

Von der Softwarekartographie zur Corporate Intelligence

Dierk Jugel¹, Michael Falkenthal¹, Christian Schweda²,
Michael Pretz³, Alfred Zimmermann¹

¹ Hochschule Reutlingen, Architecture Reference Lab,
vorname.nachname@reutlingen-university.de

² iteratec GmbH, Christian.Schweda@iteratec.de

³ SOA Innovation Lab e.V., Michael.Pretz@daimler.com

Abstract: Graphische Modelle komplexer Systeme, sog. Views, werden in vielen Ingenieurs- und Managementdisziplinen als Mittel für die Dokumentation und zur Planung eben dieser Systeme verwendet. Mithilfe dieser Modelle, welche das System aus unterschiedlichen Blickwinkeln, sog. Viewpoints, betrachten, lassen sich die unterschiedlichen Interessenslagen (Concerns) der am System interessierten Rollen (Stakeholder) adäquat abbilden. Für das Management von Unternehmensarchitekturen bietet die Softwarekartographie ein Werkzeuginstrumentarium, mit dem sich unterschiedliche Sichten konsistent und automatisiert zu einem statischen Bericht über den Systemzustand zusammenführen lassen. Durch einen derartigen Bericht lassen sich jedoch nicht die veränderlichen Informationsbedarfe im Rahmen einer explorativen Analyse des Systems befriedigen. In dieser Publikation beschreiben wir einen Ansatz, der das Visualisierungsinstrumentarium der Softwarekartographie mit Instrumenten für die dynamische Aggregation von Daten zusammenführt, um am System interessierten Rollen interaktive Blickwinkel bieten zu können.

1 Motivation und Hinführung

Betrachtet man Unternehmen aus dem Blickwinkel des Enterprise Architecture Managements (EAM), bestehen Unternehmen heute mehr denn je aus komplexen Strukturen, die sich aus einer Vielzahl von Architekturartefakten, bspw. in Form von Applikationen oder Geschäftsprozessen, zusammensetzen. Die Architekturartefakte stehen dabei in vielfältigen Beziehungen zueinander. Dementsprechend stehen Ingenieurs- und Managementdisziplinen, die das Unternehmen in seiner Gesamtheit zum Gegenstand haben, vor der Herausforderung, relevante Eigenschaften und Charakteristika des Unternehmens für die Entscheider aufzubereiten. Diese Herausforderung wird noch dadurch verschärft, dass die Entscheider keine homogene Gruppe mit einer gemeinsamen Interessenslage bilden. Vielmehr sind verschiedene am System interessierte Rollen (*Stakeholder*) zu unterscheiden, die unterschiedliche Interessenslagen (*Concerns*) im Bezug auf das Unternehmen oder relevante Teile desselben haben. Um diese Interessenslagen zu befriedigen, werden unterschiedliche Blickwinkel (*Viewpoints*) auf das Unternehmen bezogen und entsprechend relevante Charakteristika in graphischen Modellen (*Views*) abgebildet. Die verschiedenen Blickwinkel zielen dabei nicht selten auf überschneidende Ausschnitte des Un-

ternehmens ab, d.h. ein Charakteristikum wird in verschiedenen Modellen repräsentiert. Dadurch entstehen Konsistenzerfordernisse zwischen den Modellen, bzw. den ihnen zu Grunde liegenden Blickwinkeln. Dementsprechend müssen die verwendeten Techniken zur Modellierung diesen Erfordernissen Rechnung tragen. Das Visualisierungsinstrumentarium der Softwarekartographie ([ELSW06, LMW05]) setzt die Konsistenzerfordernisse auf Basis einer einheitlichen Beschreibung des Unternehmens passend zu einem gesamthaften Metamodell¹ um. Dabei werden die Blickwinkel über dem Metamodell als Abbildungen definiert und konkrete graphische Modelle aus dieser Beschreibung generiert. Das Instrumentarium ermöglicht damit ein Berichtswesen über die Struktur des Unternehmens. Statische Berichte auf Basis vordefinierter Blickwinkel können genutzt werden um wiederkehrende und ex ante bekannte Interessenslagen der Interessenten adäquat abzudecken. Mit Hilfe ad-hoc definierter Perspektiven ist es daneben möglich, kurzfristig auftretende Interessen abzudecken ([BGS10]). Eine perspektivenübergreifende explorative Analyse der Charakteristika des Unternehmens jedoch ist mit Berichten auf Basis unveränderlicher Blickwinkel nicht möglich. So lassen sich *konfluente* Veränderungen an den Blickwinkeln oder übergreifendes Hervorheben mit dem Instrumentarium der Softwarekartographie nicht umsetzen. Im Folgenden wollen wir aufzeigen, wie sich das Instrumentarium der Softwarekartographie zu einem umfassenden Analyseinstrumentarium entwickeln lässt. Die Fähigkeit, Stakeholder umfassend bei der Entscheidungsfindung durch ein interaktives Analyseinstrumentarium zu unterstützen, nennen wir *Corporate Intelligence*. Mit diesem Instrumentarium der *Corporate Intelligence* lassen sich weitergehende Anforderungen realisieren. In Abschnitt 2 werden diese Anforderungen anhand eines Anwendungsszenarios abgeleitet. Abschnitt 3 stellt verwandte Ansätze vor. Abschnitt 4 zeigt die Grundprinzipien der Softwarekartographie auf, während Abschnitt 5 Erweiterungen zur Umsetzung der Corporate Intelligence beschreibt. Abschnitt 6 reflektiert die Ergebnisse der Arbeit kritisch und zeigt weitere Entwicklungspotenziale sowie künftige Forschungsthemen auf.

