt des Anstoßens der zweite relevante Aspekt, den ein Geschäftsanwendungsfall zu erfüllen hat.
- (3) Das betrifft sowohl die Bezeichnung des Geschäftsakteurs als auch die des Geschäftsanwendungsfalls.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

Geschäftsprozesse

- (1) Das interne Verhalten eines Geschäftsanwendungsfalls wird durch einen oder mehrere Geschäftsprozesse beschrieben
- (2) Geschäftsprozesse umfassen zwei Arten von Abläufen
 - (1) Orchestrierung
 - (1) Interner Ablauf, der durch die von einem Geschäftsakteur durchgeführten Aktivitäten und deren Reihenfolge beschrieben wird
 - (2) Choreographie
 - (1) Interaktionen mit (externen) Geschäftsakteuren
- (3) Die in den Geschäftsprozessbeschreibungen enthaltenen Informationen werden zur Identifikation der Services und deren Spezifikation benötigt

[C&M-Di-Ge11:62]

- (1) Ziel der serviceorientierten Architektur ist die IT-basierte Automatisierung der Geschäftsanwendungsfälle und damit der diese realisierenden Geschäftsprozesse.
- (2) An einem Geschäftsprozess sind ein oder mehrere Geschäftsakteure (engl. business actor) beteiligt. Hinsichtlich des Ablaufs ist zu unterscheiden, ob dieser innerhalb eines Geschäftsakteurs erfolgt oder zwischen zwei Geschäftsakteuren.
 - (2.1) Die Orchestrierung entspricht einer Beziehung zwischen zwei Aktivitäten, durch die angegeben wird, dass eine Aktivität nach der nächsten ausgeführt wird (Sequenz).
 - (2.2) Die Choreographie entspricht dem Austausch von Nachrichten zwischen den beteiligten Geschäftsakteuren. Die Nachrichten entsprechen in der serviceorientierten Serviceaufrufen und Serviceantworten.
- (3) Der interne Ablauf fließt ein in die Spezifikation des dynamischen Verhaltens des von diesem Geschäftsakteur bereitgestellten Services.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

BPMN (Business Process Model and Notation)



- (1) BPMN dient zur Modellierung von Geschäftsprozessen
- (2) Standard der OMG (aktuell in der Version 2.0):
 - (1) Grafische Notation von Prozessdiagrammen
 - (2) Semantik der Diagrammelemente
 - (3) Formales Meta-Modell (erst ab der Version 2.0)
 - (4) Abbildung nach BPEL
- (3) Folgende Arten von Geschäftsprozessen werden unterstützt:
 - (1) Private (Internal) Processes
 - (2) Abstract (Public) Processes
 - (3) Collaboration (Global) Processes

40

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

Hinweis: BPMN stand bis zum Erscheinen der Version 2.0 für "Business Process Modeling Notation", was ausdrückt, dass es sich ausschließlich um eine Notation ohne explizites Meta-Modell handelte.

(1) BPMN vereinheitlicht die Darstellung von Geschäftsprozessen, die bisher mit unterschiedlichen Werkzeugen und Notationen modelliert wurden. Die Geschäftsprozesse werden auf einer abstrakten Ebene modelliert, die auch für Manager und Geschäftsanalysten verständlich ist. Im Gegensatz zu BPEL enthalten BPMN-Prozessdiagramme keine technische Spezifikation, um die Prozesse direkt auf Systemen auszuführen.

(2) Der Standard wurde erstmals 2004 veröffentlicht. Auf die Version 1.2 [OMG-BPMN-1.2] folgte die aktuelle Version 2.0 [OMG-BPMN-2.0].

(2.2) Die Semantik ist semi-formal definiert.

(2.3) Das einheitliche Meta-Modell mit einem standardisierten Speicherformat ermöglicht die Austauschbarkeit von Prozessmodellen zwischen verschiedenen Tools und die Modelle können als Ausgangspunkt für eine modellgetriebenen Entwicklung dienen.

(2.4) Andere Organisationen bieten auch Abbildungen auf andere Ausführungssprachen an, z.B. auf XPDL.

(3) Aus [OMG-BPMN-1.2] und [Wh04].

(3.1) Diese Prozesse beschreiben Abläufe innerhalb einer Organisation.

(3.2) Bei dieser Art von Geschäftsprozess kommuniziert ein interner Prozess mit einem externen Partner. Im Gegensatz zum internen Prozess werden die Aktivitäten auf der Seite des externen Partners nicht näher spezifiziert. Es wird nur der Nachrichtenaustausch mit dem externen Partner spezifiziert.

(3.3) Es wird die Choreographie zwischen mehreren Geschäftsprozessen von verschiedenen Organisationen beschrieben (B2B-Szenarien).

B2B	Business-to-Business
OMG	Object Management Group
XML	Extensible Markup Language
XPDL	XML Process Definition Language

[OMG-BPMN-2.0] Object Management Group: Business Process Model and Notation (BPMN) 2.0 Request For Proposal, <http://www.bpmn.org/Documents/BPMN%202-0%20RFP%2007-06-05.pdf>

[OMG-BPMN-1.2] Object Management Group: Business Process Modeling Notation (BPMN) 1.2, <http://www.omg.org/spec/BPMN/1.2>

[Wh04] Stephen A. White: Introduction to BPMN, <http://bpmn.org/Documents/Introduction%20to%20BPMN.pdf>, May 2004.

BPMN: Objekte

(1) In BPMN werden zwei Arten von Objekten unterschieden

Flussobjekte (Flow Objects)	Ereignisse (Events): Start Intermediate End
	Aktivitäten (Activities): Aufgabe (Task) Teilprozess (Subprocess)
	Gateways: Exclusive (XOR) Parallel (AND) Inclusive (OR)
Verbindende Objekte (Connecting Objects)	<ul style="list-style-type: none"> → Sequenzfluss (Sequence Flow) ◇→ Bedingter Fluss (Conditional Flow) ← Voreingestellter Fluss (Default Flow) ○-----> Nachrichtenfluss (Message Flow)

(1) Die zwei Arten sind die Flussobjekte (engl. flow objects) und die Verbindenden Objekte (engl. connecting objects). Die grafische Notation ist aus dem Standard [OMG-BPMN-1.2] entnommen.

