

# Generische Einschränkung der Modellierungsfreiheit in fachkonzeptuellen Modellen

Martin Jührisch<sup>1</sup>, Gunnar Dietz<sup>2</sup>, Werner Esswein<sup>3</sup>

<sup>1</sup>Westfälische Wilhelms-Universität Münster  
Projekt MIRO  
Röntgenstr. 9-13  
48149 Münster  
juhrisch@uni-muenster.de

<sup>2</sup>Universität Hamburg  
Projekt eCampus II  
Schlüterstr. 70  
20146 Hamburg  
gunnar.dietz@uni-hamburg.de

<sup>3</sup>Technische Universität Dresden  
Lehrstuhl für Wirtschaftsinformatik,  
insbes. Systementwicklung  
01062 Dresden  
werner.esswein@tu-dresden.de

**Abstract:** Die zunehmende Verwendung von Informationsmodellen über technische Softwareaspekte hinaus, die ansteigende Anzahl an Modellerstellern sowie die Forderung nach Modellvergleichbarkeit bzw. Modellbewertbarkeit machen eine an Konventionen orientierte konstruktive Einschränkung der Freiheitsgrade bei der fachkonzeptuellen Modellierung notwendig. In diesem Artikel wird eine generische Methode vorgestellt, welche durch die Einführung von „Beschreibungsrahmen“ (Description Kits) sowohl eine Einschränkung der Modellierungsfreiheit in Bezug auf natürlichsprachliche Aspekte in fachkonzeptuellen Modellen erlaubt, als auch eine restriktive Benutzung bestehender Modellierungssprachen ermöglicht. Als Anwendungsfall wird die Konfiguration service-orientierter Architekturen diskutiert und die Nützlichkeit des Ansatzes in einer Pilotstudie illustriert, durchgeführt im Projekt MIRO an der Universität Münster.

## 1 Einleitung

In der Fachliteratur wird die Einschränkung der Modellierungsfreiheit insbesondere unter dem Aspekt der Modellintegration bei verteilten Modellierungsprojekten diskutiert [Be07]. Im Rahmen der konzeptuellen Modellierung werden Unternehmen systematisch analysiert und mit Hilfe semiformaler Modellierungssprachen rekonstruiert [Be95]. Der hohe Freiheitsgrad bei der Modellierung und die fehlende Standardisierung von Modellelementen in semiformalen Modellierungssprachen führen zu einer Reihe von Integrationskonflikten bei der Zusammenführung fachkonzeptueller Modelle [Pf07].

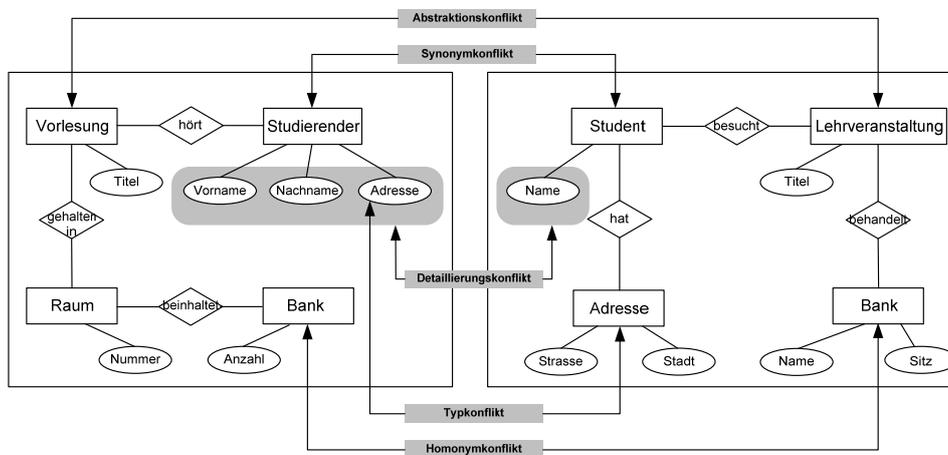


Abb. 1: Verschiedene Arten von Integrationskonflikten (in Anlehnung an [Pf07])