2 Anforderungen an die Corporate Intelligence

In diesem Abschnitt beschreiben wir ein Anwendungsszenario, in dem ein aus unserer Sicht optimales interaktives Visualisierungsinstrumentarium zur Analyse komplexer Sachverhalte innerhalb einer Unternehmensarchitektur aufgezeigt wird. Ausgehend von diesem Szenario leiten wir in Abschnitt 2.2 Anforderungen an die Corporate Intelligence ab.

2.1 Anwendungsszenario

Ziel der Analyse, die durch das Anwendungsszenario realisiert werden soll, ist die Aufdeckung von Konsolidierungspotential in Bezug auf Applikationen innerhalb einer Unternehmensarchitektur. Dazu bestellt ein Enterprise Architekt des Unternehmens eine Be-

¹ In [ELSW06] wird dieses Metamodell auch *Informationsmodell* genannt.

sprechung mit allen relevanten Stakeholdern ein. Jeder Stakeholder ist fachlich für eine oder mehrere Applikationen verantwortlich. Die Besprechung findet in einem als Enterprise Architektur Cockpit ausgebauten Besprechungsraum statt. Der Raum verfügt über eine Art Powerwall mit drei großen Präsentationsflächen nebeneinander. Da das Unternehmen mehrere Standorte besitzt und nicht alle Stakeholder vor Ort sein können, existiert an jedem relevanten Standort solch ein Cockpit. Alle im Folgenden beschriebenen Interaktionen innerhalb des Cockpits werden in Echtzeit an die anderen Standorte propagiert. Darüber hinaus können alle Blickwinkel des Cockpits einzeln an Arbeitsplatzrechnern, Notebooks oder sonstigen mobilen Geräten dargestellt werden.

Um Konsolidierungspotentiale aufdecken zu können, sind für die Stakeholder die vier im Folgenden beschriebenen Blickwinkel wesentlich. Dabei enthalten sämtliche Blickwinkel dieselben Applikationen, jedoch in verschiedenen Zusammenhängen.

- Die *Applikationsliste* enthält wesentliche, im Fokus stehende, Applikationen und ausgewählte Eigenschaften in einer tabellarischen Struktur.
- Der *Bebauungsplan* zeigt die Zuordnung von Applikationen zu Organisationseinheiten und Geschäftsprozessen in Form einer Matrix auf.
- Das *Portfoliodiagramm* ordnet Applikationen hinsichtlich zweier Dimensionen ein. In unserem Szenario sind dies exemplarisch die Dimensionen Komplexität und Kosten. Die Komplexität ist eine Kennzahl, die beschreibt, aus welchen Artefakten eine Applikation zusammengesetzt und wie aufwendig dementsprechend die Wartung ist.
- Die *Informationsflussgrafik* zeigt Informationsflüsse zwischen Applikationen auf.

Neben den beschriebenen Blickwinkeln spielt die zeitliche Dimension bei der Analyse von Unternehmensarchitekturen eine wichtige Rolle ([Han10] S. 11). Aus diesem Grund wird zwischen der Ist- und der Soll-Bebauung unterschieden ([Han10] S. 436ff). Dabei beschreibt die Ist-Bebauung den aktuellen Zustand der Bebauung, die Soll-Bebauung den Zielzustand, der in der Zukunft erreicht werden soll. Sowohl für die Ist- als auch für die Soll-Bebauung sind die vier oben beschriebenen Blickwinkel von Bedeutung.

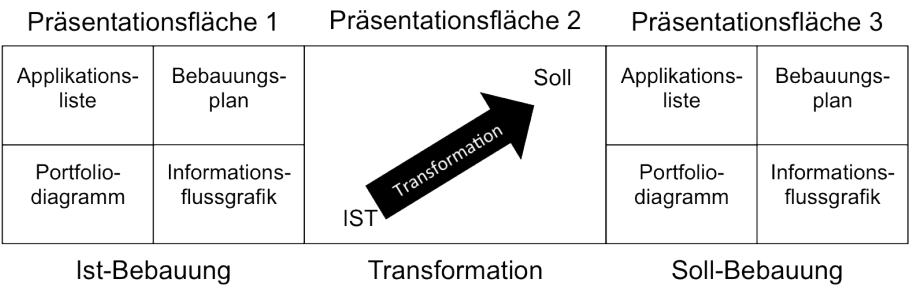


Abbildung 1: Das Enterprise Architektur Cockpit

Abbildung 1 zeigt das Enterprise Architektur Cockpit, das eine aus unserer Sicht optimale Analysemöglichkeit für die Aufdeckung des Konsolidierungspotentials bietet. Die linke große Präsentationsfläche des Cockpits wird in unserem Anwendungsszenario in vier Teilflächen unterteilt, die die vier Blickwinkel der Ist-Bebauung zeigen. Analog werden auf der rechten Präsentationsfläche die Blickwinkel der Soll-Bebauung dargestellt. Die mittlere Präsentationsfläche wird für die Darstellung der Transformation vom Ist zum Soll genutzt. Auch hierbei ist eine Aufteilung der Fläche zur Darstellung mehrerer Blickwinkel der Transformation möglich. Mit dieser Einteilung des Enterprise Architektur Cockpits stehen den Stakeholdern für die angestrebte Analyse alle notwendigen Informationen zur Verfügung.

