Softwarekartographie: Visualisierung von Anwendungslandschaften und ihrer Schnittstellen

Florian Matthes, André Wittenburg

Software Engineering betrieblicher Informationssysteme – sebis
Lehrstuhl für Informatik 19 – Institut für Informatik
Technische Universität München
Boltzmannstraße 3, 85748 Garching
{florian.matthes, andre.wittenburg}@in.tum.de

Abstract: Die verschiedenen betrieblichen Informationssysteme eines Unternehmens arbeiten nicht autark sondern sind in einem komplexen Netz – der Anwendungslandschaft – miteinander verwoben. Die einzelnen Beziehungen unterscheiden sich hinsichtlich ihrer technischen Realisierung und den fachlichen Aufgaben, so dass eine adäquate Beschreibung beide Dimensionen berücksichtigen muss.

In unserem Forschungsprojekt Softwarekartographie haben wir den Ist-Zustand hinsichtlich der Beschreibungstechniken für Anwendungslandschaften in Zusammenarbeit mit namhaften deutschen Unternehmen erfasst und die Unternehmen nach ihren Anforderungen und relevanten Aspekten für Softwarekarten befragt. Dabei wurden erhebliche Defizite bei der adäquaten Beschreibung der Vernetzung der Anwendungslandschaften und ihrer Schnittstellen deutlich.

Aufbauend auf dieser Ist-Analyse stellen wir ein Modell zur Beschreibung und Visualisierung von Schnittstellen in Anwendungslandschaften vor. Die technische Sicht dieses Modells beinhaltet die Kommunikationsart (synchron vs. asynchron), die verschiedenen Technologien und die unterstützenden Systeme. Auf fachlicher Ebene sind die definierten Verträge der Schnittstellen relevant, welche Geschäftsobjekte an den Schnittstellen über Services zugreifbar sind, welche Art des Zugriff (lesend vs. schreibend) erlaubt und wie die Kommunikationssteuerung (Online vs. Offline vs. Manuell) erfolgt.

1 Softwarekartographie und Softwarekarten

Ziel des Forschungsprojektes Softwarekartographie ist es, einen grundlegenden Begriffsapparat zur Beschreibung von Anwendungslandschaften aufzubauen, ein Modell für Softwarekarten zu entwickeln und das Modell adäquat in graphische Repräsentationen umzusetzen. Diese Softwarekarten sollen Anwendungslandschaften und die relevanten Aspekte der betrieblichen Informationssysteme, aus denen sich die Anwendungslandschaft zusammensetzt, visualisieren.

Wir haben zunächst in Zusammenarbeit mit den IT-Strategie-Abteilungen namhafter deutscher Unternehmen (u.a. BMW Group, Deutsche Post UB Brief, T-Com) den Ist-Zustand zur Beschreibung von Anwendungslandschaften analysiert.

Ein Ergebnis des Forschungsprojektes ist, dass eine Analyse von Anwendungslandschaften eine Betrachtung auf unterschiedlichen Ebenen (siehe Abbildung 1) erfordert (vgl. [MW04]):

Abbildung 1 -
Statische Betrachtungsebenen

Auf der obersten Ebene muss die Analyse die unternehmerischen und strategischen Ziele eines Unternehmens berücksichtigen. Diese Zielsetzung eines Unternehmens wird auf der mittleren Ebene in den Geschäftsprozessen des Unternehmens abgebildet bzw. verändert existierende. Zusätzlich müssen gesetzliche Regelungen und Maßgaben betrachtet werden, die Zielsetzungen und damit auch die Geschäftsprozesse beeinflussen.

Neben diesen operativen Belangen ergeben sich weitere administrative und dispositive Aufgaben (z.B. Buchhaltung, Personalwesen, Controlling, Dokumentenmanagement etc.), die bei der Erreichung der operativen Ziele unterstützen und ebenso für die Analyse der Anwendungslandschaft von Relevanz sind.