Modellübergreifende Referenzen können z. B. durch unterschiedliche Begriffssysteme nicht aufgelöst und Überlappungsbereiche zwischen den Teilmodellen durch unterschiedliche Abstraktionsgrade nicht identifiziert werden. Zur Vermeidung der Konflikte und zur Herstellung der Vergleichbarkeit von Modellen existieren nun zwei Ansätze: Zum einen wenden Ansätze zur Modelltransformation syntaktische Transformationen sowie semantische Tests an, um Überlappungsbereiche zu identifizieren [Ge07]. Diese Ansätze stellen nur minimale Annahmen an das vorliegende Modell. Zum anderen wird bei der Konstruktion der Modelle angesetzt, um die Konstruktion einfach vergleichbarer Modellierungsartefakte zu begünstigen. Ansätze dieses Bereichs basieren auf der Annahme, dass der Vergleich von Modellen mit willkürlicher Struktur die Identifikation von semantischen Überlappungsbereichen behindert. Becker et al. halten fest, dass, um in einem verteilten Modellierungsprojekt vergleichbare Resultate zu erhalten, die Freiheitsgrade des Modellierers stark eingeschränkt werden müssen [Be07]. Dies bedeutet, dass bei Vorliegen eines identischen Sachverhalts verschiedene Personen das gleiche Sprachkonstrukt zur Beschreibung nutzen sollten [Be95].

Basishypothese ist damit die Annahme, dass Einschränkungen bei der Modellierung den späteren Vergleich der Teilmodelle merklich vereinfachen [Pf07]. Als Konsequenz schränken Konventionen die Freiheitsgrade bei der Modellierung ein, um ein bestimmtes Maß an Compliance zu garantieren [Ro03]. Empirische Untersuchungen zeigen, dass in einem verteilten Modellierungsprozess erstellte Teilmodelle sich in Bezug auf das genutzte Vokabular, den Grad der Abstraktion und den Detaillierungsgrad unterscheiden [HS06]. Die Einschränkung der Modellierungsfreiheit für den Zweck der Schaffung vergleichbarer Teilmodelle betrifft damit im Kern die Eliminierung von Synonymen und die Schaffung semantisch disjunkter Sprachkonstrukte [Pf07]. Die Forderung nach semantisch disjunkten Sprachkonstrukten bewahrt vor Abstraktionskonflikten – alle verteilten Modelle teilen dann die gleiche Ebene der Abstraktion.

Die Modellbewertbarkeit stellt den zweiten Grund zur Einschränkung der Modellierungsfreiheit dar. Zum einen werden angesichts der zu erwartenden Fülle von Modellen innerhalb einer Prozesslandschaft manuelle Auswertungen stark erschwert. Zum anderen dienen konzeptuelle Modelle präskriptiv als Entwurf für eine zu implementierende Soll-Struktur der Organisation und ihrem Informationssystem. Die Herausforderung besteht darin, ein umfassendes Modellsystem zu erstellen, welches zum Zeitpunkt der Fertigstellung Gültigkeit besitzt – also auf aktuelle Marktanforderungen reagiert. Daher ist es sinnvoll, auf Grundlage der erhobenen Prozesslandschaft teilautomatisierte Analysen und Auswertungen als Indikatoren für zukünftige Reorganisationsmaßnahmen durchführen zu können bzw. konzeptuelle Modelle so vorzubereiten, dass sie als Input für einen Transformationsprozess in Entwurfsmodelle dienen können. Für die Auswertbarkeit ist die o. g. Forderung nach Vergleichbarkeit eine notwendige Voraussetzung [Pf07]. Zusätzlich müssen die Sprachkonstrukte eine geeignete Abstraktionsstufe und Domänennähe besitzen, um inhaltlich auswertbar zu sein.

In der vorliegenden Arbeit soll ein eigener Ansatz präsentiert werden, der die Entwicklung domänenspezifischer Sprachkonstrukte und deren eingeschränkte Nutzung in konzeptuellen Modellen generisch erlaubt. Die Methode hat ihren Ursprung in der Anpassung und Optimierung der service-orientierten Architektur an der Universität Münster [BHT07]. Organisatorische Soll-Modelle resp. Anforderungsmodelle sollen dabei die Rolle erfüllen, Input in eine Übersetzung in Servicekompositionen zu sein.

Der Artikel ist wie folgt strukturiert: Der nächste Abschnitt führt kurz die Motivation für eine Einschränkung der Freiheitsgrade bei der konzeptuellen Modellierung ein. Im dritten Abschnitt wird eine eigene Methode vorgestellt, welche die Grundlage für die generische Einschränkung der Modellierungsfreiheit in Modellen darstellt. Der Artikel schließt mit einem Anwendungsbeispiel zur Konfiguration von service-orientierten Architekturen und fasst im Ausblick wesentliche Ideen und offene Punkte zusammen.