Der erste Schritt der Analyse ist die Auswahl eines Stakeholders durch den leitenden Enterprise Architekten. Anschließend werden die dem ausgewählten Stakeholder im Fokus stehenden Applikationen durch die drei Präsentationsflächen angezeigt. Mit Hilfe des *semantischen Zooms* ist eine Änderung der Detaillierungsstufe der einzelnen Blickwinkel möglich, sodass zusätzlich die im erweiterten Fokus stehenden Applikationen dargestellt werden können. Im Gegensatz zu einem herkömmlichen Zoom, mit dessen Hilfe die Größe von Objekten innerhalb einer Darstellung skaliert werden kann, wird eine Darstellung mit Hilfe des semantischen Zooms grundlegend geändert. Hierbei können sich Typ und Bedeutung von Informationen durch eine Transition zwischen genereller und detaillierter Sichten auf Informationen verändern [Mod97]. Sind in diesem Beispiel auf der generellsten Detaillierungsebene die für einen Stakeholder im Fokus stehenden Applikationen dargestellt, können die im erweiterten Fokus stehenden Applikationen durch Erhöhung der Detaillierungsstufe hinzugefügt werden. Die Notwendigkeit der Unterscheidung zwischen Applikationen, die im engsten Fokus des Stakeholders sind, und denen, die im erweiterten Fokus stehenden, ist die Problematik des endlichen Platzes und der Übersichtlichkeit. Für einen Stakeholder kann eine Vielzahl von Applikationen relevant sein. Werden all diese Applikationen dargestellt, verliert der Stakeholder schnell den Überblick. Aus diesem Grund sieht der Stakeholder im engsten Fokus nur die Applikationen, die er als am wichtigsten für sich erachtet. Nichts desto trotz darf er die restlichen Applikationen nicht aus dem Hinterkopf verlieren. Dies ist beispielsweise bei der Verteilung von Budget wichtig. Ebenso ist durch den semantischen Zoom eine Navigation von den Applikationen bis hin zu den darunterliegenden Technologien und der Hardware ohne den Verlust des Gesamtzusammenhangs möglich. Eine Navigation in Richtung der Geschäftsarchitektur hin zu Geschäftsfunktionen und Domänen ist ebenfalls durchführbar.

Haben die Stakeholder eine Applikation als Kandidaten für die Konsolidierung identifiziert, wird die Applikation mit Hilfe der Möglichkeit der Zuordnung von Annotationen markiert. Es besteht die Möglichkeit, eine Annotation mit einer kurzen Beschreibung zu versehen. Durch die *MouseOver* Funktion kann der der Annotation zugeordnete Kommentar dargestellt werden. Diese Annotation ist als Hinzufügen von Metainformationen zu sehen. Hierbei ist die Festlegung des Teilnehmerkreises, für den die Annotation sichtbar ist, äußerst wichtig, sodass nicht jeder Anwender, der den Blickwinkel einsehen kann, auch die Annotation sehen kann.

Ein weiteres Feature für die Analyse ist die Funktionalität der *What-If Analyse*. Hierbei kann eine Applikation ausgewählt und beispielsweise als abgeschaltet annotiert werden.

Dies hat zur Folge, dass in allen Blickwinkeln die von der Abschaltung betroffenen Applikationen automatisch hervorgehoben bzw. annotiert werden. Auch hierbei kommt der semantische Zoom zum Einsatz, mit dessen Hilfe die darzustellenden Applikationen festgelegt werden können. Sind in der geringsten Detaillierungsstufe nur die Applikationen dargestellt, die einen direkten Informationsfluss von oder zur abgeschalteten Applikationen besitzen, können im Weiteren Applikationen dargestellt werden, die keinen direkten Informationsfluss zwischen sich und der Applikation besitzen, jedoch trotzdem indirekt über andere Applikationen davon betroffen sind. Mit Hilfe der *Zurück* Funktion kann jede Interaktion rückgängig und der vorherige Zustand wiederhergestellt werden.

2.2 Anforderungen

Matthes et al. formulieren in [MBLS08] verschiedene Anforderungen, die Werkzeuge für die Visualisierung und visuelle Modelle erfüllen müssen. Die Anforderungen sind dabei in Arbeitsgruppen mit den Industriepartnern der Werkzeugstudie erhoben worden. Aufbauend auf diese Anforderungen haben wir eine Verfeinerung vorgenommen und dabei das Feedback von Industriepartnern aus dem SOA Innovation Lab² einfließen lassen. Die erhobenen Anforderungen sind in der Folge aufgezählt.

A1: Beliebige Blickwinkel Es können beliebige Blickwinkel und graphische Darstellungsformen definiert werden. Dabei besteht die Möglichkeit, verschiedene Teilarchitekturen innerhalb einer Unternehmensarchitektur einzubeziehen. Beispielsweise kann ein Sachverhalt dargestellt werden, der Teile der Business, Applikations- und Technologiearchitektur (vgl. TOGAF [The12]) beinhaltet.

A2: Rollen und Rechte Die einem Stakeholder angebotenen Blickwinkel und deren Bestandteile werden entsprechend seiner Rollen und Rechte ausgewählt. Dabei beziehen sich die Interessenslagen eines Stakeholders auf dessen Rollen. Ein Applikationsspezialist interessiert sich beispielsweise für Applikationen innerhalb eines Unternehmensbereichs und deren Schnittstellen, wohingegen die darunterliegende Hardware nicht in seinem Interessensgebiet liegt. Zudem lassen sich Einschränkungen der Sichtbarkeit von Elementen einer Unternehmensarchitektur entsprechend der Berechtigungen ableiten. Ein Stakeholder kann nur die Elemente sehen, für die er berechtigt ist.

A3: Stakeholder vordefinierte Blickwinkel Zusätzlich zu Anforderung 2 ist es dem Stakeholder möglich, innerhalb des durch ihn einsehbaren Bereichs spezielle Blickwinkel zu konfigurieren. Der Stakeholder kann aus einer Reihe von vordefinierten Darstellungstypen wählen. So kann er z.B. auf eine Portfoliodarstellung oder eine graphartige Darstellung zurückgreifen. Die Darstellungstypen orientieren sich dabei an den Best-Practice Darstellungsformen aus dem EAM Patternkatalog der TU München ([BELM08]).

²Das SOA Innovation Lab (www.soa-lab.de) ist eine Initiative von Großunternehmen und Behörden im deutschsprachigen Raum zu SOA und Architekturmanagement.

