



Universität Augsburg
Prof. Dr. Hans Ulrich Buhl
Kernkompetenzzentrum
Finanz- & Informationsmanagement
Lehrstuhl für BWL, Wirtschaftsinformatik,
Informations- & Finanzmanagement

UNIA
Universität
Augsburg
University

Diskussionspapier WI-188

Identifikation und Gestaltung von Services - Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich

von

Veronica Winkler

Februar 2007

in: Wirtschaftsinformatik, 49, 4, 2007, S. 257-266.

Identifikation und Gestaltung von Services – Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich

Identification and design of services – proceeding and exemplary use in the financial services sector

Dr. Veronica Winkler

Universität Augsburg

Lehrstuhl für Betriebswirtschaftslehre, Wirtschaftsinformatik & Financial Engineering

Prof. Dr. Hans Ulrich Buhl

Universitätsstrasse 16

86135 Augsburg

Telefon: 0821-5984141

Fax: 0821-598-4225

E-Mail: veronica.winkler@wiwi.uni-augsburg.de

URL: <http://www.wi-if.de>

Identifikation und Gestaltung von Services – Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich

Kernpunkte:

An einen Service, der in einer Service-orientierten Architektur eines Unternehmens zur Verfügung gestellt werden soll, werden oftmals die folgenden fachlichen Anforderungen gestellt:

- Er soll übergreifend über mehrere Ausprägungen einer Problemstellung einsetzbar sein.
- Er soll die redundante Implementierung von Funktionalitäten weitgehend vermeiden.
- Er soll möglichst schnell und unkompliziert genutzt werden können.

Diesen Anforderungen kann bei der Entwicklung eines Service wie folgt entsprochen werden:

- Einen wichtigen Beitrag leistet die Identifikation der Services, d. h. die Beantwortung der Frage, welche Aktionen der Geschäftsprozesse eines Unternehmens durch einen Service realisiert werden sollten.
- Des Weiteren trägt dazu die geeignete fachliche Gestaltung der identifizierten Services bei.

Stichworte:

Service-orientierte Architektur, Serviceidentifikation, Servicegestaltung, Web Services, Finanzdienstleistungsbranche

Keywords:

Service-Oriented Architecture, identification of services, design of services, Web Services, financial services sector

Zusammenfassung:

Im Beitrag wird beschrieben, wie bei der Entwicklung von Services für eine Service-orientierte Architektur (SOA) vorgegangen werden kann. Im Vordergrund steht dabei – im Gegensatz zu den meisten bisher vorliegenden Veröffentlichungen zum Thema SOA – die fachliche Gestaltung und nicht die technische Umsetzung von Services. Der Beitrag beginnt mit einem Überblick über SOA und der Definition der Anforderungen, denen ein in einer SOA genutzter Service entsprechen soll. Anschließend wird betrachtet, wie vorgegangen werden kann, um diesen Anforderungen bei der Identifikation von Services, bei deren fachlicher Gestaltung und bei deren softwaretechnischer Umsetzung zu entsprechen. Abschließend erfolgen eine Zusammenfassung und eine kritische Würdigung der Ergebnisse.

Abstract:

This article describes how can be proceeded to develop services for a Service-Oriented Architecture (SOA). Contrary to the majority of the current publications concerning SOA, it is focussed the design and not the implementation of the services. The article starts with an overview over SOA and the definition of the requirements, which a service to be used in a SOA should meet. In the next step is considered, how can be proceeded to correspond to these requirements during the identification, the design and the implementation of services. The article ends with a summary and a critical appreciation of the results.

1 Einleitung

Die Finanzdienstleistungsbranche befindet sich in einem ständigen Wandel: So ändern sich häufig gesetzliche Rahmenbedingungen, des Weiteren ist die Produktpalette – um konkurrenzfähig zu bleiben – laufend um innovative Angebote zu erweitern. Folglich bestehen hohe Anforderungen an die Flexibilität der Systeme, die bei den derzeit häufig vorliegenden monolithischen Anwendungen nicht gegeben ist. Ein weiteres Problem ist, dass in der historisch gewachsenen Systemlandschaft und damit in der Vielzahl der heterogenen Anwendungen, die sich bei einem Finanzdienstleister im Einsatz befinden, bisweilen gleiche Funktionalitäten redundant und zum Teil inkonsistent implementiert sind. Deutlich wird dies bspw. bei der vielfach genutzten Funktionalität der Nullstellenbestimmung. Immer wenn in einer Anwendung Produktparameter, wie bspw. Anlagedauern oder Sparbeiträge, individuell festzulegen sind oder eine Optimierung durchzuführen ist, muss die Nullstelle einer finanzwirtschaftlichen Gleichung – i. d. R. mit Hilfe von Näherungsverfahren – bestimmt werden [BDHN77, 24]. So kommen für die Berechnung eines Effektivzinses bspw. sowohl Systeme, die diesen nach Preisangabenverordnung (PAngV) berechnen, zentrale Beratungsanwendungen als auch dezentral vorliegende Excelsheets in Frage. Diese verwenden jedoch unterschiedliche Näherungsverfahren und Parameter, was zum Teil unterschiedliche Ergebnisse – bspw. 5,12%, 5,15% und 5,20% – bedingt. Auf den ersten Blick erscheinen die Unterschiede vernachlässigbar gering. Allerdings wirken sich auch geringe Differenzen bei entsprechend langer Laufzeit und hohen Beträgen deutlich auf das Anlageergebnis aus. Da die Gründe für die Inkonsistenz nicht nachvollziehbar sind, führt diese sowohl zu Unverständnis bei den Kunden als auch Beratern.

Ein neuer Architekturansatz, der eine Lösung der genannten Probleme verspricht und in der Finanzdienstleistungsbranche immer größere Verbreitung erfährt, ist die Service-orientierte Architektur (SOA) [BCOZ04, 136ff; ToSt05, 509ff]. Im Gegensatz zu den derzeit häufig vorliegenden monolithischen Anwendungen bestehen Anwendungen in einer SOA nicht isoliert voneinander, sondern setzen sich zum Ausführungszeitpunkt dynamisch aus einzelnen Services und deren Kombinationen zusammen. Für die Finanzdienstleistungsbranche wäre die Umsetzung einer SOA somit von großem Vorteil, da in verschiedenen und heterogenen Anwendungen genutzte Funktionalitäten als wiederverwendbare und standardisierte Services verfügbar wären und damit die o. g. Redundanz- und Inkonsistenzprobleme vermieden werden könnten.

Im folgenden Beitrag wird dargestellt, wie bei der Identifikation, Gestaltung und softwaretechnischen Umsetzung von Services, die im Rahmen einer SOA zur Verfügung gestellt werden sollen, vorgegangen werden kann. Zur beispielhaften Veranschaulichung der Aus-

führungen dient die Finanzdienstleistungsbranche, da sie ein traditionelles Anwendungsgebiet der Wirtschaftsinformatik darstellt und SOA wie beschrieben von hoher Relevanz für sie ist. Als konkretes Beispiel wird hierbei die bei Finanzdienstleistern vielfach genutzte Funktionalität zur Bestimmung einer Nullstelle verwendet. Dabei fokussiert die Arbeit im Gegensatz zum Großteil der bestehenden Literatur die fachliche, nicht die technische Seite der Entwicklung von Services. Der Beitrag beginnt mit einem Überblick über SOA und einer Abgrenzung der Arbeit von anderen Veröffentlichungen in diesem Themenbereich. Davon ausgehend lassen sich die Anforderungen an einen in einer SOA genutzten Service aufstellen. Wie diesen bei der Identifikation, fachlichen Gestaltung und softwaretechnischen Umsetzung von Services entsprochen werden kann, ist Inhalt der Kapitel 3 bis 5. Das beschriebene Vorgehen für die Identifikation und Gestaltung von Services basiert dabei auf Erfahrungen aus der Mitarbeit an Projekten bei verschiedenen deutschen Finanzdienstleistern. Hier wurden monolithische Systeme auf eine Service-orientierte Architektur bspw. in Bereichen wie dem Portfoliomanagement oder der Kreditabwicklung umgestellt. In Kapitel 6 erfolgen eine Zusammenfassung und kritische Würdigung der Ergebnisse. Dabei wird im Beitrag davon ausgegangen, dass die betrachteten Services nur innerhalb des Unternehmens bereitgestellt und genutzt werden (Intra Enterprise Services) und diese auch nicht auf oder mit kritischen Kundendaten operieren.