## 2 Einschränkung der Modellierungsfreiheit

Die Essenz der Einschränkung der Modellierungsfreiheit in Bezug auf die Modellierung bestimmter Aspekte im Informationsmodell ist die Begrenzung des Sprachvokabulars auf eine Menge domänenspezifischer, semantisch disjunkter Sprachkonstrukte. Damit hat nicht nur das konstruierte Informationsmodell, sondern auch bereits die Modellierungssprache eine semantische Verbindung mit der Anwendungsdomäne [Pf07]. Aus Anwendungssicht semantisch bedeutende Operationen auf konzeptuellen Modellen können damit schon auf Sprachebene definiert werden.

Als Beispiel lässt sich die spezifische Problemsituation im Fall der verteilten Prozessaufnahme in öffentlichen Verwaltungen heranziehen. [Be07] stellen mit PICTURE eine Methode vor, mit der die Modellierungsfreiheit auf eine Menge an domänenspezifischen Sprachkonstrukten eingeschränkt wird. Durch die Eliminierung von Typ, Synonym, Homonym und Abstraktions-Konflikten kann der semantische Modellvergleich auf die Ebene der Syntax zurückgeführt werden [Pf07]. Flankiert wird die Einschränkung der zur Verfügung stehenden Fachsprache durch eine spezielle Prozessmodellierungssprache mit einfacher Syntax.

Da die Sprachkonstrukte aus der Fachsprache abgeleitet werden, sind sich Fachexperten in ihrer Rolle als Modellierer über deren Semantik bewusst. Die Wahl unpassender Modellierungskonstrukte wird vermieden, die Operationalisierbarkeit der Modelle ermöglicht.

Eine Fachsprache ist allerdings immer natürlich gewachsen, ihr Vokabular und ihre Grammatik sind nicht statisch, sondern Resultat des Sprachgebrauchs in einer Sprachgemeinschaft. Die Festlegung auf domänenspezifische Sprachkonstrukte auf Metamodellebene erscheint vor dem Hintergrund des dynamischen Wandels problematisch. Darüber hinaus besitzen domänenspezifische Modellierungssprachen keine ausreichende Anzahl an Sprachkonstrukten, um alle Phänomene in der jeweiligen Domäne abzubilden. Folgerichtig wird eine Vielzahl von domänenspezifischen Modellierungssprachen benötigt. Der hier vorgestellte Ansatz widmet sich der Entwicklung situationsabhängig adaptierbarer fachlicher Sprachkonstrukte, die für ein Projekt in einer bestimmten betrieblichen Domäne möglicherweise auch nur für einen kurzen Zeitabschnitt konstruiert und im Zeitverlauf angepasst werden. Dafür werden entweder existierende Sprachkonstrukte angepasst oder völlig neu aus existierenden Sprachfragmenten konstruiert. Ermöglicht wird dies durch die Unterscheidung zwischen Objekt- und Metamodelierungssprache.

Folgt man der Meta-Object-Facility (MOF) Architektur der Object Management Group (OMG) wird Sprache in der Systementwicklung in zwei Phasen genutzt [OMG02]:

- Zum einen kommt sie in der Modellierung auf Objektmodell-Ebene bei der Analyse und Dokumentation des Diskursbereichs zum Einsatz.

- Die Phase der Sprachbildung umfasst dagegen die Beschreibung der Syntax und Semantik einer Modellierungssprache unter Rückgriff auf eine Metasprache, die wieder als Modellierungssprache ausgelegt sein kann. In diesem Fall spricht man von einem Metamodell (M2M) [Fr99].

Da eine domänenspezifische Modellierungssprache – als künstliches Artefakt von einem Methodenentwickler geschaffen – über eine modifizierbare Grammatik verfügt, kann diese darüber hinaus frei an bestimmte Bedingungen angepasst werden. Software, die die Entwicklung einer Modellierungssprache unterstützt, wird unter dem Konzept der Meta-CASE Tools zusammengefasst. Bekannte Werkzeuge sind MetaEdit+ [KRT05] oder das cubetto® Toolset [Cu08].