A4: Interaktionsmöglichkeiten Der Stakeholder kann auf verschiedene Arten mit den Darstellungen interagieren ohne die dabei zugrunde liegenden Daten über die Unternehmensarchitektur zu ändern:

A4a Graphisches Annotieren Der Stakeholder kann einzelne Elemente in einem View auswählen und hervorheben. Zusätzlich zur Hervorhebung ist ein kurzer Kommentar möglich.

A4b Graphisches Filtern Der Stakeholder kann die Grundgesamtheit aller Elemente, die angezeigt werden, auf ausgewählte Elemente beschränken, d.h. filtern.

A4c Semantisches Zoomen Der Stakeholder kann für ausgewählte Elemente den Detaillierungsgrad erhöhen, d.h. hierarchisch untergeordnete Teilelemente aus- und einblenden.

A4d MouseOver Hierbei werden dem Anwender zusätzliche Informationen zu einem Element dargestellt, wenn dieser mit dem Maus Cursor über das jeweilige Element fährt oder es anklickt.

A4e Rückgängig machen Bei allen Interaktionen ist es wichtig, den Zustand vor der jeweiligen Interaktion zu speichern, sodass der vorherige Zustand jederzeit wiederhergestellt werden kann.

A5: Verbundene Perspektiven Für eine gesamthafte Analyse der Architektur werden mehrere Blickwinkel gleichzeitig angewendet und dargestellt. Die Interaktionsformen werden dementsprechend, wo möglich, von einem konkreten View auf alle verbundenen Views propagiert. So z.B. wird die Hervorhebung eines Elements auf allen Views simultan ausgeführt.

A6: Kontrollierte Kollaboration Kontrollierte Kollaborationen ermöglichen eine standortunabhängige Zusammenarbeit mehrerer Architekten. Hierbei wird davon ausgegangen, dass alle an einer Kollaboration beteiligten Architekten dieselbe Sicht betrachten. Führt ein Teilnehmer eine Interaktion durch, wird die dadurch auslösende Änderung der Sicht an alle Teilnehmer propagiert und umgesetzt. Hierbei besteht die Möglichkeit der Transparenzeinschränkung dahingehend, dass die Sichtbarkeit von Interaktionen, z.B. das Annotieren von abzulösenden Applikationen, durch Rollen und Rechte eingeschränkt werden kann.

A7: What-If Analysen Zur Unterstützung von Entscheidungsträgern können What-If Analysen durchgeführt werden. Eine What-If Analyse ist eine Form der Simulationsunterstützung und zeigt Konsequenzen einer Veränderung von Elementen innerhalb der Unternehmensarchitektur auf. Ein Anwender kann dadurch beispielsweise die Auswirkungen der Abschaltung einer Applikation oder den Ausfall eines Servers analysieren. Den Entscheidungsträgern werden dadurch sämtliche Beziehungen zu betroffenen Elementen graphisch dargestellt. Hierbei ist zu unterscheiden, wie weitreichend diese Analyse durchgeführt werden soll. Für eine umfassende Analyse sind die Beziehungen erster Ordnung zu einem Element meist nicht ausreichend. Daher kann der Anwender den Grad der Ordnung frei wählen und sich mit Hilfe des semantischen Zooms durch die einzelnen Beziehungen navigieren.

A8: Einfaches Bedienungskonzept Für die Benutzbarkeit des Analyse- und Visualisierungsinstrumentariums ist ein einfaches Bedienungskonzept von essentieller Bedeutung. Viele der vorgestellten Anforderungen existieren auf die ein oder andere Weise bereits in verschiedenen EAM Werkzeugen, jedoch ist der Aufwand der Konfiguration und Zusammenstellung solcher Darstellungen sehr hoch und setzt ein weitreichendes Expertenwissen voraus, dass bei vielen Anwendern nicht gegeben ist. Dies hat zur Folge, dass viele Funktionalitäten, die per sé von einem Werkzeug angeboten werden, entweder durch Unkenntnis der Existenz oder der Umsetzung nicht genutzt werden.

3 Verwandte Ansätze

[BGP11] beschreiben einen Ansatz zur Adaption des Kontrollraumkonzepts für den Anwendungsbereich von Unternehmensarchitekturen. Das Kontrollraumkonzept wird beispielsweise bei der Überwachung von Kraftwerken oder des Schienenverkehrs eingesetzt. Ein Kontrollraum besteht typischerweise aus einer Vielzahl von Monitoren, die dem Verantwortlichen einen Überblick über die aktuelle Situation geben. Die Idee dieses Ansatzes ist die Schaffung einer Übersicht der aktuellen Situation einer Unternehmensarchitektur mit dem Fokus auf Geschäftsprozesse und Applikationen. Dies soll durch eine kombinierte Darstellung von Überwachungsinformationen von Geschäftsprozessinstanzen und der zugeordneten Applikationen erfolgen. Die Autoren stellen eine Architektur für das sogenannte “Business Software Control Center” vor, ein Visualisierungskonzept fehlt jedoch.

[SMR12] beschäftigen sich mit der Frage, wie die Generierung interaktiver Visualisierungen von Unternehmenslandschaften realisiert und ein zugehöriges Referenzmodell oder -framework aussehen könnte. Die Untersuchungen der Autoren ergaben, dass ein solches nicht existiert. Aus diesem Grund stellen die Autoren ein konzeptuelles Framework vor, welches auf den von [BEL⁺07b] vorgestellten Prinzipien der Softwarekartographie basiert. Im Weiteren definieren die Autoren Anforderungen an solch ein Framework und stellen einen Prototypen vor. Aus der Publikation geht jedoch nicht im Detail hervor, welche Arten der Interaktion ermöglicht werden sollen. Die definierten Anforderungen überschneiden sich teilweise mit den unseren, da bspw. auch dort die Möglichkeit der Impact Analyse angesprochen wird. Der beschriebene Ansatz beruht auf Interaktionen innerhalb eines Blickwinkels. Der Aufbau eines Architektur Cockpits mit mehreren zusammenhängenden Views ist nicht Bestandteil dieses Ansatzes.

