

Softwarekartographie: Visualisierung von Anwendungslandschaften und ihrer Schnittstellen

Florian Matthes, André Wittenburg

Software Engineering betrieblicher Informationssysteme
Lehrstuhl für Informatik 19 – Technische Universität München
Boltzmannstraße 3, 85748 Garching
{matthes, wittenbu}@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, das zwischen einer technischen und einer fachlichen Sicht unterscheidet.

1 Softwarekartographie und Softwarekarten

Ziel des Forschungsprojektes Softwarekartographie ist es, einen grundlegenden Begriffssapparat 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. AXA Service, 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 0) erfordert (vgl. [MW04]).

Auf der obersten Ebene muss die Analyse die unternehmerischen und strategischen Ziele eines Unternehmens berücksichtigen. Diese Zielsetzung wird auf der mittleren Ebene in den Geschäftsprozessen des Unternehmens abgebildet bzw. verändert existierende.

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 Modifikationen von Zielen und Geschäftsprozessen können eine Veränderung der Anwendungslandschaft zur Folge haben, da diese Modifikationen in existierenden oder neuen Informationssystemen abgebildet werden müssen. In der Softwarekartographie unterscheiden wir zwischen planerischen, wirtschaftlichen, fachlichen, technischen und operativen Aspekten (vgl. [MW04]).

Ein Modell, das die relevanten Aspekte abbildet, muss hierbei zwischen den relevanten, den *erfassbaren* und *pflgbaren* 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.

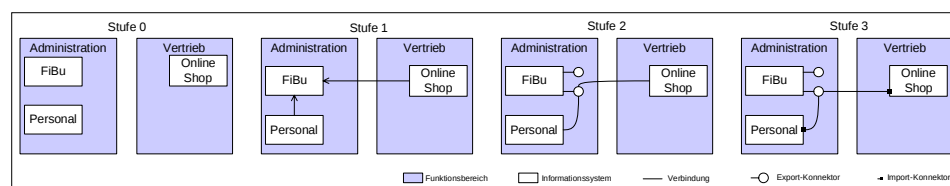


Abbildung 1 – Schnittstellenvisualisierung auf verschiedenen Detaillierungsstufen

Die Vernetzung der Anwendungslandschaft wird von den existierenden Softwarekarten unserer Projektpartner unterschiedlich dargestellt und dokumentiert. Werden die Darstellungsformen generalisiert, ergeben sich die folgenden Stufen (siehe Abbildung 1), die zu dem Modell in Abbildung 2 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 eine zweidimensionale Anordnung.

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*.

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 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 2).

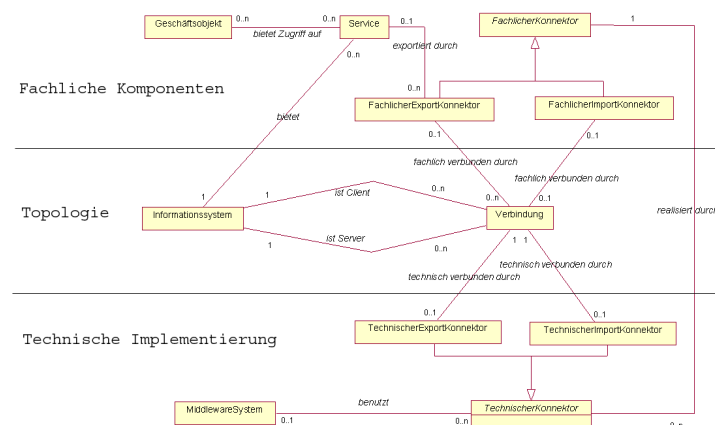


Abbildung 2 – Konzeptuelles Modell für Informationssysteme und Konnektoren

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.

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.

Das Klassendiagramm in Abbildung 2 ist ein von uns entwickelter Vorschlag zur Zusammenfassung der einzelnen Stufen und Konzepte, welcher im weiteren Projektverlauf auf seine Anwendbarkeit hin untersucht werden muss.