### **3 Entwicklung von Beschreibungsrahmen („Description Kits“)**

Im vorliegenden Paper werden Beschreibungsrahmen (Description Kits) vorgestellt, die den Konsens der Sprachgemeinschaft bezüglich der Menge und Struktur bestimmter für die Analyse relevanter sprachlicher Konzepte repräsentieren. Der Description Kit-Ansatz ist dabei generisch genug, um jede Art von Modellinformationen in deren Beschreibung zweckbezogen einzuschränken. Konkrete Beschreibungen einer fachlichen Information im Analysemodell konkretisieren die Vorstellungen des Modellierers auf einer rein sprachlichen Ebene im Rahmen des vorgegebenen Description Kits.

#### **3.1 Description Kit Language**

Der Description Kit-Ansatz stellt die Phase der Sprachbildung in den Mittelpunkt (vgl. Anhang). Im Metamodell auf Ebene 1 erfolgt die Erstellung der so genannten Beschreibungsrahmen-Sprache (Description Kit Language). Hier findet die syntaktische Festlegung der Description Kits statt. Dies beinhaltet die Hierarchisierung verschiedener Description Kit-Konzepte [JW08] sowie die Festlegung ihrer Benutzung. Letzteres erfordert eine Verknüpfung des Metamodells einer konzeptuellen Modellierungssprache, welches zu diesem Zeitpunkt bereits vorliegen muss, und dem Metamodell für die Description Kit Language. Es wird festgelegt, welche Description Kits an welchen Modell-Elementen möglich oder verpflichtend sind. Die Description Kit Language wird also auf das Metamodell der Zielsprache abgebildet.

Die Description Kit Language kann so generisch gehalten werden, dass ein oder mehrere solcher DK nur einmal im Vorfeld erstellt werden müssen und dann in verschiedenen Kontexten benutzt werden können. Im Idealfall liegt eine Description Kit Language vor, die so generisch ist, jede Modellinformation abhängig vom vorliegenden Modellierungszweck als eingeschränktes domänenspezifisches Sprachkonstrukt modellieren zu können.

Die hier vorgestellte Description Kit Language unterscheidet zwischen zusammengesetzten und atomaren Description Kits. Zusammengesetzte Description Kits (Composite Description Kits) sind wiederum eine Aggregation aus anderen Description Kits einer niedrigeren Hierarchieebene. Atomare Description Kits (Atomic Description Kits) dagegen bestehen nicht mehr aus weiteren Rahmen. Weiter kann ein Description Kit (zusammengesetzt oder atomar) im Sinne eines Metadatensatzes durch Parameter angereichert werden. Parameter sind optionale Bestandteile eines Description Kits und beim Modellieren auf Ebene 0 auszufüllen. Allen Parametern eines atomaren Description Kits können optional entweder feste Werte oder Platzhalter zugeordnet werden. Für Parameter sind Platzhalter für frei wählbare Eigenschaften und eine nichtleere Menge von möglichen Parameterwertbelegungen spezifizierbar.

Es wäre allerdings ebenso eine Description Kit Language denkbar, die im Sinne der Modellierung von Rolleninformationen das Konzept „Rolle“ festlegt. Damit werden auf Metamodellebene keine domänenspezifischen Sprachkonstrukte modelliert, sondern die Sprache für Description Kits dieser domänenspezifischen Sprachkonstrukte. Eine Erweiterung einer Metamodellierungssprache, wie sie in der Literatur bisher vorgeschlagen wurde [JW08], ist nicht mehr notwendig.

Eine erzwungene Benutzung von bestimmten Description Kits für bestimmte Modellelemente kann dabei zusätzlich nicht nur die Modellierungsfreiheit in Bezug auf die Description Kits einschränken, sondern auch eine Einschränkung in Bezug auf die Modellierung selbst bedeuten.

### **3.2 Modellierung und Nutzung von Description Kits**

Unter Benutzung dieser Description Kit Language erfolgt dann auf Ebene 0.5 die Definition von Description Kits. Hier findet der eigentliche Adaptionsprozess statt und die Anpassung an den Modellierungszweck. Die Instanziierung von Ebene 1 auf Ebene 0.5 ist im Sinne von „dieser Description Kit entspricht diesem und jenem Konzept“ zu sehen – und legt damit auch die Benutzung der Description Kits beim Modellieren fest.