4 Grundprinzipien der Softwarekartographie

Die Softwarekartographie zielt darauf ab, Konsistenz zwischen verschiedenen Views auf die Unternehmensarchitektur mit Hilfe eines gesamthaften Modells derselben sicherzustellen. Dazu nimmt sie in ihrem Instrumentarium eine strenge Trennung zwischen den darzustellenden Informationen und deren Darstellung vor. Ergänzend zu dieser Trennung jedoch

werden Beziehungen zwischen den Informationsobjekten und den Symbolen aufrechterhalten. All dies wird durch Nutzung von Modell-zu-Modell-Transformationen ermöglicht, mit Hilfe derer die Viewpoints hinter den Views spezifiziert werden. Die zentralen Konzepte des Ansatzes der Softwarekartographie sind nach [BEL⁺07a]:

- Das *Informationsmodell* beschreibt welche Typen von Objekten in den Daten über die Unternehmensarchitektur verwendet werden. Übliche Typen im Informationsmodell sind INFORMATIONSSYSTEM oder GESCHÄFTSPROZESS.
- Das *semantische Modell* enthält die konkreten Objekte, welche die Unternehmensarchitektur beschreiben. Beispielhafte Objekte in einem semantischen Modell wären `SAP3.1:IS` oder `Einkauf:GP`.
- Das *Visualisierungsmodell* definiert Typen von Symbolen und von Visualisierungsregeln, mit denen ein View beschrieben werden kann. Übliche Typen im Visualisierungsmodell sind RECHTECK oder VERSCHACHTELUNG.
- Das *symbolische Modell* enthält die konkreten Symbole und Gestaltungsregeln, welche einen konkreten View beschreiben. Beispielhafter Inhalt in einem symbolischen Modell wären `SAP3.1:Rechteck` oder `Einkauf:Polygon`.

Als Verbindung zwischen den oben genannten Konzepten wirkt eine Modell-zu-Modell-Transformation, die den anzuwendenden Viewpoint beschreibt. Diese Transformation übersetzt Typen im Informationsmodell in Symboltypen aus dem Visualisierungsmodell, analog werden Eigenschaften der Informationsobjekte zu graphischen Eigenschaften der Symbole, und Beziehungen zwischen den Informationsobjekten zu Gestaltungsregeln transformiert. Folgende Transformationsregel, notiert in einer an den MOF Query View Transformation (QVT) Standard angelehnten Syntax, illustriert die Funktionsweise der Transformationen in einem Viewpoint:

```
transform IS2Rectangle {  
  from infoObject: InformationSystem  
    to symbol: Rectangle {  
      label = infoObject.name  
    }  
}
```

Durch die Modell-zu-Modell-Transformation wird dabei auch eine Korrespondenz zwischen den Objekten des Modells und deren symbolischen Repräsentationen hergestellt. Diese Korrespondenz wiederum kann als Grundlage für die möglichen Interaktionen der Corporate Intelligence dienen. Abbildung 2 stellt die Grundkonzeption der Softwarekartographie mit den relevanten Modellen und Metamodellen dar. Bereits in dieser Grundkonzeption sind die Anforderungen **A1** und **A3** erfüllt. Die Erfüllung von Anforderung **A2** lässt sich in der Konzeption der Softwarekartographie durch eine geeignete Zugriffssteuerung auf das semantische Modell umsetzen, durch die nur der aktuellen Rolle zugängliche Objekte der entsprechenden Typen zur Verfügung gestellt werden. Auch wird durch die

Modell-zu-Modell-Transformation die Grundlage für Anforderung **A5** gelegt, da das semantische Modell eine gemeinsame Basis für die relevanten Views bildet, welche dadurch wiederum miteinander verbunden sind.

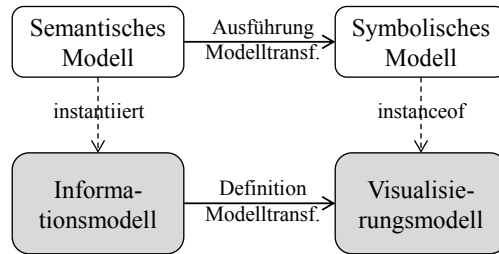


Abbildung 2: Metamodelle und Modelle der Softwarekartographie (Eigene Darstellung nach [BEL⁺07a])

Bei der Definition der Modell-zu-Modell-Transformation im Viewpoint können verschiedene Beschreibungssprachen Anwendung finden. In der Softwarekartographie wurden verschiedene derartige Beschreibungssprachen erfolgreich verwendet [Ach10]. Dies waren die objekt-orientierte Programmiersprache Java, die deklarative Atlas Transformation Language (ATL) für die Beschreibung allgemeiner Modell-zu-Modell-Transformationen. Später wurden domänenspezifische Sprachen, genannt “Viewpoint Definition Language” (VDL) verwendet, mit denen die Abstraktionslücke bei der Verwendung zu allgemeiner Sprachen angegangen wurde. Als Endpunkt der Entwicklung stellt Schweda in [Sch11] eine VDL vor, die wiederverwendbare Bausteine für die Definition von Viewpoints, sog. *Viewpoint Building Blocks* (VBB) einführt. Diese Viewpoint Building Blocks werden zu Viewpoints zusammengesetzt und mit Konzepten aus dem Informationsmodell (oder darauf aufbauend) parametrisiert. Durch die Anhebung des Abstraktionsniveaus wird, wie in [BGS10] beschrieben, die Definition neuer Viewpoints durch Unternehmensarchitekten erleichtert. Bei den VBBs werden zwei Arten von Bausteinen unterschieden:

- *Symbolerzeugende VBBs* transformieren ein einzelnes Informationsobjekt in das korrespondierende Symbol. Die zugrundeliegende Transformation wird dabei zwischen Typen des Informationsmodells und Symboltypen des Visualisierungsmodells beschrieben und bildet *identifizierbar* ab. Dementsprechend kann für ein Symbol immer das dahinter liegende Informationsobjekt ermittelt werden.