(Ereignisse, Events) Es gibt verschiedene Arten von Ereignissen. Mit Ereignissen kann z.B. das Eintreffen einer Nachricht, ein zeitliches Ereignis oder eine Ausnahme beschrieben werden. Der Standard definiert verschiedene Symbole, die innerhalb des Kreises angezeigt werden können und die die Art eines Ereignisses spezifizieren.

(Aufgabe Task) Eine Aufgabe ist eine atomare Aktivität.

(Teilprozess, Subprocess) Ein Teilprozess kann mit dem Pluszeichen expandiert werden um die enthaltenen Aktivitäten sichtbar zu machen.

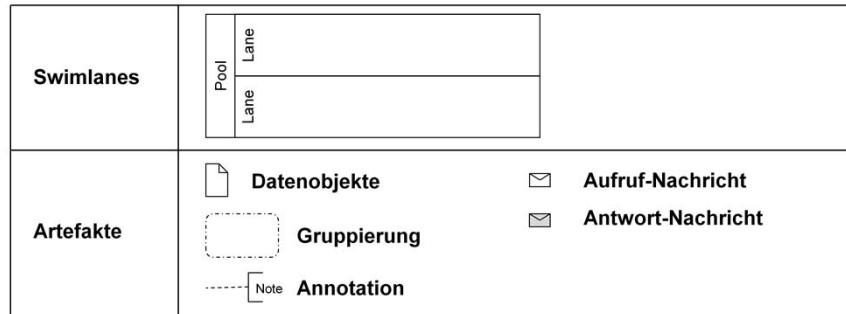
(Gateway) Ein Gateway ermöglicht den Prozessfluss zu verzweigen und zusammenzuführen. Zum Verzweigen und Zusammenführen werden die gleichen Gateways verwendet.

- Exclusive XOR: Es wird auf Basis einer Bedingung ein Zweig ausgeführt.
- Parallel: Die nachfolgenden Zweige werden parallel ausgeführt.
- Inclusive: Auf Basis einer Bedingung werden ein oder mehrere Zweige ausgeführt.

(Connection Objects) Connecting Objects verbinden Flussobjekte miteinander.

- Sequence Flow: Beschreibt den Prozessfluss zwischen zwei Flow Objects.
- Conditional Flow: Dieser Prozessfluss wird ausgeführt, wenn die zugehörige Bedingung wahr ist.
- Default Flow: Dieser Prozessfluss wird nur ausgeführt, wenn kein anderer Conditional Flow wahr ist.
- Message Flow: Verbindet Pools oder Aktivitäten in verschiedenen Pools und beschreibt den Nachrichtenaustausch. Der Message Flow ist besonders wichtig beim Modellieren von Public oder Collaboration Processes.

BPMN: Swimlanes und Artefakte



- (1) Die Swimlanes und die Artefakte bilden mit den zuvor eingeführten Flow Objects und Connection Objects die (vier) Kategorien von BPMN-Diagrammelementen

(Swimlanes) Es gibt zwei Arten von Swimlanes: Pool und Lane. Ein Pool beschreibt eine Organisation oder Rolle, die als ein Prozesspartner am Gesamtprozess beteiligt ist. Mittels Lanes kann ein Pool weiter unterteilt werden.

(Artefakte) Artefakte (engl. artifacts) erlauben es dem Modellierer, die grundlegende Notation von BPMN zu erweitern und den Geschäftsprozess mit zusätzlichem Kontext zu ergänzen.

(Datenobjekte) Ein Datenobjekt beschreibt ein Artefakt, das vom Geschäftsprozess bearbeitet wird.

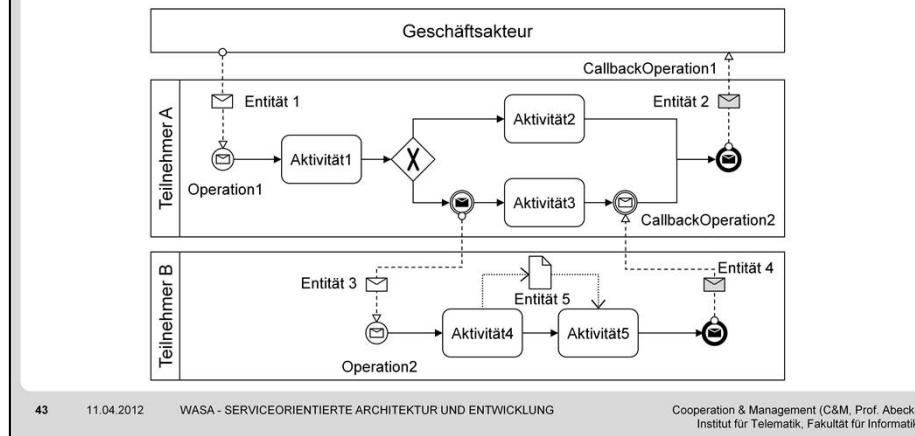
(Nachricht) Dienen dazu, die Interaktion zwischen Teilnehmern (jeweils durch einen Pool repräsentiert) zu beschreiben.

(Gruppierung) Gruppierungen können verwendet werden, um einzelne Elemente visuell zusammenzufassen.

(Annotation) Mit Annotationen können Teile eines Diagramms kommentiert werden.

Modellierung von Geschäftsprozessen mit BPMN

- (1) Ein Geschäftsprozess eines Teilnehmers wird als eine Abfolge von Aktivitäten beschrieben
- (2) Die Interaktion zwischen Teilnehmern erfolgt durch den Austausch von Nachrichten



43

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[C&M-Di-Ge11:63]

Das Modell ist ein verallgemeinertes BPMN-Kollaborationsdiagramm, das die internen Abläufe von zwei Teilnehmern sowie deren Interaktion und außerdem die Interaktion mit einem externen Geschäftsakteur beschreibt.

(1) Ein Teilnehmer wird in BPMN durch einen Pool, der den Namen dieses Teilnehmers trägt, repräsentiert. Teilnehmer können hierbei externe Geschäftsakteure oder Angestellte innerhalb des Geschäfts repräsentieren, wobei unter Angestellten innerhalb des Geschäfts sowohl konkrete Rollen als auch Organisationseinheiten verstanden werden können.