Insbesondere werden auf Ebene 0.5 adaptive Komponenten eingeführt, welche zum Teil natürlichsprachliche Konzepte darstellen. Wie schon ursprünglich beim Pattern-Ansatz [JW08] bedeutet die Einschränkung auf die auf dieser Ebene eingeführten Konzepte, die dargestellten Sprachkonflikte (vgl. Abbildung 1) teilweise aufzulösen.

Um instanziierte Description Kits (sogenannte „Descriptions“) auf Ebene 0 zu nutzen, sind Description Kits auf Ebene 0.5 zu entwerfen – im Sinne adaptierbarer domänenspezifischer Sprachkonstrukte. Dies könnten je nach Description Kit Language konkrete Rollentypen, konkrete Dokumenttypen (Email, Brief, etc.) oder konkrete Pattern [JW08] sein (vgl. Abbildung 2)

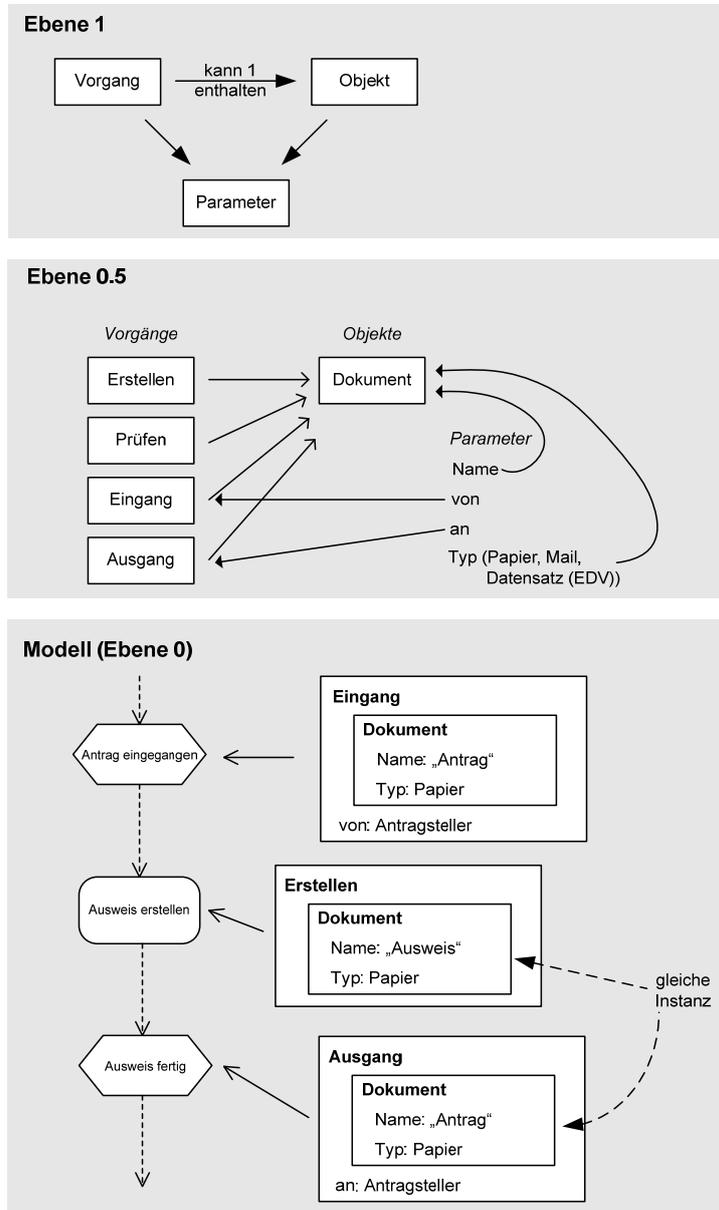


Abb. 2: Anwendungsbeispiel Description Kit Ansatz (stark vereinfacht)

Nach der Definition der Description Kits auf Ebene 0.5 kann die eingeschränkt beschreibbare Modellinformation im Modell auf Ebene 0 modelliert bzw. bereits vormodellierte Descriptions genutzt werden.