- *Strukturerzeugende VBBs* verbinden mehrere symbolerzeugende VBBs und determinieren das Layout des Views über absolute sowie relative Positionierung über Gestaltungsregeln. Die Gestaltungsregeln werden dabei aus Beziehungen zwischen den Informationsobjekten abgeleitet. Die Definition dieser Ableitung wiederum erfolgt auf Ebene der Typen, d.h. der konkrete strukturerzeugende VBB wird durch einen Beziehungstyp oder Attributtyp parametrisiert, der auf einen festgelegten Gestaltungsregeltyp abgebildet wird.

Ein Beispiel aus [Sch11] illustriert das Zusammenspiel symbol- und strukturerzeugender VBBs. Ein symbolerzeugender VBB ist für die Transformation von INFORMATIONSSYSTEMEN in Rechtecke verantwortlich, der umgebende strukturerzeugende VBB vom Typ *Graph* fügt diese Rechtecke zu einem Graphen zusammen. Dabei werden die SCHNITTSTELLEN zwischen den Informationssystemen als Kanten im Graphen repräsentiert. Die Verknüpfungen der VBBs sind in Abbildung 3 dargestellt.

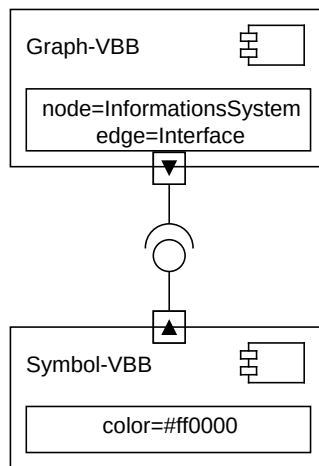


Abbildung 3: Verknüpfung von VBBs (Eigene Darstellung nach [Sch11])

Zentral für die Konzeption der VDL ist die Unterscheidung von zwei Teiltransformationen, welche die Transformation in einem Viewpoint bilden:

- In der ersten Teiltransformation wird beschrieben, wie die Informationen aus dem semantischen Modell aggregiert und gefiltert werden. Dadurch entsteht virtuell³ ein *aggregiertes semantisches Modell*.
- In der zweiten Teiltransformation wird beschrieben, wie die Objekte aus dem aggregierten semantischen Modell in Symbole und Gestaltungsregeln übersetzt werden.

³Dieses virtuelle semantische Modell lässt sich konzeptionell fassen, existiert aber bei der Verarbeitung der Informationen nicht zwingend komplett.

Durch die Einführung eines *View-Metamodells* lässt sich die Grundkonzeption des Ansatzes der VDL-basierten Erzeugung von Views ähnlich fassen, wie der Ansatz eingangs vorgestellte Ansatz der Softwarekartographie. Das *View-Metamodell* stellt dabei die Typen, Eigenschaften und Beziehungen dar, welche von den Objekten des aggregierten semantischen Modells instantiiert werden. Abbildung 4 stellt das Zusammenspiel der verschiedenen Modelle und Metamodelle dar.

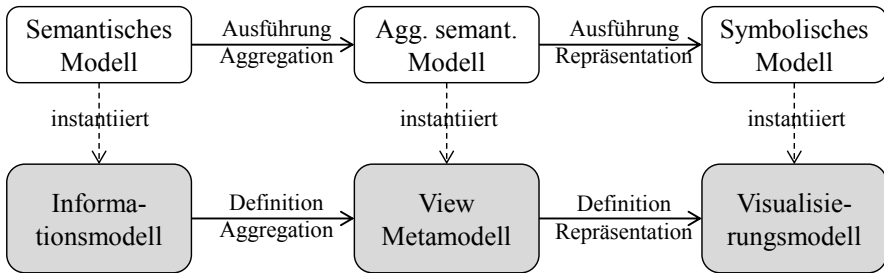


Abbildung 4: Metamodelle und Modelle VDL-basierter Viewpoints (Eigene Darstellung nach [Sch11])

5 Der Weg zur Corporate Intelligence

In diesem Abschnitt beschreiben wir die notwendigen Erweiterungen zur Realisierung des Konzepts der *Corporate Intelligence*. Insbesondere mit der Unterscheidung zwischen den beiden Teiltransformationen *Aggregation* und *Repräsentation* bildet die VDL-basierte Viewpointdefinition einen Startpunkt für die Umsetzung der Anforderungen der *Corporate Intelligence*. Hierbei sind zwei Arten von Interaktionen zu unterscheiden:

- zum Einen Hervorhebungen, bei denen zusätzliche Informationen in der Analyse verfügbar und diese konsistent in mehreren Views auf Basis verschiedener Viewpoints angezeigt werden sollen
- zum Anderen Änderungen der Aggregation, bei denen die Kriterien für Aggregation und Filterung der im semantischen Modell befindlichen Objekte angepasst und in mehreren Views auf Basis verschiedener Viewpoints angezeigt werden sollen.