Auf der untersten Ebene werden die Geschäftsprozesse/-objekte implementiert bzw. durch betriebliche Informationssysteme unterstützt, die durch unterschiedliche Technologien realisiert sind, verschiedene Softwarearchitekturen benutzen etc.

Neben dieser *statischen Analyse* auf den unterschiedlichen Betrachtungsebenen ist eine *dynamische Analyse*, die die Evolution der Anwendungslandschaft berücksichtigt, von äquivalenter Bedeutung. Die Änderung von Zielen und Geschäftsprozessen kann eine Veränderung der Anwendungslandschaft zur Folge haben, da die Änderungen in existierenden oder neuen Informationssystemen abgebildet werden müssen. Ebenso kann das Auslaufen von Wartungsverträgen oder Produkten zu Veränderungen der Anwendungslandschaft führen.

Aus diesen drei Betrachtungsebenen und der dynamische Komponente haben sich im Rahmen der Anforderungsanalyse, bei der wir die Unternehmen nach ihren Anforderungen an Softwarekarten befragt haben, die folgenden Kategorien von relevanten Aspekten für Softwarekarten herauskristallisiert:

Die *planerischen Aspekte* erfassen die Veränderung der Anwendungslandschaft über die Zeit. Parallel laufende Programme und Projekte verändern die Anwendungslandschaft permanent und müssen aufeinander abgestimmt und priorisiert werden. Die Unterscheidung von Ist-, Soll- und Plan-Anwendungslandschaften führt in Kombination mit dem Lebenszyklen von Informationssystemen (in Planung, in Entwicklung, im Test, in Produktion, in Ablösung und abgelöst) zu einer zeitlichen Analyse der Anwendungslandschaft.

Die *wirtschaftlichen Aspekte* umfassen die verschiedenen Kostenarten, die bei der Entwicklung, dem Betrieb, der Wartung, dem Einkauf etc. von Informationssystemen ent-

standen sind, entstehen bzw. entstehen werden. Verschiedenen Kostenarten, IT-Kennzahlen [Kü03] und Balanced Scorecards [KN91] sollen visualisiert werden.

Fachliche Aspekte kombinieren u.a. Organisationseinheiten, Prozesse, Geschäftsobjekte und Funktionsbereiche mit den Informationssystemen. Auch die Anzahl von Nutzern oder der quantifizierbare Nutzen von Informationssystemen (z.B. mittels Function Points [IFP01]) zählen zu den fachlichen Aspekten.

Technische Aspekte erstrecken sich von der Implementierungssprache eines Informationssystems, über die Verbindungen bis hinzu Eigenschaften wie Architektur oder genutzter Middleware. Im Gegensatz zu Beschreibungssprachen wie UML [OMG03] fokussiert die Softwarekartographie auf die Zusammenhänge in der gesamten Anwendungslandschaft. Ziele wie Homogenisierung von Datenbanksystemen, *Enterprise Application Integration* [Ke02] oder Individual- vs. Standardsoftware werden betrachtet.

Operative Aspekte beziehen sich auf den unmittelbaren Betrieb von Informationssystemen und die verbundenen Ereignisse. Domino-Effekte bei Ausfällen oder der Ablauf von zeitgesteuerten Prozessen werden im Fokus der Anwendungslandschaft berücksichtigt.

Ein Modell, das die relevanten Aspekte abbildet, muss hierbei zwischen den relevanten, den *erfassbaren* und *pflegbaren* Aspekten unterscheiden, da Unternehmen zwar alle Informationen besitzen, jedoch die Pflege der Informationen, der mit Aufwand und Kosten verbunden ist, einen entsprechenden Nutzen und Mehrwert besitzen muss.

2 Schnittstellen und Konnektoren

In diesem Beitrag konzentrieren wir uns auf den technischen und fachlichen Aspekt der Schnittstellen für Softwarekarten, bei der unterschiedliche Geschäftsobjekte über verschiedenste Verbindungen zwischen Informationssystemen zugänglich gemacht werden.