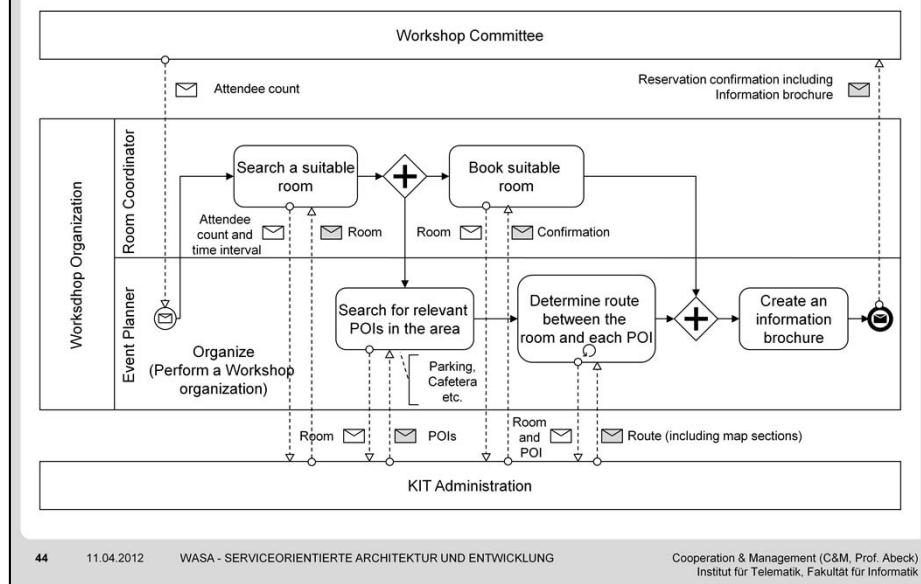
Die Abfolge von Aktivitäten kann ausschließlich (exklusiv X), einschließlich (Kreis) oder parallel (+) modelliert werden.

(2) Hierbei kann zwischen aufrufenden Nachrichten und Antwortnachrichten, die bspw. Callback-Operationen in asynchronen Aufrufen entsprechen, unterschieden werden. Erstere werden in Form eines weißen Umschlags (-> Entität 1) und letztere in Form eines grauen Umschlags (-> Entität 2) dargestellt.

(Entität 5) Datenobjekt, das ein Geschäftsobjekt bzw. eine Geschäftsentität repräsentiert und das zwischen Aktivitäten ausgetauscht wird.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

KCG: BPMN-Prozess zur Organisation eines Workshops auf dem KIT-Campus



44

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[C&M-DA-Bo11:4]

Ein Beispiel eines in BPMN modellierten Geschäftsprozesses, zu dessen systemtechnischen Unterstützung der KITCampusGuide (KCG) vorteilhaft genutzt werden kann, ist die Organisation eines Workshops auf dem Campus. Neben dem KCG, der für die Bestimmung aller für den Workshop relevanten POIs (PointOfInterest) und die zugehörige Wegewahl zuständig ist, übernimmt ein Facility-Management-System als ein zweites für die Unterstützung dieses Geschäftsprozesses benötigtes System die Reservierung und die Buchung der für den Workshop benötigten Räume. Somit ist für die Unterstützung dieses Geschäftsprozesses eine Anwendungsintegration erforderlich, was bei komplexeren Geschäftsprozessen ganz typisch ist.

(KIT Administration) Diese Rolle stellt die durch die beiden Systeme unterstützten Funktionen zur Raumbuchung, POI-Bestimmung und Wegewahl zur Verfügung.

(Workshop Committee) Dieses Gremium initiiert die Organisation eines Workshops auf dem KIT-Campus, indem eine Aufforderung mit der Anzahl der erwarteten Teilnehmer ("Attendee count") an den "Event planer" geschickt wird.

(Event Planer, Room Coordinator) Die Organisation des Workshops umfasst die Suche und das Buchen eines geeigneten Raums und die Suche der für den Workshop relevanten POIs sowie die Bestimmung der Wege zwischen dem Raum und diesen POIs. Die gefundenen Informationen in einem letzten Schritt des Workshop-Organisations-Prozesses übersichtlich in einer Broschüre aufbereitet.

[CM-DA-Bo11] Jaouad Bouras: Derivation of Service Implementations from Specified Service Designs, Diplomarbeit, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

LZ ENTWICKLUNG ÜA GESCHÄFTSANALYSE

- (1) Wie stehen die Geschäftsanalyse-Artefakte untereinander und mit dem Servicedesign in Beziehung?
- (2) Welchen Vorteil bietet OWL zur Beschreibung des Domänenmodells?
- (3) Welche zwei unterschiedliche Aspekte drückt die Beziehung zwischen einem Geschäftsakteur und einem Geschäftsanwendungsfall aus?
- (4) Was versteht man unter Orchestrierung bzw. Choreographie und welche Beschreibungselemente stellt BPMN hierfür bereit?
- (5) Welche Geschäftsakteure bzw. Organisationseinheiten sind an der Workshop-Organisation beteiligt und was ist deren Beitrag?

(1) Spezifikation der Services aufgrund der Geschäftsanforderungen

Geschäftsanforderungen

Servicekandidat

Servicekandidat

Servicekandidat

Servicekandidat Existenter Service

Spezifikation

Servicedesign

Angebote
ne Serviceschnittstelle

Servicekomponente

Benötigte
Serviceschnittstelle

Benötigte
Serviceschnittstelle

Einsatz von SoaML zur formalen Beschreibung
der im Servicedesign auftretenden Artefakte

[C&M-DI-Ge11:157]

(1) Das Servicedesign fokussiert die Spezifikation der Services. Aufbauend auf den Ergebnissen der Geschäftsanalyse werden seitens des IT-Architekten Servicekandidaten identifiziert und anschließend vollständige Servicespezifikationen entwickelt.

(Servicekandidat) Repräsentieren vorläufige Services und dienen zur Gruppierung von Operationskandidaten.