Die entsprechenden Interaktionsmechanismen der *Corporate Intelligence* wirken dabei auf unterschiedliche Bestandteile der transformationsbasierten Viewgenerierung. Dabei wird mit dem *annotierten semantischen Modell* eine gemeinsame Informationsquelle zur Beschreibung der Annotationen geschaffen. Interaktionen, die auf Hervorhebungen, d.h. auf die Annotation mit zusätzlichen Informationen, abzielen, richten ihre Wirkung auf dieses annotierte semantische Modell. Interaktionen, die Änderungen an den Aggregationen

vornehmen, wirken auf die erste Teiltransformation ein. Um diese, in Abbildung 5 illustrierte, Grundkonzeption der *Corporate Intelligence* zu ermöglichen, müssen verschiedene Anpassungen und Erweiterungen an der bausteinbasierten Transformation vorgenommen werden. Diese betreffen das neue annotierte semantische Modell, bzw. das zugehörige Metamodell, siehe Abschnitt 5.1, die symbolerzeugenden VBBs, siehe Abschnitt 5.2, und die strukturerzeugenden VBBs, siehe Abschnitt 5.3.

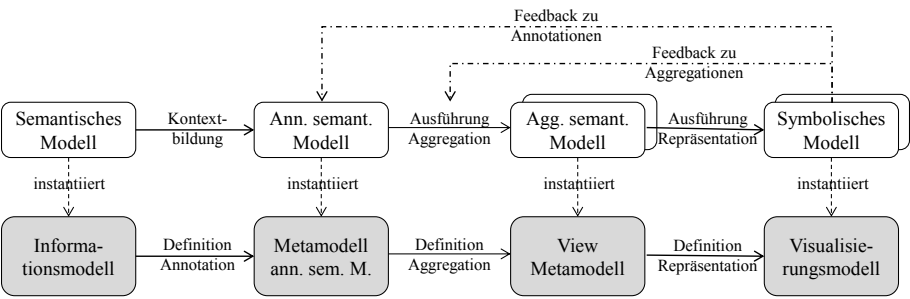


Abbildung 5: Metamodelle und Modelle annotierbarer VDL-Viewpoints

5.1 Annotiertes Informationsmodell

Ergänzend zu klassischen Konzepten zur Beschreibung von semantischen Modellen, stellt der Mechanismus des annotierten semantischen Modells eine weitergehende Mächtigkeit zur Verfügung. Diese Mächtigkeit zentriert sich um das Konzept der ANNOTATION, die auf Ebene des Metamodells für das semantische Modell zur Verfügung gestellt wird. Die Annotation erlaubt es beliebige Informationsobjekte ad-hoc auszuzeichnen. Hinausgehend über den konkreten Kontext der aktuellen Bearbeitung hat diese Annotation keine statisch oder übergreifend festgelegte Semantik. Abbildung 7 zeigt das Metamodell für das annotierte semantische Modell, Abbildung 7 eine konkrete Anwendung der Annotation im Kontext der Analyse einer Unternehmensarchitektur.

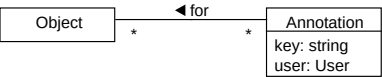


Abbildung 6: Metamodell des annotierten semantischen Modells

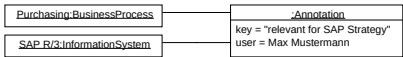


Abbildung 7: Annotiertes semantisches Modells

Die Annotation dient als Grundlage für die Hervorhebung von Symbolen in konkreten Views. Dabei wird die Annotation des Informationsobjekts in eine graphische Hervorhebung eines aus dem Objekt transformierten Symbols verwendet. Die graphische Hervorhebung orientiert sich dabei an der konkreten Parametrisierung des symbolerzeugenden VBBs, vgl. Abschnitt 5.2. Mithilfe des annotierten semantischen Modells, bzw. der Annotation wird die Grundlage für die Erfüllung der Anforderungen **A4d**, **A5** und **A7** aus Abschnitt 2 gelegt.

5.2 Symbolerzeugende VBBs

Die symbolerzeugenden VBBs werden im Hinblick auf das Annotationskonzept des annotierten semantischen Modells angepasst. Dies ist zum Einen ein Mechanismus, um die Annotationen bei der Transformation, d.h. bei der Erzeugung der konkreten Symbole, zu berücksichtigen. Abhängig von der Parametrisierung des VBBs werden verschiedene graphische Hervorhebungen zur Anwendung gebracht. Mögliche Hervorhebungen greifen auf ungenutzte Visualisierungsvariablen, z.B. Füllfarbe oder Randfarbe zurück. Eindeutigkeit bei der Hervorhebung lässt sich durch den eindeutigen Rückbezug von konkreten Symbolen zu Informationsobjekten gewährleisten. Zum anderen stellen die symbolerzeugenden VBBs die Callback-Funktionen zur Verfügung, mit Hilfe derer ein konkretes Informationsobjekt annotiert und die Annotation auch wieder entfernt werden kann. Damit lassen sich Anforderungen **A4a** und **A6** erfüllen. Auch werden im Zusammenspiel mit dem annotierten semantischen Modell die Anforderungen **A4d**, **A5** und **A7** erfüllt.

5.3 Strukturerzeugende VBBs

Die strukturerzeugenden VBBs werden im Hinblick auf das graphische Filtern und das semantische Zoomen erweitert. Dazu wird pro strukturerzeugendem VBB ein graphischer Mechanismus für das semantische Zoomen fest definiert. Dies ist am Beispiel einer graphartigen Darstellung bestehend aus Knoten und Kanten der Übergang zu einem Hypergraphen, dessen Knoten weitere Kindknoten enthalten. Für das graphische Filtern stellen die strukturerzeugenden VBBs Mechanismen zur Verfügung, die bestehenden Annotationen aus dem annotierten semantischen Modell in eine Filterung umzuwandeln. Dadurch wird die Aggregation, welche dem annotierbaren semantischen Modell zu Grunde liegt, verändert und wirkt sich entsprechend in allen verbundenen Views aus. Mithilfe dieser Mechanismen werden die Anforderungen **A4b** und **A4c** erfüllt.