2 SOA im Überblick und Anforderungen an Services

SOA „[...] sind das abstrakte Konzept einer Softwarearchitektur, in deren Zentrum das Anbieten, Suchen und Nutzen von so genannten [Services] über ein Netzwerk steht“ [DJMZ05, 7], die jeweils eine bestimmte Funktionalität bieten. Wie eingangs erwähnt, setzen sich Anwendungen in einer SOA zum Ausführungszeitpunkt dynamisch aus einzelnen Services oder deren Kombinationen zusammen („lose Kopplung“). Dazu interagieren Servicenutzer – meist eine Anwendung oder ein Service –, Serviceanbieter und Servicevermittler. Die Kommunikation zwischen den Beteiligten erfolgt über offene Standards, wodurch sich die Services unabhängig von Hard- und Software, Programmiersprache und Betriebssystem einsetzen lassen. Weitere Details hierzu finden sich bei der softwaretechnischen Umsetzung von Services in Abschnitt 5.1. [DJMZ05, 12ff]

Ein wichtiger Vorteil dieser Architektur ist damit die Modularität: Die im Unternehmen genutzten Funktionalitäten, wie bspw. bestimmte Berechnungen, werden als Services zur Verfügung gestellt. Die Services kapseln die Implementierungsdetails und bieten, wenn entsprechend konstruiert, klar definierte, minimale und standardisierte Schnittstellen, sodass sie in den heterogenen Anwendungen eines Unternehmens verwendbar sind. Damit verringert sich nicht nur der Entwicklungs- und Wartungsaufwand. Weitere positive Effekte sind die Stan-

dardisierung und die Wiederverwendbarkeit der im Unternehmen genutzten Funktionalitäten und damit die Vermeidung inkonsistenter Ergebnisse.

Die meisten Veröffentlichungen zu SOA thematisieren dabei Fragestellungen aus IT-Sicht, welche die konkrete Umsetzung von SOA unter Verwendung bestimmter Technologien wie Web Services oder die Nutzung von Grid-Technologien in diesem Kontext diskutieren, oder geben einen Überblick über SOA [ACKM03, 1ff; DJMZ05, 1ff; EbFi03, 1ff; HuSi05, 75ff; Papa03, 3ff; Poci06, 1ff; RiHS05, 413ff; ScWe06, 71ff]. Ein weiterer Schwerpunkt liegt auf der Umsetzung nicht funktionaler Anforderungen an Services, wie Servicequalität oder Preisgestaltung [BHMS06, 268ff; GoKG03, 121ff; KeLu03, 57ff]. Schließlich findet sich noch eine Vielzahl von Publikationen, welche die Umsetzung von SOA bei einem bestimmten Unternehmen beinhalten (z. B. [BCOZ04, 136ff; ToSt05, 509ff]). Wenig Beachtung erfahren bisher dagegen die in diesem Beitrag fokussierten Fragen nach der geeigneten Identifikation (vgl. Kapitel 3) und fachlichen Gestaltung (vgl. Kapitel 4) von Services. Eine Abgrenzung der hierzu vorliegenden Literatur findet sich in Kapitel 3. Deren Beantwortung stellt jedoch den ersten und grundlegenden Schritt bei der Umsetzung einer SOA in einem Unternehmen dar und trägt wesentlich dazu bei, ob die an eine SOA gestellten Erwartungen erfüllt werden. Dabei lassen sich – basierend auf den Ausführungen zu SOA – die folgenden Anforderungen an einen Service aufstellen:

- (A1) Der Service soll übergreifend über mehrere Ausprägungen einer Problemstellung in einer Vielzahl von heterogenen Anwendungen eines Unternehmens einsetzbar sein. Dadurch kann sowohl die bestehende Heterogenität an verwendeten Vorgehensweisen und Implementierungen für die gleiche Problemstellung in einem Unternehmen als auch die Inkonsistenz der Ergebnisse reduziert werden.
- (A2) Des Weiteren soll ein Service so gestaltet sein, dass eine redundante Implementierung der gleichen Funktionalität weitgehend vermieden wird.
- (A3) Außerdem soll eine schnelle und unkomplizierte Einbindung eines Service in bestehende Anwendungen gewährleistet sein, da nur dann das Ziel von SOA, dass Services in einer Vielzahl von Anwendungen genutzt werden, realistisch erreichbar ist.

Wie bereits erwähnt, betrachten bisherige Veröffentlichungen zu SOA vorwiegend die bei der Umsetzung von Services verwendeten Technologien. Um aber im Ergebnis einen Service zu erhalten, der den genannten Anforderungen entspricht, müssen sowohl der Service selbst als auch das Softwaremodul, in dem der Service umgesetzt ist, gemäß diesen Anforderungen gestaltet sein. Wie hierbei vorgegangen werden kann, ist Inhalt der folgenden Kapitel.

3 Identifikation von Services in einem Unternehmen

In einer SOA setzen sich Anwendungen aus einzelnen Services zusammen und auch Services selbst können wiederum aus mehreren Services bestehen. Um in einem Unternehmen die vielfach bestehenden monolithischen Anwendungen SOA-gemäß umzugestalten, ist es damit zunächst erforderlich, die Services und deren Granularität zu bestimmen, d. h. welche Funktionalität soll durch eine Service realisiert werden.

Bisher werden derartige Fragestellungen in der Literatur insbesondere für die Identifikation von Komponenten untersucht. Da an die Identifikation von Services aber ähnliche Anforderungen – wie die Kapselung der Funktionalität, die Ansprechbarkeit über eine Schnittstelle etc. – bestehen, können einige der für Komponenten gewonnenen Erkenntnisse auch für die Identifikation von Services Anwendung finden. Aufgrund der Ähnlichkeit stellt sich die Frage, ob nicht anstelle einer SOA auch eine komponentenorientierte Architektur geeignet wäre, um die in der Einleitung geschilderten Probleme zu lösen. Hierzu lässt sich sagen, dass Komponenten zwar viele der genannten Anforderungen erfüllen, diese im Gegensatz zu den Services aber weder plattform- noch programmiersprachenunabhängig und damit Services für den Einsatz in der heterogenen Systemlandschaft eines Finanzdienstleisters besser geeignet sind (vgl. Kapitel 5.1) [MSJL06, 51; SzGM02, 224].

Die Identifikation von Komponenten ist bereits seit den siebziger Jahren Gegenstand der Diskussion in der Literatur [Parn72, 1053ff]. Veröffentlichungen zu diesem Thema beinhalten häufig einen Überblick über die einzelnen Aspekte bei der Identifikation von Komponenten oder der Identifikation zugrunde liegende Vorgehensweisen [JCIR01, 183ff; WaXZ05, 229ff; XuYI03, 1213ff; ZaZP03, 1062ff]. Des Weiteren wird die Granularität von Komponenten untersucht [Aier06, 269ff; BeMü99, 57ff] oder die Vorteilhaftigkeit einzelner Vorgehensweisen an Beispielen demonstriert [AIDi06, 1ff]. Grundlage für die folgenden Ausführungen bildet dabei v. a. die Veröffentlichung von [WaXZ05, 229ff].

Bild 1 gibt einen Überblick über die Schritte, die für die Identifikation der Services durchgeführt und im Weiteren genauer beschrieben werden. Sind die Services bestimmt, kann anschließend deren fachliche Gestaltung erfolgen (Kapitel 4).

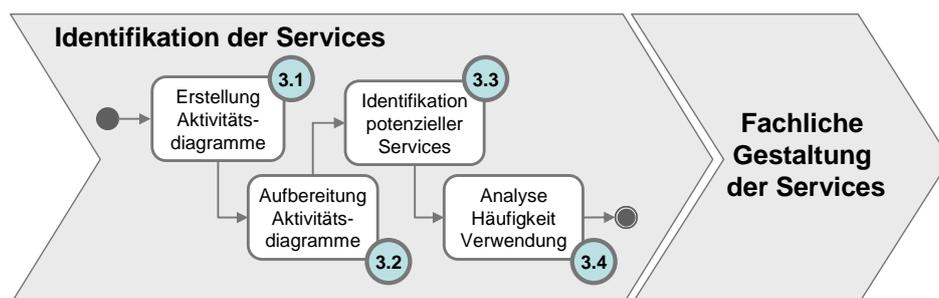


Bild 1 Schritte zur Identifikation von Services

Schritt 3.1: Erstellung der Aktivitätsdiagramme

Ausgangspunkt für die Identifikation der Funktionalitäten, die als Services angeboten werden sollen, sind die Geschäftsprozesse eines Unternehmens. Diese werden zunächst als Aktivitätsdiagramme modelliert. Hierfür sind die vorliegenden Geschäftsprozesse stufenweise zu analysieren, um herauszufinden, aus welchen Aktionen diese bestehen und in welcher Reihenfolge die Aktionen aufeinanderfolgen. Um dies systematisch durchzuführen, ist es hilfreich, schrittweise von einer sehr groben zu einer feineren Granularität überzugehen. Eine allgemein gültige Untergrenze für die Granularität kann dabei nicht vorgegeben werden, da diese in Abhängigkeit des betrachteten Unternehmens und Anwendungsszenarios sehr unterschiedlich ausfallen kann [WaXZ05, 235; BeMü, 66].