(SoaML) Diese Sprache "Service oriented architecture Modeling Language" wird wie UML von der OMG standardisiert und führt formale Beschreibungselemente zu Servicekandidaten, Servicekomponenten und Serviceschnittstellen ein.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

SoaML (Service oriented architecture Modeling Language)



- (1) Metamodell und UML-Profil zur Modellierung serviceorientierter Architekturen
- (2) Einführung von Modellierungselementen zur Abstraktion einer konkreten Implementierung
 - (1) Die Servicebeschreibung kann in einem nachfolgenden Schritt mittels konkreter Technologien umgesetzt werden
- (3) Der OMG-Standard ersetzt die bis zu diesem Zeitpunkt bestehenden proprietären Ansätze zur Servicemodellierung
- (4) SoaML ermöglicht die Modellierung auf zwei konzeptionellen Ebenen
 - (1) Die ServicesArchitecture dient der Beschreibung der Servicebeziehungen unabhängiger Teilnehmer
 - (2) Durch eine ParticipantArchitecture werden die Services eines einzelnen Teilnehmers im Detail beschrieben

47

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

SoaML ist ein OMG-Standard, der aus einem Aufruf (Request for Proposal RFP) zu einem "UML Profile and Metamodel for Services" (UPMS) entstanden ist [OMG-SoaML-2008-04-08:17/29] bzw. [OMG-SoaML-V1.0Beta:3/11]. Die aktuelle Version des Standards [OMG-SoaML-V1.0Beta] ist der nachfolgenden Beschreibung zugrunde gelegt.

- (1) Es werden bewusst die semantisch gleichwertigen Ansätze angeboten [C&M-Di-Ge11:18].
 - Profil, um UML2-Werkzeugherstellern eine effiziente Bereitstellung von SoaML zu ermöglichen
 - Metamodell, um zukünftige mit Profilen schwer zu erreichenden Erweiterungen zu ermöglichen
 - (2) Das mittels SoaML-Elementen erstellte Modell der serviceorientierten Architektur ist technologieunabhängig.
 - (2.1) Zur Umsetzung der in SoaML modellierten Serviceschnittstelle kann z.B. die Web-Service-Technologie, konkret WSDL, genutzt werden'.
 - (3) Ein solcher Ansatz wurde u.a. von IBM in Form eines "UML-Profile for Software Services" entwickelt.
 - (4) Anmerkung: Die zwei Ebenen haben eine gewisse Entsprechung zu der bei der Servicekomposition unterschiedenen Choreographie und Orchestrierung.
 - (4.1) Ein Teilnehmer (engl. participant) kann eine Organisation, ein System oder eine einzelne Softwarekomponente sein.
 - (4.2) Es wird zwischen Services unterschieden, die der Teilnehmer bereitstellt (engl. provides) oder benötigt (engl. requires).
- Die detaillierte Beschreibung der Services eines Teilnehmers erfolgt in zwei Schritten:
- (i) Abstrakte Beschreibung der Fähigkeiten (engl. capabilities)
 - (ii) Bereitstellung dieser Fähigkeiten durch konkrete Services [C&M-Di-Ge11:19].

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

[OMG-SoaML-2008-08-04] OMG: Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS), OMG document: ad/2008-08-04, 2008.

[OMG-SoaML-V1.0Beta] OMG: Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS), OMG Adopted Beta Specification, final publication on 15.11.09.

(1) **Capability**

(1) Funktionale Einheiten, die durch Operationen beschrieben werden

(2) **ServiceInterface**

(1) Serviceschnittstelle an Service- bzw. Anfrageendpunkten

(3) **Participant**

(1) Modelliert einen Teilnehmer einer SOA

(4) **ServiceContract**

(1) Verknüpfung von Teilnehmern, die durch ihre Schnittstellen beschrieben sind

(5) **ServicesArchitecture**

(1) Zusammenarbeit von zwei oder mehr Teilnehmern über Verträge

(1) Dieses Modellierungselement ist von der UML "Class" abgeleitet. Es handelt sich um Sammlungen von Funktionalitäten und Fähigkeiten, die in ihrer Gesamtheit ein bestimmtes Ergebnis erzielen können. Dabei wird von den Akteuren vollständig abstrahiert. Praktische Verwendung findet die "Capability" bei der Identifizierung von Servicekandidaten, ihrer Katalogisierung und Analyse. Dabei ist es möglich, funktionale Abhängigkeiten zu modellieren.

(2) Im Gegensatz zur "Capability" ist das "ServiceInterface" bindender Bestandteil eines Architekturmodells. Ein Teilnehmer kann über (in Form von "Capability" modellierte) Fähigkeiten verfügen – er kann sie allerdings nur mit Hilfe eines "ServiceInterface" an einem Serviceendpunkt bereitstellen. Ein "ServiceInterface" ist mit einer "Capability" in einer "expose"-Beziehung (freigeben, exponieren) verbunden.

(3) Ein "Participant" modelliert einen Teilnehmer an einer SOA in einem sehr weit gefassten Sinn. So kann es sich u.a. um eine Person, eine Organisation, ein System, eine Anwendung oder eine Komponente handeln – wichtig ist dabei nur, dass der Teilnehmer mit anderen Teilnehmern über Servicebereitstellung und Servicenutzung zusammenarbeitet um gemeinsam ein Ziel zu erreichen.

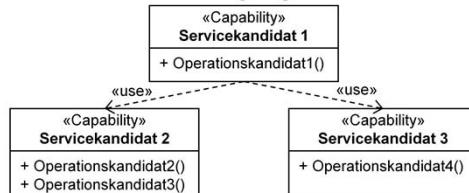
(4) Hierbei handelt es sich um eine Spezialisierung der UML "Collaboration". Der "ServiceContract" besteht aus zwei oder mehr Teilnehmern bzw. Rollen mit Schnittstellen-Typen, erlaubt sind hier das SoaML-Modellierungselement "ServiceInterface" oder das UML-Modellierungselement "Interface".

(2) (4) Sowohl in einem "ServiceInterface" als auch in einem "ServiceContract" kann Verhalten festgelegt werden, z.B. um gültige Aufrufsequenzen der Operationen zu spezifizieren. Wenn bereits das "ServiceInterface" zu einer Rolle Verhalten festgelegt hat, muss das im "ServiceContract" angegebene Verhalten damit vereinbar sein. Der Standard empfiehlt allerdings, nur entweder im "ServiceInterface" oder im "ServiceContract" Verhalten zu definieren.

(5) Dieses Modellierungselement bringt Teilnehmer mit ihren Fähigkeiten, angebotenen und benötigten Schnittstellen sowie Verträge mit ihren typisierten Rollen und spezifiziertem Verhalten zusammen und bietet so einen Überblick über eine gesamte Architektur. In einer "ServicesArchitecture" wird also ersichtlich, wie zwei oder mehr Teilnehmer über ihre Endpunkte zusammenarbeiten, um ein Ziel zu erfüllen.