Die Vernetzung der Anwendungslandschaft wird von den existierenden Softwarekarten unser Projektpartner unterschiedlich dargestellt und dokumentiert, wobei Detailinformationen teilweise in externen Datenspeichern (Tabellen etc.) gepflegt werden. Werden die Darstellungsformen generalisiert, ergeben sich die folgenden Stufen (siehe Abbildung 2), die zu dem Modell in Abbildung 3 führen:

Stufe 0 visualisiert die Informationssysteme in Zusammenhang mit Organisationseinheiten, Prozessen und Funktionsbereichen. Auf diese Stufe werden keine direkten Verbindungslinien zwischen den einzelnen Informationssystemen gezeichnet, sondern die Verbindungen ergeben sich indirekt durch die zweidimensionale Anordnung nach Funktionsbereichen, Prozessen etc.

Werden die Informationssysteme mittels Linien oder Pfeilen miteinander verbunden (*Stufe 1*), soll dies Schnittstellen dokumentieren, die einzelne Systeme miteinander verbinden. Die Pfeilenden der Linien stellen entweder die Datenflussrichtung oder die Rolle

des Client vs. des Servers dar und führt somit zu einer Überladung dieses Gestaltungsmittels.

In der betrieblichen Praxis wird der Begriff *Schnittstelle* zur Bezeichnung einer Verbindung zwischen zwei Informationssystemen verwendet, was nicht der im Software Engineering üblichen Terminologie entspricht. So definiert UML eine Schnittstelle als „A named set of operations that characterize the behavior of an element.“ [OMG03]; gleiches gilt für Definitionen bei CORBA [OMG04] oder Szyperski [Sz02]. Daher vermeiden wir in unserem Modell den überladenen Begriff der *Schnittstelle*.

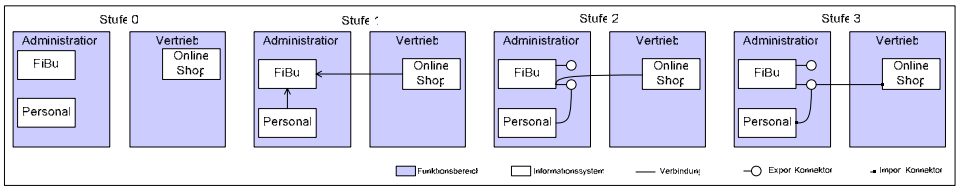

Abbildung 2 – Schnittstellenvisualisierung auf verschiedenen Detaillierungsstufen

Die existierenden Definitionen und Visualisierungen von Schnittstellen unterscheiden des Weiteren nicht zwischen einer technischen und einer fachlichen Sicht. Werden Schnittstellen im Kontext der Anwendungslandschaft analysiert, so treten die exakten Definitionen der einzelnen Methoden und deren Signaturen in den Hintergrund. Stattdessen sind die Konnektoren mit Kommunikationsart, Art des Zugriffs (lesend vs. schreibend) und die Technologie entscheidend. Wird von dem eigentlichen Vertrag, den ein *Export-Konnektor* schließt, abstrahiert, so sind der *Service* und die betroffenen *Geschäftsobjekte* für das Netz in der Anwendungslandschaft von größerer Bedeutung (siehe Abbildung 3).

Stufe 2 führt somit zu einer Betrachtung von Export-Konnektoren, die Services für andere Informationssysteme anbieten und einer Unterscheidung zwischen einer fachlichen und einer technischen Ebene. Ein Informationssystem kann hierbei den gleichen Service, der durch einen fachlichen Export-Konnektor angeboten wird, mittels verschiedener Technologien exportieren, für einen fachlichen Export-Konnektor können somit mehrere technische Export-Konnektoren existieren (in den Abbildungen nicht visualisiert).

Wird eine Verbindung mit Export-Konnektoren analysiert, so agiert das exportierende System als Server, welches Services gegenüber anderen Systemen (den Clients) anbietet. In einer *Stufe 3* werden auf Seite der Clients die Import-Konnektoren ergänzt. Diese Import-Konnektoren nutzen beispielsweise nur Teile der Export-Konnektoren oder greifen nur lesend auf einen Export-Konnektor zu, der sowohl lesenden als auch schreibenden Zugriff anbietet.