Bild 2 zeigt einen Ausschnitt der in den Geschäftsprozessen eines Finanzdienstleisters enthaltenen Aktionen. Dabei wurde aus Übersichtlichkeitsgründen zum einen auf eine Modellierung aller Aktionen und zum anderen – bis auf die für unser Beispiel relevante Aktion „Bestimmung einer Nullstelle“ – auf die Modellierung der Abfolge der Aktionen verzichtet. So liegen auf oberster Stufe bspw. die Aktionen „Durchführung einer Anlageberatung“, „Durchführung einer Finanzierungsberatung“ oder „Durchführung einer Immobilienberatung“ vor. Die Aktion „Durchführung einer Anlageberatung“ untergliedert sich u. a. in die Aktionen „Erfassung von Kundendaten“, „Anbieten möglicher Portfolios“ und „Abschluss eines Portfolios“. Die Aktion „Anbieten möglicher Portfolios“ lässt sich in „Bestimmung effizienter Portfolios“ und „Auswahl geeigneter Portfolios“ aufspalten. Auch diese lassen sich weiter zerlegen. So kann eine Aktion innerhalb der „Bestimmung effizienter Portfolios“ und der „Auswahl geeigneter Portfolios“ die „Bestimmung einer Nullstelle“ sein, um den Effektivzins der Portfolios oder die erforderliche Anlagedauer zu berechnen. Aber auch diese Aktion lässt sich untergliedern: In die „Ermittlung der Startwerte“, die „Festlegung der Abbruchkriterien“ und in die „Durchführung des Näherungsverfahrens“.

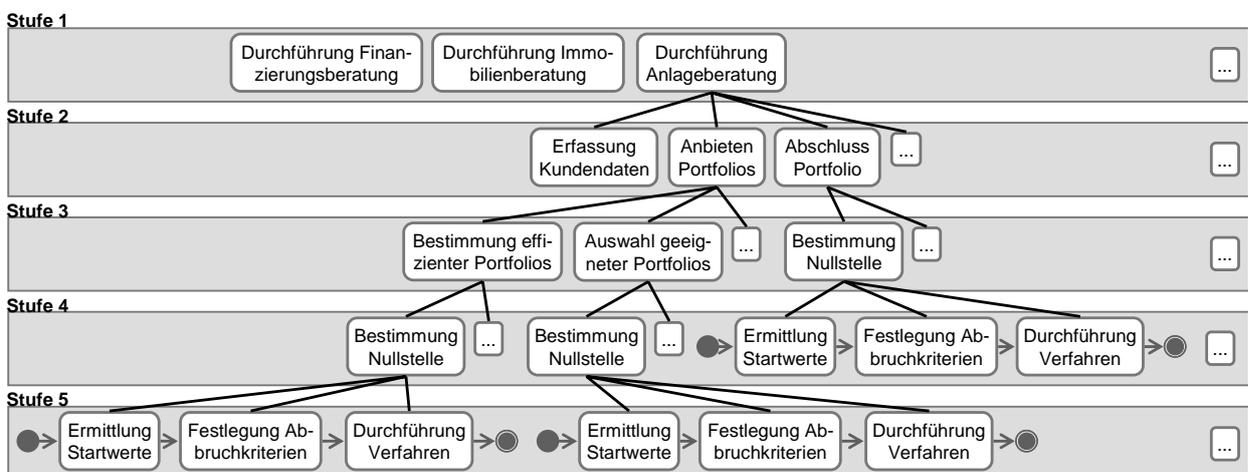


Bild 2 Ausschnitt von Aktionen bei einem Finanzdienstleister

Schritt 3.2: Aufbereitung der Aktivitätsdiagramme

Wie aus Bild 2 ersichtlich wird, sind in den Aktivitätsdiagrammen zum Teil gleiche Aktionen mehrmals modelliert. Um die Übersichtlichkeit zu erhöhen und damit die Identifikation der Services zu erleichtern, erfolgt im nächsten Schritt eine Aufbereitung der für die einzelnen Granularitätsstufen vorliegenden Aktivitätsdiagramme: Mehrmals modellierte gleiche Aktionen werden zusammengefasst. Bestehen bleiben die Beziehungen zwischen den Aktionen, d. h. in welcher anderen Aktion die betrachtete Aktion Verwendung findet, und deren Reihenfolge, d. h. in welcher Abfolge die Aktionen durchzuführen sind. An dieser Stelle ist anzumerken, dass bei der Erstellung der Aktivitätsdiagramme (Schritt 3.1.) auf eine einheitliche Benennung der Aktionen zu achten ist. Da für die Aktionen keine Semantik hinterlegt ist, wäre es sonst schwierig bis unmöglich, mehrmals modellierte Aktionen zu identifizieren.

Wie in Bild 3 dargestellt, werden die mehrmals modellierten Aktionen „Bestimmung einer Nullstelle“ und die in dieser Aktion enthaltenen und ebenfalls mehrmals abgebildeten Aktionen „Ermittlung der Startwerte“, „Festlegung der Abbruchkriterien“ und „Durchführung des Näherungsverfahrens“ zusammengefasst.

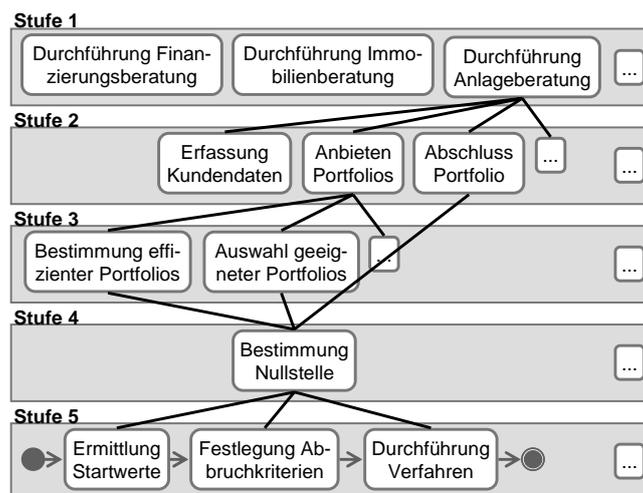


Bild 3 Aufbereitete Aktivitätsdiagramme

Schritt 3.3: Identifikation potenzieller Services

Um ausgehend von den vorliegenden Aktivitätsdiagrammen feststellen zu können, welche Aktionen sich als Service eignen, werden die an einen Service gestellten Anforderungen betrachtet. Dabei finden in diesem Schritt ausschließlich die Anforderungen (A2) und (A3) Berücksichtigung. Ob der sich ergebende Service übergreifend über mehrere Ausprägungen einer Problemstellung einsetzbar ist (Anforderung (A1)), hängt dagegen insbesondere von der fachlichen Gestaltung der Services (vgl. Kapitel 4) ab. Bezüglich der Anforderungen (A2) und (A3) lässt sich Folgendes feststellen:

- Anforderung (A3) fordert eine möglichst unkomplizierte und schnelle Einbindung von Services in eine Anwendung. Einen wichtigen Beitrag hierzu kann die Kapselung mehrerer Aktionen in einem Service leisten. Dadurch verringert sich zum einen die Zahl der aufzurufenden Services in der jeweiligen Anwendung, zum anderen ist keine Logik für deren Abfolge zu implementieren. Dabei sollten insbesondere die Aktionen zu einem Service zusammengefasst werden, welche immer gemeinsam Verwendung finden.