Servicekandidaten in SoaML

- (1) Ziel dieses ersten Schritts der Designphase ist die Beschreibung benötigter Services
 - (1) Werden als Servicekandidaten bezeichnet
- (2) Ein Servicekandidat besteht aus Operationskandidaten
- (3) Servicekandidaten können in einer Benutzt-Abhängigkeit (<<use>>) zueinander stehen
- (4) In SoaML steht das Modellelement <<Capability>> zur Beschreibung von Servicekandidaten zur Verfügung



49

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[C&M-DI-Ge11:63]

(1.1) Die Bezeichnung des "Kandidaten" ist angelehnt an RUP/SOMA und wird auch von Erl verwendet. Es handelt sich um Vorschläge für Services und noch nicht implementierte Services. Ob ein Kandidat tatsächlich implementiert wird, muss zu einem späteren Zeitpunkt entschieden werden.

(2) Operationskandidaten werden auch als Fähigkeitskandidaten (engl. capability candidates) bezeichnet und sind, wie Servicekandidaten, Vorschläge für notwendige Operationen.

(3) Die <<uses>>-Beziehung drückt aus, dass ein Servicekandidat einen anderen benötigt, um seine Funktionalität zu erbringen [:64].

(4) <<Capability>> ist ein Stereotyp, der in SoaML aus einer Erweiterung der Metaklasse <<Class>> als Modellelement eingeführt wird, um eine Menge von Fähigkeiten zu gruppieren. Eine solche Gruppe von Fähigkeiten entspricht genau dem, was unter einem Servicekandidaten zu verstehen ist.

Wichtige Eigenschaften von Capabilities sind [OMG-SoaML-V1.0Beta]:

- (i) Spezifizieren eine kohäsive Menge von Funktionen oder Ressourcen, die ein Service anbieten könnte
- (ii) Repräsentieren allgemeine Funktionalität oder Eignungen eines Teilnehmers
- (iii) Lassen sich vernetzen (engl. networks of capabilities)
- (iv) Werden zur Identifikation von benötigten Services und zu deren Organisation in Katalogen benutzt
- (v) Werden durch UML Packages modelliert
- (vi) Werden in architekturellen Schichten organisiert

(Operationskandidat) Da eine <<Capability>> eine <<Class>> erweitert, lassen sich dieser Operationen zuordnen. Eine Operation entspricht einer einzelnen Fähigkeit und stellt somit ein Operationskandidat dar.

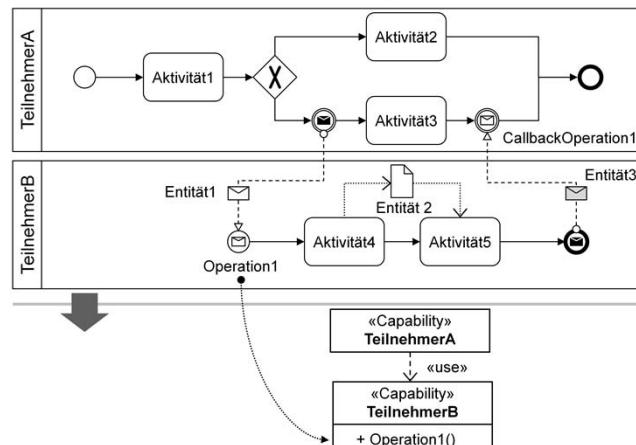
(<<use>>) Durch diesen Stereotyp wird eine Verwendungsabhängigkeit (engl. usage dependency) ausgedrückt.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

[OMG-SoaML-V1.0Beta] OMG: Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS), OMG Adopted Beta Specification, final publication on 15.11.09.

Ableitung von Servicekandidaten und Operationskandidaten

- (1) Servicekandidaten und Operationskandidaten lassen sich systematisch aus einem Geschäftsprozess ableiten



50 11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)

Institut für Telematik, Fakultät für Informatik

[C&M-Di-Ge11:65]

- (1) Die Geschäftsprozesse werden im Rahmen der Geschäftsanalyse aus den Geschäftsanwendungsfällen entwickelt. Als Sprache bietet sich die Business Process Model and Notation (BPMN) an.

(TeilnehmerA, TeilnehmerB) Für jeden BPMN-Pool im Geschäftsprozess wird ein Servicekandidat erstellt. Dieser Ableitungsschritt erfolgt in Anlehnung an RUP/SOMA [:66].

Erklärung: Ein Teilnehmer übernimmt eine Geschäftsrolle und repräsentiert dadurch genau eine Gruppe von Fähigkeiten.

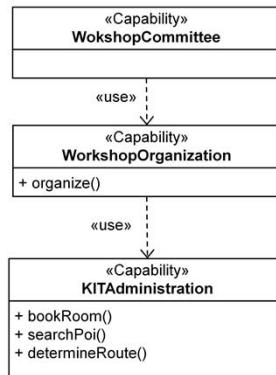
(<<use>>) Indem ein Teilnehmer (im Beispiel "A") einen anderen Teilnehmer ("B") durch Zusenden einer Aufrufnachricht ("Entität1") in Anspruch nimmt, steht dieser ("A") in einer Verwendungsabhängigkeit zu dem aufgerufenen Teilnehmer ("B").

(Operation1) Die Operationskandidaten ergeben sich aus der Betrachtung der Interaktion zwischen den Teilnehmern. Durch eine Aufrufnachricht ("Entität 1") nimmt ein Teilnehmer ("A") eine Operation (im Beispiel "Operation1") des aufgerufenen Teilnehmers ("B") in Anspruch. Diese Operation wird als Operationskandidat des Teilnehmers, der die Operation anbietet, aufgenommen ("Operation1()") [:66].

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

KCG: Service-/Operationskandidaten zur Workshop-Organisation

- (1) Ausgangspunkt der Ableitung ist der im Rahmen der Geschäftsanalyse entwickelte und in BPMN vorliegende Geschäftsprozess



[CM-DA-Bo11:81] , [C&M-DI-Ge11:158]

(1) In diesem Prozess wurden neben dem Studierenden, der die Auskunft über die Route zu einem Ziel auf dem Campus benötigt, drei weitere Geschäftsakteure in Form von BPMN-Pools identifiziert.

(<<Capability>>) Jeder dieser Geschäftsakteure bzw. Prozess-Teilnehmer wird gemäß dem zuvor beschriebenen Vorgehen zu einem Servicekandidaten.