Die Visualisierung der Export-Konnektoren ist an das von UML [OMG03] und anderen Visualisierungssprachen bekannten Lollipop-Symbols angelehnt, die Import-Konnektoren werden durch ein ausgefülltes Rechteck auf der Seite des Clients dargestellt.

Die Berücksichtigung der Konnektoren und deren Visualisierung ermöglicht es beispielsweise Transaktionsraten, laufende Projekte (die Konnektoren verändern) oder Handlungsbedarf (Flaschenhälse etc.) darzustellen.

Das Klassendiagramm in Abbildung 3 ist ein von uns entwickelter Vorschlag zur Zusammenfassung der einzelnen Stufen und Konzepte, welcher im weiteren Projektverlauf auf seine Anwendbarkeit hin untersucht werden muss.

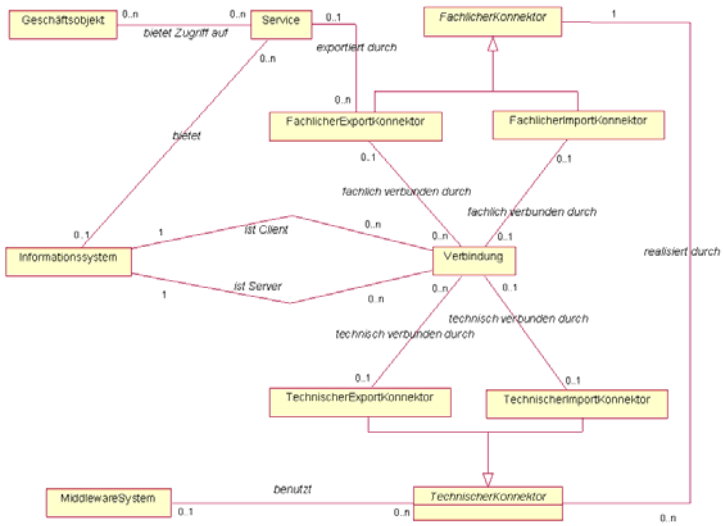

Abbildung 3 – Konzeptuelles Modell für Informationssysteme und Konnektoren

Anhand der Kardinalitäten an den Assoziationen ist zu erkennen, dass die Erfassung der Konnektoren optional ist, da eine nachträgliche Aufnahme der Informationen möglich sein soll. Die befragten Unternehmen verfügen nicht ad hoc über alle notwendigen Informationen und wollen auch Beziehungen ohne Konnektoren erfassen können.

3 Visualisierung von Konnektoren auf Softwarekarten

Das folgende Beispiel einer Softwarekarte hat als Kartengrund[1] die Anwendungsbereiche eines hypothetischen Finanzkonzerns, der stark vereinfacht ist und stellt gegenüber den existierenden Darstellungen, die bis zu 200 Systeme auf einer Karte visualisieren, nur 19 Systeme dar. Für weitere Beispiele von Softwarekarten wird auf [MW04] verwiesen.

[1] Ein Kartengrund ist innerhalb der Softwarekartographie der Untergrund einer Softwarekarte, auf den die Informationssysteme und ihre Aspekte aufgetragen werden.