Bei der Analyse der Aktionen in Bild 3 lassen sich die Aktionen „Ermittlung der Startwerte“, „Festlegung der Abbruchkriterien“ und „Durchführung des Näherungsverfahrens“ in einem Service „Bestimmung einer Nullstelle“ zusammenfassen, da diese ausschließlich gemeinsam Verwendung finden. Würden diese nicht in einem Service gekapselt, sondern als eigene Services angeboten, müsste überall, wo eine Nullstelle zu bestimmen ist, nicht ein Service, sondern drei aufgerufen und zusätzlich mehrmals definiert werden, in welcher Abfolge die drei Services Verwendung finden.

- Um der Anforderung (A2) zu entsprechen, werden alle Aktionen ermittelt, die in verschiedenen Aktionen einer größeren Granularitätsstufe enthalten sind. Nachdem diese dort redundant zu implementieren wären, stellen diese Aktionen potenzielle Services dar.

So findet die Aktion „Bestimmung einer Nullstelle“ in unterschiedlichen Aktionen Verwendung und sollte deshalb als Service angeboten werden.

Schritt 3.4: Analyse der Häufigkeit der Verwendung

Schließlich wird die Häufigkeit der Verwendung der potenziellen Services ermittelt. Als Services sollen insbesondere diejenigen in Schritt 3.3 identifizierten Services bereitgestellt werden, die häufig Verwendung finden. D. h. zum einen, dass diese in vielen verschiedenen Aktionen enthalten sind. Hierfür können die im Schritt 3.2 aufbereiteten Aktivitätsdiagramme herangezogen werden. Zum anderen sollte untersucht werden, wie häufig diese Funktionalitäten innerhalb eines bestimmten Zeitraumes (z. B. einem Monat) im Unternehmen genutzt werden. Dabei kann eine Kennzahl, was „häufig“ bedeutet, nicht allgemein vorgegeben werden, sondern wird für das jeweilige Unternehmen individuell festgelegt.

Wie in Bild 3 dargestellt, findet die Nullstellenbestimmung lediglich in drei übergeordneten Aktionen Verwendung. Allerdings ist anzumerken, dass Bild 3 nur einen Ausschnitt aller Aktionen eines Finanzdienstleisters enthält. Hinsichtlich der Häufigkeit der Nutzung dieser Funktion im Unternehmen lässt sich annehmen, dass die Nullstellenbestimmung, gemäß der Ausführungen in der Einleitung, eine bei einem Finanzdienstleister vielfach durchgeführte Aktion darstellt. So werden bei einem großen Finanzdienstleister täglich rund 25.000 Beratungen durchgeführt, in denen jeweils ca. 5 Nullstellen zu bestimmen sind.

4 Fachliche Gestaltung von Services

Nachdem die in Kapitel 3 dargestellten Schritte durchgeführt und damit geeignete Services identifiziert wurden, kann die fachliche Gestaltung der Services erfolgen. Ein mögliches Vorgehen hierfür wird im Folgenden beschrieben (einen Überblick über die einzelnen Schritte gibt Bild 6).

Wie in Kapitel 3 beschrieben, setzt sich ein Service aus einer Aktion oder mehreren zusammen. Zur Entwicklung eines Service wird die fachliche Funktionalität für die Durchführung jeder im Service enthaltenen Aktion spezifiziert. Dabei lassen sich zwei Fälle unterscheiden:

- Für die Durchführung einer Aktion existieren dokumentierte Vorgehensweisen.
- Es existieren keine dokumentierten Vorgehensweisen und die fachliche Funktionalität für die Durchführung der Aktion ist vom Unternehmen selbst zu entwickeln.

Schritt 4.1: Identifikation möglicher Vorgehensweisen

Im ersten Schritt wird für jede in den Aktivitätsdiagrammen enthaltene Aktion ermittelt, ob auf dokumentierte Vorgehensweisen zurückgegriffen werden kann, welche die in der Aktion gewünschte Funktionalität realisieren. Als Ergebnis können folgende Fälle vorliegen:

- **Fall 4.1'**: Es existieren keine dokumentierten Vorgehensweisen. Folglich ist die fachliche Funktionalität selbst zu spezifizieren.
- **Fall 4.1''**: Es liegt eine dokumentierte Vorgehensweise vor.
- **Fall 4.1'''**: Es stehen mehrere dokumentierte Vorgehensweisen zur Auswahl.

Bspw. kann für die im Service „Bestimmung einer Nullstelle“ enthaltene Aktion „Durchführung des Näherungsverfahrens“ auf dokumentierte Vorgehensweisen zurückgegriffen werden. Dabei finden im Rahmen dieser Arbeit die folgenden in der Literatur geläufigen Näherungsverfahren Verwendung [Enge96, 1ff]: Intervallschachtelung, Newton-Verfahren, Pegasus-Verfahren, Regula Falsi und Sekantenverfahren. Dementsprechend liegt Fall 4.1''' vor. Für die „Ermittlung der Startwerte“ existieren dagegen keine dokumentierten Vorgehensweisen (Fall 4.1'). Folglich ist selbst zu spezifizieren, wie hierbei geeignet vorgegangen werden kann.

Schritt 4.2: Auswahl geeigneter Vorgehensweisen

Stehen für die Durchführung einer Aktion mehrere dokumentierte Vorgehensweisen zur Auswahl (Fall 4.1'''), wird im nächsten Schritt die für die Nutzung im Service am besten geeignete Vorgehensweise identifiziert.

Schritt 4.2a: Klärung der Voraussetzungen für die Durchführung der Vorgehensweisen

Zuerst werden die Voraussetzungen für die Durchführung der Vorgehensweisen geklärt. Dabei können die folgenden Fälle eintreten:

- **Fall 4.2a'**: Die Vorgehensweisen besitzen unterschiedliche, sich ausschließende Voraussetzungen. Damit ist keine Auswahl erforderlich, da für die jeweils gegebenen Voraussetzungen immer nur eine Vorgehensweise in Frage kommt.
- **Fall 4.2a''**: Es liegen keine oder zumindest sich nicht ausschließende Voraussetzungen für die Durchführung der Vorgehensweisen vor. D. h. unabhängig von den jeweils vorliegenden Voraussetzungen können mehrere Vorgehensweisen eingesetzt werden.

Für die Nullstellenbestimmung liegt Fall 4.2a'' vor. So kann zwar das Newton-Verfahren nur genutzt werden, wenn sich die Startwerte nahe genug an der Nullstelle befinden und keine Wendepunkte und lokalen Extrema vorliegen [Enge96, 27]. Allerdings stehen anstelle dessen immer auch die anderen Näherungsverfahren zur Auswahl, da diese keine Voraussetzungen für die Durchführung erfordern.

Schritt 4.2b: Auswahl der Vorgehensweisen anhand von Bewertungskriterien

Liegt Fall 4.2a'' vor, wird im nächsten Schritt eine Bewertung und darauf basierend eine Auswahl und evtl. ein Ausschluss ineffizienter Vorgehensweisen vorgenommen. Hierfür erfolgt eine Aufstellung von Bewertungskriterien an die Vorgehensweisen. Diese unterscheiden sich i. A. in Abhängigkeit der betrachteten Aktion, für deren Durchführung die Vorgehensweisen eingesetzt werden. Dabei können sich die folgenden Fälle ergeben:

- **Fall 4.2b'**: Es liegt eine dominante Vorgehensweise vor, die gegenüber den anderen Vorgehensweisen hinsichtlich aller Bewertungskriterien als mindestens gleichwertig zu beurteilen ist und diese hinsichtlich mindestens eines Bewertungskriteriums übertrifft. Da i. A. konkurrierende Bewertungskriterien vorliegen, trifft dies selten zu.
- **Fall 4.2b''**: Es kommen auch nach der Bewertung noch mehrere Vorgehensweisen für den Einsatz im Service in Frage.

Für die Nullstellenbestimmung ist folglich eine Bewertung und Auswahl der Näherungsverfahren vorzunehmen. Hierfür werden die folgenden – für die Aktion „Bestimmung einer Nullstelle“ spezifischen – Bewertungskriterien an die Näherungsverfahren angelegt.