(+ organize()) Am Beispiel dieses Operationskandidaten soll die Ableitung verdeutlicht werden: Im BPMN-Prozess tritt eine Interaktion zwischen dem "WorkshopCommittee" und der "WorkshopOrganization" auf, in der das "WorkshopCommittee" zur Ausführung der Aktivität "Organize (Perform a Workshop Organization)" eine Anfragenachricht mit dem "Ziel" der Route an die "WorkshopOrganization" sendet, wodurch eine Operation ausgelöst wird, die die gleiche Bezeichnung wie die Aktivität trägt. Als Antwortnachricht erhält das "WorkshopCommittee" die "Route inklusive Karte" zurück.

In dem die Servicekandidaten beschreibenden SoaML-Modell wird diese Interaktion in folgender Weise wiedergegeben:

- (i) Der Servicekandidat "WorkshopCommittee" steht in einer Benutzt-Beziehung (<<use>>) zum Servicekandidaten "WorkshopOrganization".
- (ii) Der Servicekandidat "WorkshopCommittee" beinhaltet den Operationskandidaten "organize()".

Die drei Operationskandidaten zur "KITAdministration" können aus den Entitäten ermitteln, die im BPMN-Prozess modelliert sind, abgeleitet werden. Eine systematische Ableitung würde die Modellierung des internen Ablaufs (also die Orchestrierung) des Geschäftsakteurs "KITAdministration" erfordern.

[CM-DA-Bo11] Jaouad Bouras: Derivation of Service Implementations from Specified Service Designs, Diplomarbeit, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

LZ ENTWICKLUNG ÜA SOAML UND SERVICEIDENTIFIKATION

- (1) In welcher Form liegt die Sprache SoaML vor und was wird durch diese geleistet?
- (2) Woraus entsteht das SoaML-Element <>Capability<> und wozu wird es im Servicedesign genutzt?
- (3) Woraus und wie lassen sich Servicekandidaten, deren Beziehungen und Operationskandidaten systematisch identifizieren?
- (4) Das Vorgehen ist am KCG-Beispiel der Workshop-Organisation an einem Operationskandidaten aufzuzeigen

Servicespezifikation

- (1) Die Servicespezifikation ist der zweite Schritt in der Servicedesign-Phase
 - (1) Erweiterung der Servicekandidaten zu einem vollständigen Servicedesign
- (2) Bestandteile eines Servicedesigns
 - (1) Serviceschnittstelle
 - (1) Beteiligte Rollen
 - (2) Bereitgestellte und erforderliche Operationen
 - (1) Ausgetauschte Nachrichten
 - (3) Interaktionsprotokoll
 - (2) Servicekomponente
 - (1) Bereitgestellte und erforderliche Services
 - (2) Internes Verhalten

[C&M-Di-Ge11:55, 17]

(1) Der Begriff der Spezifikation trägt hier die Bedeutung einer Tätigkeit analog zur Identifikation [:55].

(1.1) Das Ergebnis der Spezifikationstätigkeit ist das Servicedesign im Sinne eines in SoaML modellierten Artefakts.

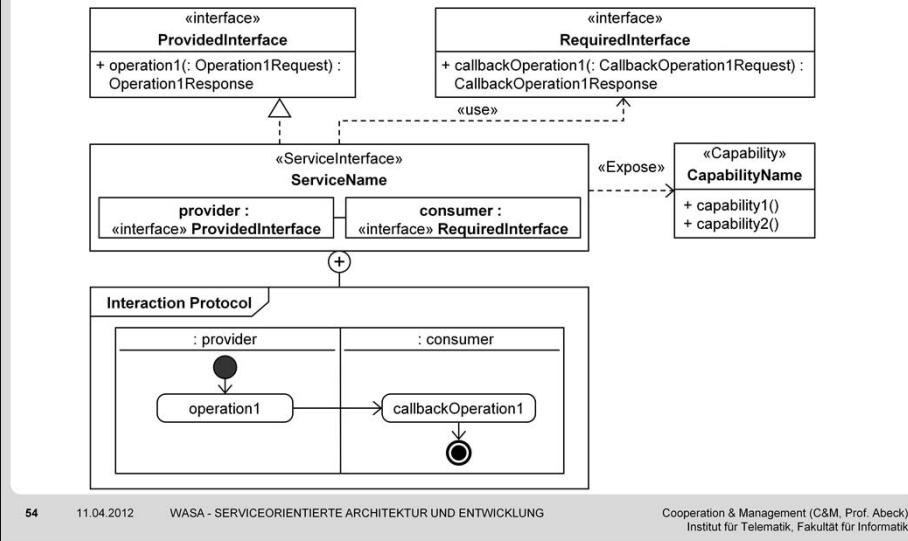
(2) Beide Bestandteile (Schnittstelle und Komponente) lassen sich in SoaML durch speziell hierfür vorgesehene Modellierungselemente (ServiceInterface und Participant) beschreiben.

(2.1) (2.2) Während die Serviceschnittstelle die Sicht des Servicenuzers auf den Service einnimmt, steht bei der Servicekomponente das Zusammenspiel des Services mit anderen Services, insbesondere mit den benötigten Services, im Fokus.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

SoaML: Serviceschnittstelle

- (1) Die Serviceschnittstelle ist Teil des Servicedesigns



54

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[C&M-DI-Ge11:21]

(1) Ein weiterer Teil ist die Servicekomponente, auf die später noch näher eingegangen wird.

(<<ServiceInterface>> ServiceName) Stereotyp, der die Metaklasse "Class" erweitert und mit dem sowohl bereitgestellte (engl. provided) als auch benötigte (engl. required) Services beschrieben werden (was bei der Modellierung der Servicekomponenten ersichtlich wird) [:20].

(provider, consumer) Die bei der Nutzung des Services auftretenden Rollen (Bereitsteller, Nutzer) werden als Eigenschaft (engl. Property) des <<ServiceInterface>> beschrieben und sind über die bereitgestellte bzw. genutzte Schnittstelle des Serviceinterfaces typisiert.

(<<interface>> ProvidedInterface) Durch das Standard-UML-Interface beschriebene Schnittstelle, die die von dem Service bereitgestellten Operationen beinhaltet und über eine Realisierungsbeziehung mit dem <<ServiceInterface>> verknüpft ist.