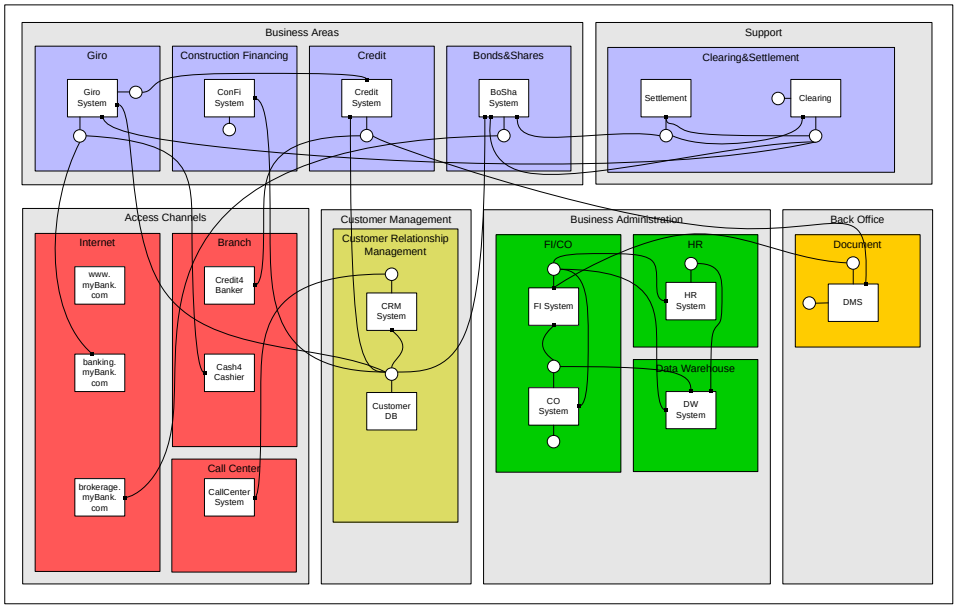

Abbildung 4 – Softwarekarte: Anwendungslandschaft mit Verbindungen und Konnektoren

Abbildung 4 zeigt eine Softwarekarte mit der Anwendungslandschaft des Beispiel-Konzerns und ihrer Schnittstellen, wobei die einzelnen Informationssysteme den Anwendungsbereichen zugeordnet werden. Visualisiert werden

- die Funktionsbereiche (Rechtecke mit Beschriftung und farbigem Hintergrund),
- die Informationssysteme (Rechtecke mit Beschriftung und weißem Hintergrund),
- die Verbindungen (Linien),
- die fachlichen Export-Konnektoren (Lollipop-Symbol) und
- die fachlichen Import-Konnektoren (ausgefülltes Quadrat).

Die Darstellung ist bedingt durch die zahlreichen Linien für Schnittstellen sehr unübersichtlich, zeigt jedoch wo Knotenpunkte – Systeme mit vielen Verbindungen – existieren. Die Einführung der fachlichen Export-Konnektoren als eigenes Symbol ermöglicht es hierbei zu erkennen, welche Systeme die gleichen Dienste auf der fachlichen Ebene importieren.

Die Abbildung 5 reduziert die Softwarekarte aus Abbildung 4 auf die Verbindungen des Informationssystems „Giro System“ und fügt die Zugriffsoptionen der Konnektoren hinzu. Auf dieser Softwarekarte ist es somit möglich, die Verbindungen eines einzelnen Systems genauer zu analysieren. Ein entsprechendes Werkzeug muss es ermöglichen, durch eine Auswahl eines einzelnen Elements – z.B. eines Konnektors – zusätzliche Informationen (Name, Attribute, Beschreibung etc.) zu erhalten.