- *Das Näherungsverfahren soll eine möglichst hohe Konvergenzgeschwindigkeit aufweisen, d. h. Ziel ist, die Nullstelle möglichst schnell zu bestimmen.*
- *Des Weiteren soll das Näherungsverfahren eine möglichst hohe Konvergenzwahrscheinlichkeit bieten. Dies bedeutet, dass das Näherungsverfahren die Nullstelle mit einer möglichst hohen Wahrscheinlichkeit findet.*

Daraus ergeben sich für die Nullstellenbestimmung die beiden Rangfolgen [Enge96, 26-42]:

Tabelle 1 Bewertung der Näherungsverfahren für die Nullstellenbestimmung

Konvergenzgeschwindigkeit (vom Schnellsten zum Langsamsten)	Konvergenzwahrscheinlichkeit (vom Sichersten zum Unsichersten)
Newton-Verfahren	Intervallschachtelung
Pegasus-Verfahren	Pegasus-Verfahren
Sekantenverfahren und Regula Falsi	Sekantenverfahren und Regula Falsi
Intervallschachtelung	Newton-Verfahren

Wie aus der Tabelle ersichtlich ist, lässt sich folglich kein dominantes Näherungsverfahren identifizieren. Allerdings liegen zwei dominierte Näherungsverfahren vor: Die Regula Falsi und das Sekantenverfahren. Da die Beurteilung des Pegasus-Verfahrens hinsichtlich beider Bewertungskriterien besser ausfällt, werden diese nicht im Service eingesetzt. Somit finden für die Bestimmung der Nullstelle das Newton-Verfahren, die Intervallschachtelung und das Pegasus-Verfahren Berücksichtigung.

Allerdings ist an dieser Stelle anzumerken, dass sich die Vorgehensweisen hinsichtlich der verschiedenen Kriterien u. U. nicht immer so eindeutig bewerten lassen, wie im Beispiel zur Nullstellenbestimmung. Dies trifft insbesondere dann zu, wenn die Erfüllung eines Kriteriums nicht quantitativ messbar ist.

Schritt 4.3: Anpassung der Aktivitätsdiagramme

Nachdem die für die Durchführung einer Aktion geeigneten Vorgehensweisen identifiziert wurden, werden die bestehenden Aktivitätsdiagramme angepasst. In den Fällen 4.1'' und 4.2b' liegt nur eine Vorgehensweise vor. Folglich ersetzt diese die bisherige Aktion. Für die Fälle 4.2a' und 4.2b'' werden alle (nicht dominierten) Vorgehensweisen im Service genutzt. Auch in diesem Fall erfolgt eine Aufnahme der Vorgehensweisen anstelle der bisherigen Aktion in die Aktivitätsdiagramme.

Für die Nullstellenbestimmung bedeutet dies, dass die Aktionen „Durchführung des Newton-Verfahrens“, „Durchführung des Pegasus-Verfahrens“ und „Durchführung der Intervallschachtelung“ die Aktion „Durchführung des Näherungsverfahrens“ ersetzen.

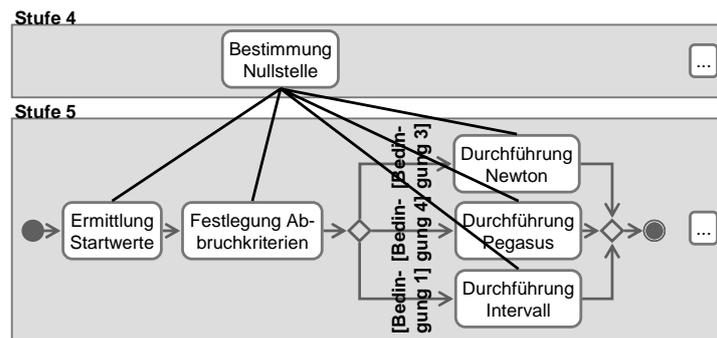


Bild 4 Ausschnitt aus dem um die Vorgehensweisen erweiterten Aktivitätsdiagramm

Neben dem Ersetzen der bisherigen Aktionen durch die Vorgehensweisen sind in den Fällen 4.2a' und 4.2b'', in denen mehrere Vorgehensweisen zur Auswahl stehen, die Bedingungen (s. Bild 4) für die Durchführung der Vorgehensweisen festzulegen, d. h. es ist eine entsprechende Auswahllogik zu definieren. Dies wird im Folgenden betrachtet.

Da für den Fall 4.2a' in Abhängigkeit der vorliegenden Voraussetzungen immer nur eine Vorgehensweise für die Durchführung in Betracht kommt, ist eine eindeutige Auswahllogik möglich. Im Fall 4.2b'' liegen dagegen sich nicht ausschließende Voraussetzungen vor und folglich können auch nach deren Überprüfung noch mehrere Vorgehensweisen eingesetzt werden. Aus diesem Grund ist zusätzlich eine Rangfolge für die Durchführung der Vorgehensweisen zu bilden. Hierfür lassen sich die in Schritt 4.2b aufgestellten Bewertungskriterien verwenden. Da diese i. A. jedoch konkurrierend sind, ist zunächst eine Priorisierung der Bewertungskriterien vorzunehmen. Stehen dann mehrere Vorgehensweisen für die Durchführung der Aktion zur Auswahl, wird die Vorgehensweise durchgeführt, die dem am höchsten priorisierten Bewertungskriterium am besten entspricht.

Für die Nullstellenbestimmung liegen die in Schritt 4.2b ausgewählten Näherungsverfahren vor. Dabei wird für die Nullstellenbestimmung das Bewertungskriterium „Konvergenzgeschwindigkeit“ wichtiger als „Konvergenzwahrscheinlichkeit“ eingestuft. Grund hierfür ist, dass der Service so gestaltet sein soll, dass bei dessen Durchführung die Antwortzeit so kurz wie möglich ist. Die Nullstellenbestimmung beginnt folglich immer mit dem schnellstmöglichen Näherungsverfahren, für welches die Voraussetzungen zur Durchführung gegeben sind. Um aber zu gewährleisten, dass der Service eine genauso hohe Konvergenzwahrscheinlichkeit aufweist, wie das bezüglich dieses Kriteriums am besten bewertete Verfahren, wird, wenn das schnellere Näherungsverfahren divergiert, d. h. die Nullstelle nicht findet, zum nächst langsameren, dafür aber sicheren Näherungsverfahren gewechselt.

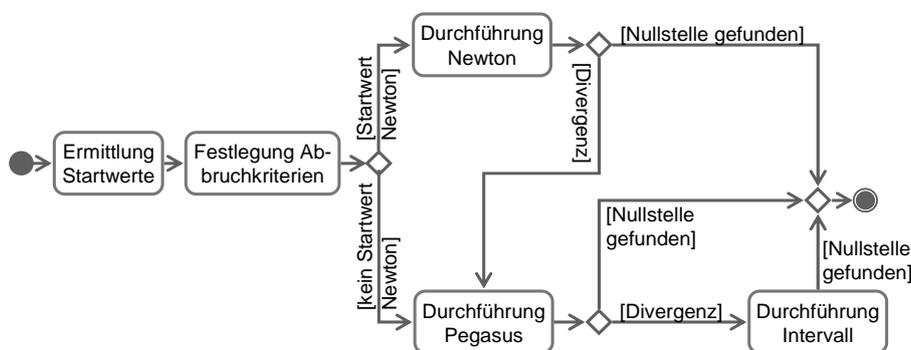


Bild 5 Auswahllogik für die Näherungsverfahren im Service „Nullstellenbestimmung“

Schritt 4.4: Überprüfung der Serviceidentifikation

Wurden die Vorgehensweisen und deren Auswahllogik ins Aktivitätsdiagramm aufgenommen, wird die vorgenommene Serviceidentifikation überprüft. Da für die Fälle 4.1' und 4.2b' (eine Vorgehensweise) lediglich die bisherige Aktion durch eine neue Aktion ersetzt wurde, ist hier keine Überprüfung erforderlich. In den Fällen 4.2a' und 4.2b'' (mehrere Vorgehensweisen) ist dagegen zu eruieren, ob die verschiedenen Aktionen, welche die bisherige Aktion ersetzt haben, in einem Service gekapselt oder jeweils als eigene Services angeboten werden sollten. Hierfür ist der in Kapitel 3 beschriebene Schritt 3.3 durchzuführen, d. h. die neuen Aktionen sind auf Basis der Anforderungen (A2) und (A3) hinsichtlich ihrer Eignung als Services zu untersuchen.