6 Zusammenfassung und Ausblick

In dieser Publikation haben wir eine Erweiterung des Instrumentariums der Softwarekartographie hin zu einem interaktiven Cockpitansatz mit zusammenhängenden Views be-

schrieben. Hierzu ist eine Verknüpfung des Datenbestands einer Unternehmensarchitektur mit Metadaten notwendig. Der Datenbestand der Unternehmensarchitektur umfasst die Dokumentation der Unternehmenslandschaft. Die Metadaten werden für die Umsetzung der Interaktionen, bspw. das Annotieren, genutzt. Bei der Ausführung von Interaktionen werden ausschließlich die Metadaten verändert. Die Beschreibung der Unternehmenslandschaft bleibt unberührt. Auf diese Daten erfolgt ausschließlich ein Lesezugriff. Aus diesem Grund ist eine transaktionale Unterstützung nicht notwendig.

Der Vorteil des vorgestellten Ansatzes im Vergleich zu statischen Darstellungen und Berichten ist, dass die Stakeholder jederzeit den aktuellen Stand der Unternehmensarchitektur im Blick haben. Statische Reports werden dagegen zu einem bestimmten Zeitpunkt erstellt und abgelegt. Daher ist nicht gewährleistet, dass jederzeit Entscheidungen auf Basis des aktuellen Datenbestands getroffen werden. Ein weiterer Vorteil ist die Filterung der Informationen. Eine Veränderung der Datenbasis erfordert hierbei die Generierung eines neuen Berichts. In unserem Ansatz ist die Definition eines Blickwinkels grundsätzlich beschrieben und konfiguriert. Eine Änderung des Filters ist in Echtzeit während der Analyse durch den Stakeholder möglich.

Besonders Anforderung **A8** ist für weitere Forschung im Bereich *Corporate Intelligence* von großer Bedeutung. Dazu wird der hier vorgestellte Ansatz als Grundlage für eine prototypische Realisierung eines *Corporate Intelligence* Cockpits verwendet, an dem anschließend konkrete Überlegungen und Analysen zur Verwendbarkeit unternommen werden können.

Literatur

- [Ach10] Philip Achenbach. *Eine domänenspezifische Sprache zur Generierung von Sichten der Unternehmensarchitektur*. Bachelor's thesis, Fakultät für Informatik, Technische Universität München, 2010.
- [BEL⁺07a] Sabine Buckl, Alexander M. Ernst, Josef Lankes, Kathrin Schneider und Christian M. Schweda. A Pattern based Approach for constructing Enterprise Architecture Management Information Models. In *8. Internationale Tagung Wirtschaftsinformatik*, Seiten 145–162, Karlsruhe, 2007.
- [BEL⁺07b] Sabine Buckl, Alexander M Ernst, Josef Lankes, Christian M Schweda und André Wittenburg. Generating Visualizations of Enterprise Architectures using Model Transformations. In *2nd International Workshop on Enterprise Modelling and Informations Systems Architectures - Concepts and Applications*, Seiten 33–46, 2007.
- [BELM08] Sabine Buckl, Alexander M. Ernst, Josef Lankes und Florian Matthes. Enterprise Architecture Management Pattern Catalog (Version 1.0). Bericht, TU München, 2008.
- [BGP11] Tobias Brückmann, Volker Gruhn und Max Pfeiffer. Towards Real-Time Monitoring and Controlling of Enterprise Architectures Using Business Software Control Centers. In *ECSA'11 Proceedings of the 5th European conference on Software architecture*, Seiten 287–294, Essen, 2011. Springer Berlin / Heidelberg.
- [BGS10] Sabine Buckl, Jens Gulden und Christian M. Schweda. Supporting ad hoc Analyses on Enterprise Models. In *5th International Workshop on Enterprise Modeling and Information Systems Architectures (EMISA2010)*, Karlsruhe, 2010.

- [ELSW06] Alexander M Ernst, Josef Lankes, Christian M Schweda und André Wittenburg. Using Model Transformation for Generating Visualizations from Repository Contents - An Application to Software Cartography. Bericht, Technische Universität München, Institut für Informatik, Lehrstuhl für Informatik 19, 2006.
- [Han10] Inge Hanschke. *Strategisches Management der IT-Landschaft : Ein praktischer Leit-faden für das Enterprise Architecture Management*. Hanser Verlag, München, 2., er-weit. Auflage, 2010.
- [LMW05] Josef Lankes, Florian Matthes und André Wittenburg. Architekturbeschreibung von Anwendungslandschaften: Softwarekartographie und IEEE Std 1471-2000. In *Ligges-meyer,P.; Pohl, K.; Goedicke, M. (Hrsg.): Software Engineering 2005*, Seiten 43–54, Essen, 2005. Köllen Druck Verlag.
- [MBLS08] Florian Matthes, Sabine Buckl, Jana Leitel und Christian M. Schweda. Enterprise Ar-chitecture Management Tool Survey 2008. Bericht, Technische Universität München, München, 2008.
- [Mod97] David Modjeska. Navigation in Electronic Worlds: A Research Review. Bericht, Com-puter Systems Research Group, University of Toronto, 1997.
- [Sch11] Christian M. Schweda. *Development of Organization-Specific Enterprise Architecture Modeling Languages Using Building Blocks*. Dissertation, Fakultät für Informatik, Technische Universität München, Germany, 2011.
- [SMR12] Michael Schaub, Florian Matthes und Sascha Roth. Towards a Conceptual Framework for Interactive Enterprise Architecture Management Visualizations. In *Modellierung*, Bamberg, 2012.
- [The12] The Open Group. ArchiMate 2.0 Specification. Bericht, The Open Group, 2012.