Im Beispiel der Rolleninformationen wären hier konkrete Rollen anzulegen. Jede Rolle wiederum kann unterschiedlich ausgestaltet werden und durch unterschiedliche Parameter ergänzt werden (z.B. Studienfach bei Student oder Institut bei einem Mitarbeiter). Dies wäre natürlich auch als „Freitext“ Parameter modellierbar. Genauso könnte man aber die Organisationsstruktur auf Ebene 0.5 zusätzlich zu den konkreten Rollen anlegen, so dass beim Modellieren nur noch die Organisationseinheit an die Rolle „Mitarbeiter“ angeheftet werden muss.

In einem Description Kit ist damit die Menge aller Beschreibungsmöglichkeiten eines eingeschränkten domänenspezifischen Sprachkonstrukt festgehalten, die später bei der konkreten Beschreibung modellierbar sind und durch das Setzen von Parametern konkretisiert werden können. Ein Description Kit legt somit das Vokabular und alle möglichen Beschreibungen eines domänenspezifischen Sprachkonstrukts fest.

Die Instanz eines Description Kits – also eine konkrete Beschreibung im Rahmen der Vorgaben – wird als Description bezeichnet. Die Description erhält einen eindeutigen Namen und ist Teil des fachkonzeptuellen Modells. Alle möglichen Beschreibungen, die der Description Kit zulässt, werden durch das Setzen der Parameter oder Zuordnung von Description Kits niedrigerer Hierarchieebene konkretisiert. Eine vorhandene Wertzuweisung kann während der Modellierung verändert werden. Zu jedem Zeitpunkt der Modellierung auf Ebene 0 ist immer genau eine Konfiguration eines Description Kits auf Ebene 0.5 aktiv. Wie bereits erwähnt, werden die Verbindungsmöglichkeiten des Description Kits mit den Konzepten der konzeptuellen Modellierungssprache auf Metamodellebene festgelegt und bleiben konstant. Die Description Kit Language sollte damit im Fall von Änderungen der Fachsprache nicht betroffen sein. In dem Fall wären nur die Description Kits von einer Adaption betroffen. Veränderungen an einem Description Kit, z. B. durch die Definition von weiteren Parametern, führen einen Description Kit von einer Konfiguration in eine andere über. Die Adaptionoperationen sollten dabei eine Anpassung ermöglichen, ohne die Konsistenz des existierenden fachkonzeptuellen Modells zu gefährden. Dies ist insofern problematisch, als bei der Modellierung der Description Kits keine Annahmen gemacht werden, wann bestimmte Adaptionoperationen realisiert werden.

Vorteil des hier vorgestellten Ansatzes ist die generische Modellierung domänenspezifischer Sprachkonstrukte. Gleichzeitig wird die Weiternutzung bestehender Modellierungssprachen ermöglicht. Durch die Einführung der zusätzlichen Ebene 0.5 ist ein Ansatz gefunden worden, der eine Erweiterung einer Metamodellierungssprache wie dem E<sup>3</sup>-Modell unnötig macht. Eine Unterstützung der Nähe der Ebenen 0.5 und 0 durch das Modellierungstool macht den Adaptionprozess wesentlich einfacher als im bisherig verwendeten Pattern-Ansatz [JW08]. Die Möglichkeit, unterschiedliche Description Kit Languages zu definieren, bietet eine höhere Flexibilität als der ursprüngliche (prinzipiell ja schon generische) Pattern-Ansatz und erlaubt eine direktere Umsetzung von Ansätzen wie die Anreicherung von Modellen durch Rolleninformationen im Sinne von [JWD08].

## **4 Anwendungsfall: Konfiguration service-orientierter Architekturen**

Mit der stetigen Konsolidierung der Web Service Standards vollzieht sich in der Diskussion zu service-orientierten Architekturen ein Wechsel weg von der Implementierung und dem Deployment von Services hin zum Service Management. Die Indikatoren: Anzahl der Standardisierungsanfragen und die Menge großer Forschungsprogramme in diesem Bereich sind evident für einen wachsenden Bedarf an Managementmethoden für die Abbildung von Geschäftsanforderungen auf Servicekompositionen. Es wird eine Anwendung des Description Kit-Ansatzes vorgestellt, die eine zielgerichtete Abstimmung zwischen den relevanten Geschäftsprozessen und der SOA in fachkonzeptuellen Modellen erlaubt.