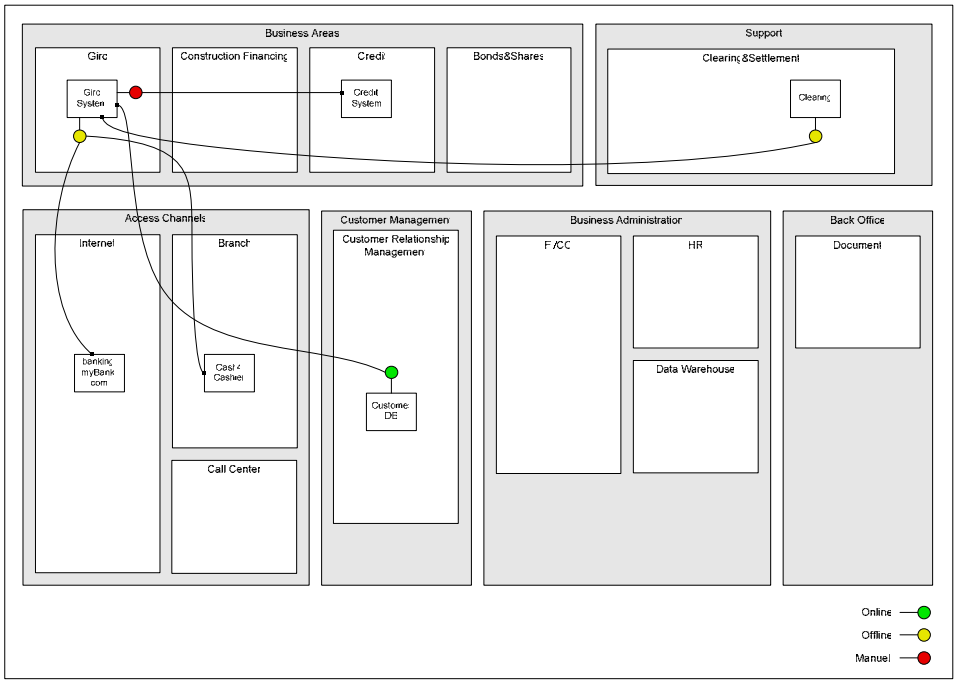

Abbildung 5 – Softwarekarte: Giro System mit Verbindungen und Konnektoren

Durch den Schichtenaufbau der Softwarekarte (Kartengrund mit Funktionsbereichen, die Informationssysteme, die Konnektoren und die Verbindungen) können einzelnen Aspekte je nach Anforderung ein- oder ausgeblendet werden. Das Nutzen von Farben ermöglicht es zusätzliche Informationen, wie beispielsweise die Unterscheidung der Kommunikationsart (Online vs. Offline vs. Manuell) auf fachlicher Ebene oder der Middleware-Systeme auf technischer Ebene, durch Einfärben der Export-Konnektoren darzustellen.

Softwarekarten, die verschiedene Sichten auf die Anwendungslandschaft ermöglichen, haben nur dann einen Mehrwert, wenn die Daten in einer Repository-gestützten Anwendung gepflegt werden und die Erstellung (semi-)automatisiert ist, um die gewünschten Darstellungen zu erhalten. Wir werden uns daher im weiteren Projektverlauf nicht nur auf die theoretischen Grundlagen beschränken sondern ebenso das Modell für Softwarekarten auf die anderen relevanten Aspekte ausweiten und das Modell mit unseren Projektpartnern abstimmen.

Die Entwicklung eines geeigneten Werkzeuges zur Softwarekartographie, welches als Repository existierende Werkzeuge der Prozessmodellierung und des IT-Management nutzen soll (ARIS Toolset, alfabet SITM, Casewise Corporate Modeler etc.), befindet sich in Planung.

Literaturverzeichnis

[IFP01] IFPUG: Function Point Counting Practices Manual. Version 4.1.1, International Function Point Users Group, 2001.

[Ke02] Keller, M.: Enterprise Application Integration. Erste Auflage, dpunkt.Verlag, Heidelberg, 2002. ISBN 3-89864-186-4.

[KN91] Kaplan, R.; Norton, D.: The Balanced Scorecard - Measures That Drive Performance. Harvard Business Review, Vol. 70, S. 71-79, 1991.

[Kü03] Kütz, M.: Kennzahlen in der IT : Werkzeuge für Controlling und Management. Erste Auflage, dpunkt.Verlag, Heidelberg, 2003. ISBN 3-89864-225-9.

[MW04] Matthes, F.; Wittenburg, A.: Softwarekarten zur Visualisierung von Anwendungslandschaften und ihren Aspekten. Technische Universität München, Fakultät für Informatik, Lehrstuhl für Informatik 19, Technischer Bericht, 2004. http://www.softwarekartographie.de

[OMG03] OMG: Unified Modeling Language Specification, Version 1.5. Object Management Group, 2003.

[OMG04] OMG: Common Object Request Broker Architecture: Core Specification - Version 3.0.3. Object Management Group, 2004.

[Sz02] Szyperski, C.: Component Software. Zweite Auflage, Addison-Wesley, London, 2002. ISBN 0-201-74572-0.