(<<interface>> RequiredInterface) Durch das Standard-UML-Interface beschriebene Schnittstelle, die die zur Nutzung des Services benötigten Operationen beinhaltet und über eine Verwendungsabhängigkeit (<<use>>) mit dem <<ServiceInterface>> verknüpft ist.

(callbackOperation1) Auf der Seite des Servicenuutzers (Serviceanfrager, Servicekonsument, engl. consumer) wird eine Operation nur dann benötigt, wenn es sich um einen asynchronen Aufruf handelt, was eine Callback-Operation erforderlich macht.

(Interaction Protocol) Beschreibt die zwischen den beteiligten Rollen ("provider", "consumer") ablaufenden Interaktionen.

In diesem Fall wird hierzu ein Aktivitätsdiagramm genutzt.

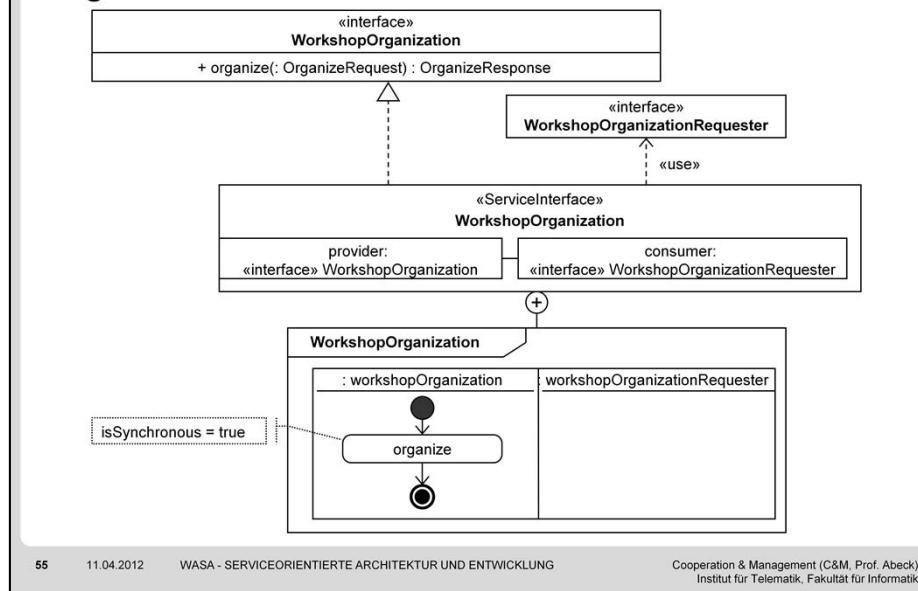
(operation1) Ist eine sog. "CallOperationAction", die besagt, dass die Operation "operation1" (vom "consumer") aufgerufen wurde und vom "provider" ausgeführt wird.

Hinweis: In der Antwortnachricht "Operation1Response" steht im asynchronen Aufruffall nur die Bestätigung, dass die Operation ausgeführt wird. Im synchronen Fall wird hierin direkt das Ergebnis an den Serviceaufrufer ("consumer") gesendet.

(callbackOperation1) Diese vom consumer bereitgestellte und ausgeführte Operation wird vom provider aufgerufen, um das Ergebnis der asynchron ausgeführten Operation "operation1" an den "consumer" zu senden. Die Übertragung des Ergebnisses erfolgt in der Antwortnachricht "CallbackOperation1Request".

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

KCG: Serviceschnittstelle für die Workshop-Organisation



[C&M-DA-Bo11:81], [C&M-DI-Ge11:164]

Die Serviceschnittstelle bedeutet ein Übergang von den aus dem Geschäft abgeleiteten Servicekandidaten hin zum technologieorientierten Servicedesign.

(WorkshopOrganizationRequester) Anstelle dieser Rollenbezeichnung könnte auch die Bezeichnung "WorkshopCommittee" gewählt werden, da gemäß BPMN-Prozess das "WorkshopCommittee" die Operation "organize()" von der "WorkshopOrganization" in Anspruch nimmt.

(isSynchronous = true) Die Operation wird synchron aufgerufen, d.h. das Ergebnis zu der an die "WorkshopOrganization" geschickten Anfragenachricht vom Typ "OrganizeRequest" steht direkt in der an das "WorkshopCommittee" gesendeten Antwortnachricht vom Typ "OrganizationResponse". Hinweis: Um das Beispiel einfach zu halten, wird davon ausgegangen, dass die Operation synchron aufgerufen wird, weshalb auf eine Callback-Operation verzichtet werden kann.

(OrganizeRequest, OrganizeResponse) Die Anfrage- und Antwortnachricht zu der Operation "organize". Zu beachten ist, dass eine nachrichtenorientierte Operation nur immer einen Ein- und Ausgabeparameter (bzw. -nachricht) haben kann.

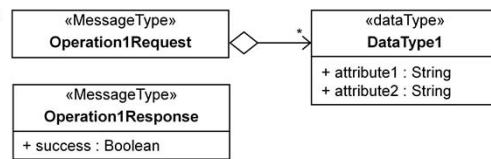
Eine Nachricht wird gebildet aus Datentypen. Hierauf wird nachfolgend eingegangen.

[CM-DA-Bo11] Jaouad Bouras: Derivation of Service Implementations from Specified Service Designs, Diplomarbeit, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

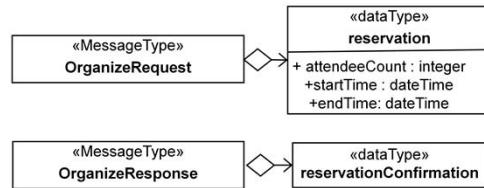
[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

KGC: Datentypen und Nachrichtentypen zur Workshop-Organisation

- (1) Die Nachrichten treten als Parameter in den bereitgestellten und benötigten Operationen innerhalb der Serviceschnittstelle auf



- (2) Die Bezeichnungen der genutzten Datentypen sind im Domänenmodell einzuführen



[C&M-DA-Bo11:Appendix34], [C&M-DI-Ge11:22, 164]

(1) In serviceorientierten Architekturen wird die nachrichtenorientierte Kommunikation gegenüber der klassischen Kommunikation mittels Remote Procedure Call (RPC) favorisiert. Beim RPC sind die Parametertypen direkt aus Datentypen gebildet und es bestehen keine expliziten Nachrichtentypen [:21]. Außerdem bestehen Unterschiede hinsichtlich der Kodierung, die beim RPC binär und herstellerspezifisch, beim Dokumenten-zentrierten Ansatz XML-basiert und damit standardisiert ist.