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.

4 Visualisierung von Konnektoren auf Softwarekarten

Das folgende Beispiel einer Softwarekarte hat als Kartengrund 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.

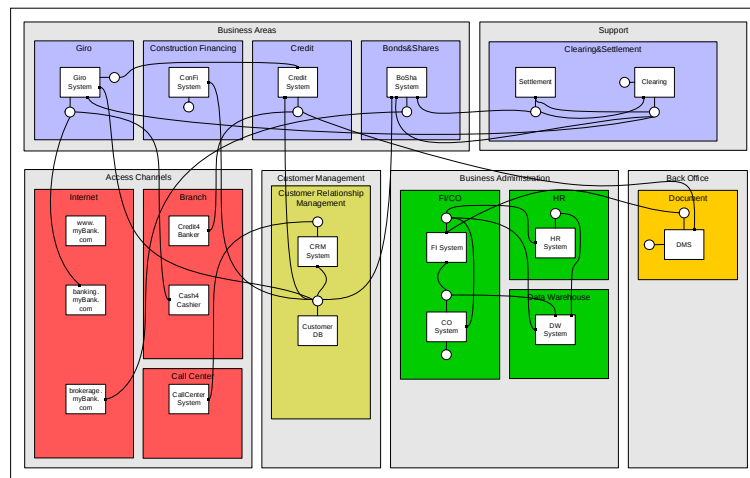


Abbildung 3 – Softwarekarte: Anwendungslandschaft mit Verbindungen und Konnektoren

Abbildung 3 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.

Abbildung 4 reduziert die Softwarekarte aus Abbildung 3 auf die Verbindungen nur eines Informationssystems und fügt die Zugriffsoptionen der Konnektoren durch farbliche Hervorhebung hinzu. Auf dieser Softwarekarte ist es somit möglich, die Verbindungen eines einzelnen Systems genauer zu analysieren.

Softwarekarten, die verschiedene Sichten auf die Anwendungslandschaft ermöglichen, besitzen nur dann einen Mehrwert, wenn die Daten in einer Repository-gestützten Anwendung gepflegt werden und die Erstellung (semi-)automatisiert ist. Wir werden daher im weiteren Projektverlauf das Modell für Softwarekarten auf die anderen relevanten Aspekte ausweiten und das Modell mit unseren Projektpartnern abstimmen.

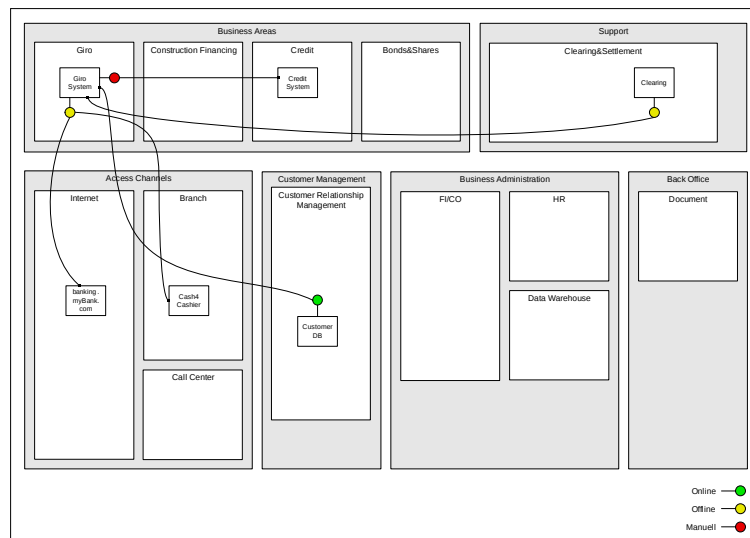


Abbildung 4 – Softwarekarte: Giro System mit Verbindungen und Konnektoren

Literaturverzeichnis

- [MW04] Matthes, F.; Wittenburg, A.: Softwarekarten zur Visualisierung von Anwendungslandschaften und ihren Aspekten. TU München, 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.