Dabei gilt auch hier, dass die Aktionen, wenn sie immer nur gemeinsam Verwendung finden, in einem Service zusammenzufassen sind. Dies trifft auf die Fälle 4.2a' und 4.2b'' zu. Es ist zwar möglich, dass im Aktivitätsdiagramm eine Vorgehensweise an einer Stelle alleine und an einer anderen Stelle mit anderen Vorgehensweisen kombiniert dargestellt ist. Da aber durch die Auswahllogik sichergestellt ist, dass in Abhängigkeit der vorliegenden Voraussetzungen immer die geeignete Vorgehensweise durchgeführt wird, kann der Service, der jede Vorgehensweise enthält, auch für die Aktion Verwendung finden, an der eigentlich nur eine Vorgehensweise benötigt wird. Die Kapselung der Vorgehensweisen samt deren Auswahllogik in einem Service entspricht des Weiteren der Anforderung (A1): Wird innerhalb des Service ausgewählt, welche Vorgehensweise durchzuführen ist, kann dieser übergreifend über die verschiedenen Ausprägungen einer Problemstellung eingesetzt werden. Würden dagegen die Vorgehensweisen jeweils als eigene Services angeboten, müsste immer zuerst entschieden werden, welcher Service geeignet ist. Ein weiterer mit der Kapselung der Auswahllogik einhergehender Vorteil ist damit die Vermeidung der Inkonsistenz bei der Auswahl der jeweiligen Vorgehensweise.

Zusammenfassend lässt sich damit sagen, dass i. A. keine erneute Serviceidentifikation erforderlich ist, da entweder nur eine Aktion durch eine andere ausgetauscht wurde oder zwar neue Aktionen hinzugekommen sind, diese aber immer gemeinsam genutzt werden können.

Da für die Nullstellenbestimmung die hinzugefügten Aktionen nur gemeinsam Verwendung finden, werden diese, gemeinsam mit der „Ermittlung der Startwerte“ und der „Festlegung der Abbruchkriterien“, im Service „Bestimmung einer Nullstelle“ gekapselt.

In diesem Kapitel wurde dargestellt, wie bei der fachlichen Gestaltung eines Service vorgegangen werden kann, wenn für die Durchführung einer Aktion eine Vorgehensweise oder mehrere zur Verfügung stehen und welche Auswirkungen sich dadurch auf die identifizierten Services ergeben. Bild 6 fasst noch einmal die einzelnen Schritte (markiert durch Kreise) und Fälle (markiert durch Kästchen) bei der fachlichen Gestaltung von Services zusammen.

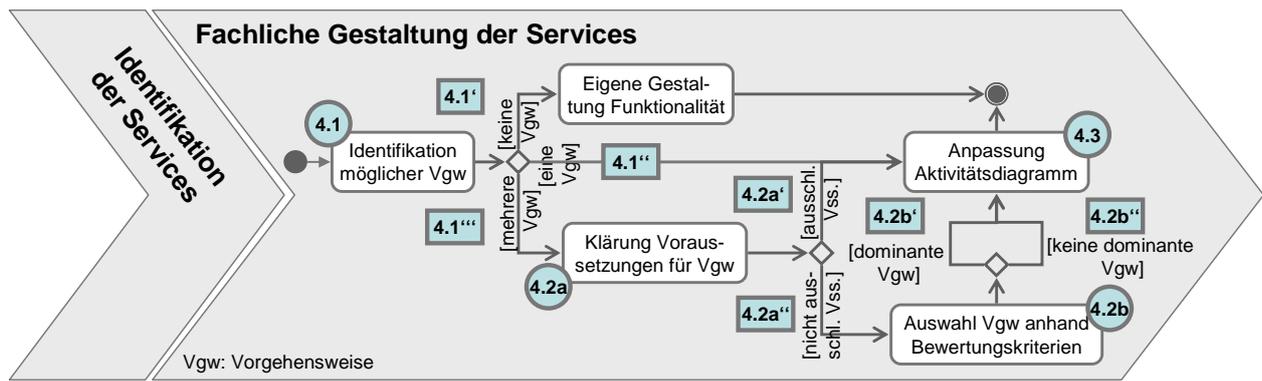


Bild 6 Vorgehen für die fachliche Gestaltung eines Service

5 Softwaretechnische Umsetzung von Services

Nachdem in Kapitel 4 auf die fachliche Gestaltung der vorher identifizierten Services eingegangen wurde, beschreibt dieses Kapitel, wie bei deren softwaretechnischer Umsetzung vorgegangen werden kann. Dabei ist es nicht Zielsetzung, den erforderlichen Quellcode detailliert zu beschreiben. Vielmehr fokussiert auch dieses Kapitel die Berücksichtigung der in Kapitel 2 genannten Anforderungen. Zunächst erfolgt eine allgemeine Beschreibung, wie sich das abstrakte Konzept SOA mittels Web Services umsetzen lässt. Anschließend wird dargestellt, wie sich der entwickelte Service als Web Service abbilden lässt, sodass dieser den Anforderungen entspricht.

5.1 Umsetzung von SOA mittels Web Services

Im Rahmen einer SOA interagieren Serviceanbieter, -vermittler und -nutzer. Die Kommunikation ermöglichen dabei die folgenden, für eine SOA grundlegenden, Komponenten: Die Kommunikationsmöglichkeit, die Beschreibung des Service und der Verzeichnisdienst. SOA ist jedoch vielmehr ein abstraktes Konzept einer Softwarearchitektur. Um dieses umzusetzen kommen verschiedene Technologien in Frage. Am weitesten verbreitet ist die Web-Services-Technologie. Zusätzlich weist diese gegenüber anderen möglichen Technologien wie bspw. CORBA, DCOM oder RMI Vorteile bezüglich Plattformabhängigkeit und Interoperabilität auf. Da sowohl Plattformunabhängigkeit als auch Interoperabilität aufgrund der meist heterogenen Systemlandschaft eines Finanzdienstleiters eine wichtige Voraussetzung für die Nutzung von Services darstellen, werden deshalb im Folgenden Web Services betrachtet [DJMZ05, 37ff; EbFi80ff].

Für die Beschreibung der Schnittstelle des Web Service findet die Web Service Description Language (WSDL), für den Verzeichnisdienst das Universal Description, Discovery and Integration Protocol (UDDI) Verwendung. Für die Kommunikation stehen verschiedene Protokolle zur Auswahl, sie erfolgt aber üblicherweise über das Simple Object Access Protocol (SOAP). Nicht vorgegeben ist die Programmiersprache für die Implementierung des Service.

Dies liegt darin begründet, dass ein wichtiger Aspekt in einer SOA ist, dass der Servicenutzer auf den Service ausschließlich über eine vordefinierte Schnittstelle zugreift, die Implementierung des Service selbst für den Nutzer aber nicht sichtbar ist (Information Hiding). Dadurch wird gewährleistet, dass der Service unabhängig von der für seine Implementierung verwendeten Programmiersprache (wieder-)verwendbar ist. [DJMZ05, 27ff, EbFi03, 175ff]

5.2 Umsetzung der identifizierten Services als Web Services

Wie sich die identifizierten Services gemäß den Anforderungen (A1) und (A3) als Web Services umsetzen lassen und welchen Beitrag die Web-Services-Technologie hierzu leisten kann, ist Bestandteil dieses Abschnitts. Anforderung (A2) wurde bereits durch die geeignete Identifikation der Services entsprochen, indem redundant zu implementierende Aktionen als Service angeboten werden. Zur Veranschaulichung der Ausführungen findet sich ein Beispiel für die Service Description und den Service zur Nullstellenbestimmung im Anhang.

Anforderung (A1) fordert die übergreifende Einsetzbarkeit des Service über mehrere Ausprägungen einer Problemstellung. In Kapitel 4 wurde durch die geeignete fachliche Gestaltung des Service bereits eine wichtige Voraussetzung hierfür geschaffen. Zur Erfüllung dieser Anforderung bei der Umsetzung des Service in einem Web Service trägt insbesondere die strikte Trennung zwischen der Schnittstelle und dem Service bei: So implementiert der Service selbst ausschließlich von der Problemstellung unabhängige Bestandteile. Die problemspezifischen Daten übermittelt der Servicenutzer an den Service über die Schnittstelle. Einen weiteren Beitrag zur problemübergreifenden Nutzung von Services in der heterogenen Systemlandschaft eines Finanzdienstleisters, aber auch zur schnellen und unkomplizierten Nutzung von Services (Anforderung (A3)), leistet die Verwendung der maschinenverarbeitbaren Sprache WSDL für die Beschreibung der Schnittstelle. Dadurch lässt sich der Service unabhängig von der im Service und in der Anwendung verwendeten Programmiersprache einsetzen [DJMZ05, 78ff].