(<<MessageType>>) In SoaML wird zur Unterstützung des nachrichtenorientierten Ansatzes der Stereotyp <<MessageType>> als Erweiterung von <<dataType>> eingeführt. Zur Beschreibung der Nachrichten- und Datentypen stehen die in der UML üblichen Konstrukte wie Attribute, Aggregationen, Kompositionen, usw. zur Verfügung [:21].

(Operation1Request) Die Aggregationsbeziehung soll ausdrücken, dass die Anfragenachricht beliebig viele (Kardinalität *) Datentypen aggregieren kann.

(success) Das Attribut drückt aus, ob die Operation erfolgreich ausgeführt wurde oder z.B. aufgrund eines aufgetretenen Fehlers kein Resultat geliefert werden kann.

(2) Es wird davon ausgegangen, dass die für das Servicedesign erforderlichen Datentypen wie "reservation" und "reservationConfirmation" in einem für dieses Servicedesign spezifischen Paket abgelegt sind [:164]. Langfristig ist anzustreben, gemeinsame Datentypen für das Design verschiedener Services gemeinsam zu nutzen, d.h. eine Wiederverwendung der Datentypen zu erreichen.

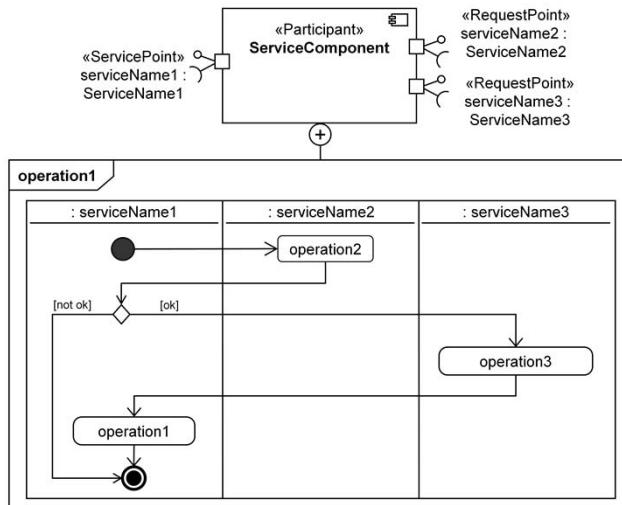
[CM-DA-Bo11] Jaouad Bouras: Derivation of Service Implementations from Specified Service Designs, Diplomarbeit, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

SoaML: Servicekomponente

- (1) Die Beschreibung der Servicekomponenten ist nach der Serviceschnittstelle der zweite Teil des Servicedesigns

- (2) In SoaML ist der Stereotyp des <<Participant>> zur Beschreibung von Servicekomponenten vorgesehen



57

11.04.2012

WASA - SERVICEORIENTIERTE ARCHITEKTUR UND ENTWICKLUNG

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

[C&M-DI-Ge11:22]

(1) Während bei der Serviceschnittstelle der Service aus der Perspektive des Servicenutzers beschrieben wurde, steht bei der Servicekomponente das Zusammenspiel des Services mit anderen Services, insbesondere mit den benötigten Services, im Fokus.

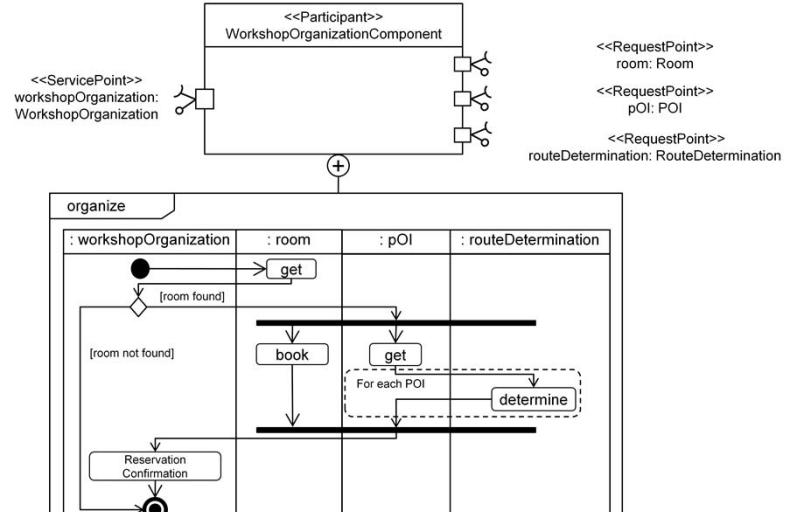
(2) Aus technischer Sicht kann ein "Participant" ein System, eine Anwendung oder eine Komponente sein, wobei Letzteres der hier zugrundeliegenden Betrachtungsweise am Nächsten kommt.

(<<ServicePoint>>) (<<RequestPoint>>) Stereotype, durch die ein von der Servicekomponente bereitgestellter bzw. benötigter Service beschrieben wird.

Im Beispiel wird der Service "serviceName1" bereitgestellt und die zwei Services "serviceName2" und "serviceName3" werden zu dessen Erbringung benötigt.

[C&M-DI-Ge11] Michael Gebhart: Qualitätsorientierter Entwurf von Anwendungsdiensten, Dissertation, Karlsruher Institut für Technologie (KIT), C&M (Prof. Abeck), 2011.

KCG: Servicekomponente für die Workshop-Organisation



Das SoaML-Diagramm zeigt die Servicekomponente, die einen Service "workshopOrganization" anbietet und zu dessen Realisierung drei Services ("room", "pOI" und routeDetermination") in Anspruch nimmt.

(<<ServicePoint>>) An diesem Servicepunkt wird der einzige Service angeboten.

(<<RequestPoint>>) Diese drei Punkte repräsentieren die für die Erbringung des Workshop-Organisations-Services in Anspruch zu nehmenden Services.

LZ ENTWICKLUNG ÜA SERVICESPEZIFIKATION

- (1) Die Bestandteile eines Servicedesigns und deren Beschreibung mittels SoaML-Modellelementen sind aufzuzählen
- (2) Wie wird ein asynchroner Aufruf im Servicedesign in SoaML modelliert?
- (3) Wie werden Serviceschnittstellen bei der Beschreibung einer Servicekomponente genutzt?