Für den Service „Bestimmung einer Nullstelle“ wird folglich die entwickelte Funktionalität für dessen Durchführung im Service implementiert, da diese übergreifend über mehrere Ausprägungen einer Problemstellung einsetzbar ist. Um die Nullstelle bestimmen zu können, benötigt der Service jedoch von der betrachteten finanzwirtschaftlichen Gleichung abhängige Parameter wie Funktionswerte etc. Welche dies sind, spezifiziert die Service Description.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurde dargestellt, wie bei der Entwicklung eines Service in einer SOA vorgegangen werden kann. Dabei lag der Schwerpunkt der Arbeit nicht auf der konkreten technischen Umsetzung einer SOA und der Diskussion möglicher Technologien. Vielmehr wurde

analysiert, welche Anforderungen an einen Service in einer SOA bestehen und wie diesen bei der Identifikation, der fachlichen Gestaltung und der softwaretechnischen Umsetzung von Services entsprochen werden kann. Dies ist – wie in Kapitel 2 erwähnt – bisher nur selten Gegenstand von Veröffentlichungen.

Abschließend soll die Frage beantwortet werden, ob sich durch die beschriebenen Schritte zur Identifikation und fachlichen Gestaltung ein Service ergibt, welcher die aufgestellten Anforderungen erfüllt. Anforderung (A1) findet sowohl durch die geeignete fachliche Gestaltung als auch softwaretechnische Umsetzung Berücksichtigung: Für die Fälle 4.2a' und 4.2b'' werden alle Vorgehensweisen gemeinsam mit der Auswahllogik in den Service integriert, sodass dieser übergreifend über mehrere Problemstellungen in einem Unternehmen genutzt werden kann. Bei der Umsetzung als Web Service erfolgt eine strikte Trennung zwischen der Implementierung der problemspezifischen Aspekte und dem Service selbst. Des Weiteren erfolgt die Implementierung der Schnittstelle unter der Verwendung von WSDL, sodass eine Nutzung des Service in den heterogenen Anwendungen eines Finanzdienstleisters möglich ist. Anforderung (A2) erfüllt die geeignete Identifikation von Services: Findet eine Aktion in mehreren unterschiedlichen Aktionen Verwendung und wäre dort redundant zu implementieren, wird diese als eigener Service angeboten. Der Anforderung, dass sich der Service möglichst schnell und unkompliziert in eine bestehende Anwendung einbinden lässt (Anforderung (A3)), wird durch die geeignete Identifikation (Zusammenfassung von nur gemeinsam genutzten Aktionen zu einem Service) und fachliche Gestaltung der Services (Kapselung der Vorgehensweise und der Auswahllogik im Service) und durch die Verwendung von WSDL bei deren softwaretechnischer Umsetzung nachgekommen.

Dabei ist an dieser Stelle anzumerken, dass der Fokus der Arbeit auf der Identifikation und Gestaltung von Services liegt. Bei der praktischen Anwendung der beschriebenen Schritte ist zu berücksichtigen, dass sich aufgrund der Komplexität der in einem Unternehmen vorliegenden Geschäftsprozesse in der Realität meist sehr unübersichtliche Aktivitätsdiagramme ergeben, wodurch sowohl die beschriebenen Schritte 3.2 (Aufbereitung der Aktivitätsdiagramme) als auch 3.3 (Identifikation der Services) erschwert werden. Aus diesem Grund sollten diese i. A. nicht manuell durchgeführt, sondern durch entsprechende Programme unterstützt werden [WaXZ05, 229].

Dabei wurde im Beitrag, wie in der Einleitung erwähnt, davon ausgegangen, dass die Services ausschließlich innerhalb eines Unternehmens bereitgestellt und verwendet werden (Intra Enterprise Services) und keine kritischen Kundendaten verarbeiten. Themen wie rechtliche Rahmenbedingungen, die Qualität des Service hinsichtlich Parametern wie Verfügbarkeit, Leistungsfähigkeit, Fehlerhäufigkeit, Sicherheit oder Nutzungskosten bzw. die Garantie einer bestimmten Servicequalität hinsichtlich dieser Parameter mittels Service Level Agreements

(SLA), das dazugehörige Monitoring, Accounting und entsprechende Sanktionen bei Nichterfüllung des SLA, Verhandlungsmechanismen zwischen den verschiedenen Beteiligten, Lizenzmanagement oder Mandantenfähigkeit standen deshalb nicht im Vordergrund. Insbesondere für externe Anbieter von Services haben diese aber höchste Priorität: Nur dann werden die von diesen bereitgestellten Services in Unternehmen auch tatsächlich eingesetzt bzw. von den Serviceanbietern überhaupt zur Verfügung gestellt. Die Erweiterung des bisherigen Fokus Intra Enterprise Services auf die Betrachtung auch von Inter Enterprise Services und damit verbunden die Untersuchung der o. g. Aspekte stellt deshalb einen zentralen Gegenstand der weiteren Forschung dar.

Anhang

In Kapitel 5 wurde dargestellt, wie sich ein Service softwaretechnisch umsetzen lässt, um den Anforderungen (A1) und (A3) zu entsprechen. Im Folgenden wird am Beispiel der Nullstellenbestimmung veranschaulicht, wie sowohl der Service als auch die Service Description implementiert wurden. Dabei ist es nicht die Zielsetzung, alle Klassen des Service samt ihren Eigenschaften und Verhalten detailliert zu beschreiben, sondern es soll analog zu Kapitel 5.2 vielmehr aufgezeigt werden, inwiefern sich bei der Implementierung die Anforderungen (A1) und (A3) umsetzen lassen.

Anforderung (A3) wird dabei – wie in Kapitel 5.2 erwähnt – zum einen durch geeignete Identifikation und Gestaltung von Services, zum anderen durch die Verwendung der maschinenlesbaren Beschreibungssprache WSDL für die Implementierung der Schnittstelle erfüllt.

Anforderung (A1), welche die übergreifende Einsetzbarkeit des Softwaremoduls über mehrere Ausprägungen einer Problemstellung fordert, findet wie folgt Berücksichtigung:

Zum einen durch eine strikte Trennung der Implementierung der Näherungsverfahren und der Implementierung der der jeweiligen Problemstellung zugrunde liegenden Gleichung im Quellcode: Der Service führt lediglich die Näherungsverfahren durch. Losgelöst davon ist der Quellcode, der die anwendungsfallsspezifische finanzwirtschaftliche Gleichung abbildet. Zum anderen wird die Gleichung vom Service als „Blackbox“ betrachtet: Sie liefert dem Service über die Schnittstelle die nötigen Werte, dieser nähert die Nullstelle immer weiter an und beendet das Verfahren, sobald die zugrunde liegende Gleichung ein Ergebnis hinreichend nahe Null zurückgibt.

Im Folgenden werden sowohl der Service selbst als auch die Service Description kurz beschrieben.

Service

Wie in Abschnitt 5.1 erwähnt, ist für die Implementierung des Service selbst keine bestimmte Programmiersprache vorgegeben. Für den vorliegenden Service zur Nullstellenbestimmung wurde Java gewählt.

Die wichtigste Klasse im Service ist die Klasse „MarginalValueCalculator“. Diese bildet die in Bild 5 dargestellte Kombination der Näherungsverfahren zur Bestimmung der Nullstelle ab. In ihr sind Methoden implementiert, welche die Berechnungen auslösen, die Startwerte bestimmen und die Näherungsverfahren durchführen. Um die einzelnen Methoden zur Bestimmung der Startwerte und zur Durchführung der Näherungsverfahren wie in Bild 5 festgelegt aufeinander abfolgen zu lassen, wird die Methode „calculateMarginalValue()“ verwendet, die den Ablauf steuert.

Service Description

Wie oben angesprochen soll die der jeweiligen Problemstellung zugrunde liegende Gleichung strikt vom Service getrennt sein. Sie liefert verschiedene für die Durchführung der Vorgehensweise nötige – aber anwendungsfallsspezifische – Werte. Dazu gehören bspw. das Ergebnis der Gleichung bei Einsetzen des berechneten Näherungswertes und die Genauigkeit, mit der das Ergebnis zu bestimmen ist. Um dem o. g. „Black-Box“-Prinzip zu entsprechen, erfolgt die Kommunikation zwischen dem Service und der anwendungsfallsspezifischen Gleichung über eine standardisierte Schnittstelle. Die Service Description gibt dabei vor, welche Werte von der zugrunde liegenden Gleichung an den Service übergeben werden und welche Werte der Service an diese zurückgibt. Die Service Description spielt damit eine wesentliche Rolle für die Erfüllung der Anforderung (A1). Sie beinhaltet keinerlei für eine bestimmte Gleichung spezifischen Aspekte, sondern dient lediglich als Vorgabe für Werte, welche die für die jeweilige Problemstellung spezifischen Gleichungen liefern müssen.

Die Service Description enthält die folgenden beiden „operations“:

- `setInitialValues`: Diese operation setzt die Ausgangswerte für die Nullstellenbestimmung und startet den Service. Wird die Nullstelle gefunden, gibt die operation den Näherungswert zurück.
- `calculateFunctionValue`: Diese operation übergibt einen vom Service ermittelten Näherungswert an die zugrunde liegende Gleichung und gibt den dazugehörigen Funktionswert an den Service zurück.

```
<definitions targetNamespace="http://service/" name="ZeroPointService">
  <types/>
  <message name="setInitialValuesInput">
    <part name="startingValue" type="xsd:decimal"/>
    <part name="minimalValue" type="xsd:decimal"/>
    <part name="maximalValue" type="xsd:decimal"/>
    <part name="epsilon" type="xsd:decimal"/>
    <part name="deltaFirstDerivation" type="xsd:decimal"/>
    <part name="stepHeightDivisor" type="xsd:decimal"/>
  </message>
  <message name="setInitialValuesOutput">
    <part name="marginalValue" type="xsd:decimal"/>
  </message>
  <message name="calculateFunctionValueInput">
    <part name="value" type="xsd:decimal"/>
  </message>
  <message name="calculateFunctionValueOutput">
    <part name="functionValue" type="xsd:decimal"/>
  </message>
</definitions>
```

```

<portType name="ZeroPoint">
  <operation name="setInitialValues" parameterOrder="startingValue
minimalValue maximalValue epsilon deltaFirstDerivation stepHeightDivisor">
    <input message="tns:setInitialValuesInput"/>
    <output message="tns:setInitialValuesOutput"/>
  </operation>
  <operation name="calculateFunctionValue" parameterOrder="value">
    <output message="tns:calculateFunctionValueOutput"/>
    <input message="tns:calculateFunctionValueInput"/>
  </operation>
</portType>
<binding name="ZeroPointPortBinding" type="tns:ZeroPoint">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="setInitialValues">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal" namespace="http://service/" />
    </input>
    <output>
      <soap:body use="literal" namespace="http://service/" />
    </output>
  </operation>
  <operation name="calculateFunctionValue">
    <soap:operation soapAction=""/>
    <output>
      <soap:body use="literal" namespace="http://service/" />
    </output>
    <input>
      <soap:body use="literal" namespace="http://service/" />
    </input>
  </operation>
</binding>
<service name="ZeroPointService">
  <port name="ZeroPointPort" binding="tns:ZeroPointPortBinding">
    <soap:address location="http://localhost:8080/services"/>
  </port>
</service>
</definitions>

```

Literaturverzeichnis

[ACKM03]

Alonso, G.; Casati, F.; Kuno, H.; Machiraju, V.: Web Services. Concepts, Architectures and Applications. Springer Verlag, Berlin 2003.

[Aier06]

Aier, S.: How Clustering Enterprise Architectures helps to Design Service Oriented Architectures. In: Proceedings of the IEEE International Conference on Services Computing (SCC'06). IEEE Computer Society, Los Alamitos 2006, S. 269–272.

[AlDi06]

Albani, A.; Dietz, J.: The Benefit of Enterprise Ontology in Identifying Business Components. In: *Bramer, M. (Hrsg.):* IFIP World Computer Congress (WCC 2006). Springer Verlag, New York 2006, S.1-12.

[BCOZ04]

Brandner, M.; Craes, M.; Oellermann, F.; Zimmermann, O.: Web services-oriented architecture in production in the finance industry. In: Informatik Spektrum 27 (2004) 2, S. 136-145.

[BDHN77]

Becker, J.; Dreyer, H.-J.; Haacke, W.; Nabert, R.: Nullstellen. In: Numerische Mathematik für Ingenieure. Teubner Verlag, Stuttgart 1977, S. 24-76.

[BeMü99]

Becker, J.; zur Mühlen, M.: Rocks, Stones and Sand – Zur Granularität von Komponenten in Workflowmanagementsystemen. In: IM Die Fachzeitschrift für Information Management & Consulting (1999) 2, S. 57 – 67.

[BHMS06]

Berbner, R.; Heckmann, O.; Mauthe, A.; Steinmetz, R.: Eine Dienstgüte unterstützende Webservice-Architektur für flexible Geschäftsprozesse. In: Wirtschaftsinformatik 47 (2005) 4, S. 268-277.

[DJMZ05]

Dostal, W.; Jeckle, M.; Melzer, I.; Zengler, B.: Service-orientierte Architekturen mit Web Services. Spektrum Akademischer Verlag, München 2005.

[EbFi03]

Eberhardt, A.; Fischer, S.: Web Services. Carl Hanser Verlag, München 2003.

[Enge96]

Engeln-Müllges, G.: Numerik-Algorithmen: Entscheidungshilfen zur Auswahl und Nutzung. VDI Verlag, Düsseldorf 1996, 8. Auflage.

[GoKG03]

Gouscos, D.; Kalikakis, M.; Georgiadis, P.: An Approach to Modeling Web Services QoS and Provision Price. In: Proceedings of the 4th International Conference on Web Information Systems Engineering Workshops (WISE 2003), Rom, Italien 2003, S. 121-130.

[HuSi05]

Huhns, M.N.; Singh, M.P.: Service-Oriented Computing: Key Concepts and Principles. In: IEEE Internet Computing 46 (2005) 1, S. 75-81.

[JCIR01]

Jain, H.; Chalimeda, N.; Ivaturi, N.; Reddy, B.: Business component identification: A formal approach. In: Proceedings of the 5th IEEE International Enterprise Distributed Object Computing Conference. IEEE Computer Society Press, Seattle 2001, S. 183–187.

[KeLu03]

Keller, A.; Ludwig, H.: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Webservices. In: Journal of Network and Systems Management 11 (2003) 1, S. 57-81.

[MSJL06]

McGovern, J.; Sims, O.; Jain, A; Little, M.: Enterprise Service Oriented Architectures - Concepts, Challenges, Recommendations. Springer Verlag, Berlin 2006.

[Papa03]

Papazoglou, M.P.: Service-oriented Computing: Concepts, Characteristics and Directions. In: Proceedings of the 4th International Conference on Web Information Systems Engineering (WISE 2003), Rom, Italien 2003, S. 3-12.

[Parn72]

Parnas, D.L.: On the Criteria To Be Used in Decomposing Systems Into Modules. In: Communications of the ACM 15 (1972) 12, S. 1053-1058.

[Poci06]

Poccianti, P.: Internal Grid Computing Experiences in Financial Companies. In: Proceeding of Science: 1st International Workshop on Grid Technology for Financial Modeling and Simulation.

http://pos.sissa.it/archive/conferences/026/014/GRID2006_014.pdf, 2006, Abruf am 2006-11-09.

[RiHS05]

Richter, J.-P.; Haller, H.; Schrey, P.: Serviceorientierte Architektur. In Informatik Spektrum 28 (2005) 5, S. 413-416.

[ScWe06]

Schmidtman, V.; Weber, U.: SOA mit Betonung auf A: Architektur integriert Business und IT nachhaltig. In Information Management & Consulting 21 (2006) 2, S. 71-77.

[SzGM02]

Szyperski, C.; Gruntz, D.; Murer, S.: Component Software: Beyond Object-Oriented Programming. Addison-Wesley, New York 2002.

[ToSt05]

Tomann, M.; Steck, W.: Erfolgreicher Einsatz von EAI-Produkten und Servicebasierten Architekturen im Retail Banking. In: *Ferstl, O. K.; Sinz, E. J.; Eckert, S.; Isselhorst, T. (Hrsg.):* Wirtschaftsinformatik 2005 – eEconomy – eGovernment - eSociety. Physica-Verlag, Heidelberg 2005, S. 509-526.

[WaXZ05]

Wang, Z.; Xu, X.; Zhan, D.: A Survey of Business Component Identification Methods and Related Techniques. In: International Journal of Information Technology 2 (2005) 4, S. 229-238.

[XuYL03]

Xu, W.; Yin, B.L.; Li, Z.Y.: Research on the business component design of enterprise information system. In: Journal of Software 14 (2003) 7, S. 1213-1220.

[ZaZP03]

Zhang, Z.M.; Zhuang, Y.T.; Pan, Y.H.: Object-Oriented Software Reverse Engineering. In: Journal of Computer Research and Development 40 (2003) 7, S.1062-1068.