Bezug nehmend auf die Abstraktionsebenen der Model Driven Architecture MDA [MD03] ordnet sich diese Abbildung als Transformationsaufgabe zwischen Analysemodellen (Computation Independent Models; CIM) und Entwurfsmodellen (Plattform Independent Models; PIM) ein. Aus dem vorliegenden Modellierungszweck lässt sich nun ableiten, dass eine automatische Überführung die Übersetzung des realweltlichen Problems in eine Form voraussetzt, die eine entsprechende maschinelle Verarbeitung ermöglicht. In der Reduzierung der semantischen Heterogenität innerhalb der Modellelemente aus Analyse- und Entwurfsmodellen liegt ein entscheidender Schritt zu einer automatisierbaren Überführung. Basishypothese ist, dass eine Verbindung zwischen Organisation und IT Domäne unter der Voraussetzung hergestellt werden kann, dass eine Menge an sprachlichen Konzepten simultan in Analyse- und Entwurfsmodellen eingesetzt wird.

Im weiteren Textverlauf wird daher die exemplarische Anwendung des Description Kit-Ansatzes für die Überwindung der semantischen Lücke zwischen semi-formaler Problembeschreibung in Analysemodellen und formaler Lösung in Entwurfsmodellen angewendet. Ziel ist die Vergleichbarkeit von Analyse- und Entwurfsmodellen zu erreichen.

Für den vorliegenden Modellierungszweck wird die Freiheit in Bezug auf die Modellierung von Objekten bzw. Teilen von Objekten eingeschränkt. Diese wurden als Konzepte gewählt, da sie zum einen in der Analyse- als auch in der Entwurfsphase Bedeutung besitzen, damit die Modellierung nicht nachhaltig negativ beeinflussen, und sich zum anderen gut für die Prüfung von Analysemodellen auf Servicekandidaten und den Abgleich zwischen Analyse- und Entwurfsmodellen eignen. Im Sinne der Description Kit-Methode wird folgendermaßen vorgegangen:

- Ebene 1 (Metamodellebene): Auf Metamodellebene wird eine hierarchische Description Kit Language entwickelt. Die Description Kit Language „Objekt“ umfasst die Konzepte Objekt, Attribut und Attributwert. Zusammen dienen sie der Modellierung von Description Kits des Typs „Objekt“ auf Ebene 0.5. Die Konzepte der Description Kit Language werden auf das existierende Metamodell der Modellierungssprache emuliert – im vorliegenden Fall wird auf das Metamodell der Ereignisgesteuerten Prozessketten (EPK) der ARIS Methode Bezug genommen [Sc00]. Damit sind die syntaktischen Beziehungen zwischen Modellierungssprache und der Description Kit Language festzulegen.
- Ebene 0.5: Mit Hilfe der Description Kit Language werden Description Kits für Objekte auf Ebene 0.5 entwickelt. Diese sind adaptiv, können also zur Laufzeit der Modellierung auf Objektmodellebene an verschiedene Anforderungen an die Modellierung von Objekten angepasst werden. In diesem Sinne bilden sie den Konsens zwischen Softwarearchitekten und Fachexperten bezüglich der Menge und Struktur bestimmter für die Analyse und den Entwurf relevanter sprachlicher Konzepte entsprechen.
- Ebene 0 (Modellebene): Auf Ebene 0 werden die Description Kits von Ebene 0.5 instanziiert und bilden individuelle Descriptions. Die Modellierung wird dabei auf den vorgegebenen Description Kit eingeschränkt.

Eine Description wird auf Ebene 0 als konkretes Objekt (z. B. Geschäftsobjekt) oder eines Teils davon interpretiert. Die Konkretisierung auf ein ganz bestimmtes Objekt erfolgt durch die Parametrisierung der Attributwerte, die u. U. durch logische Operatoren miteinander kombiniert werden können.

Description Kits vom Typ „Objekt“ haben damit einen entscheidenden Effekt auf die Aktivitäten während der Modellierung auf Modellebene, da der Freiheitsgrad bei der Beschreibung von Geschäftsobjekten bzw. Teilen von Geschäftsobjekten auf das vormodellierte Vokabular eingeschränkt wird.

#### **4.1 Konzeptuelle Modellierung mit Description Kits**

Werden Description Kits in Analysemodellen benutzt, so konkretisieren sie die Vorstellungen des Modellierers auf einer rein sprachlichen Ebene. Dafür werden Zustandsübergangsmodelle eingeführt, die basierend auf einer Erweiterung der EPK funktionale Anforderungen beschreiben (vgl. Abbildung 3).

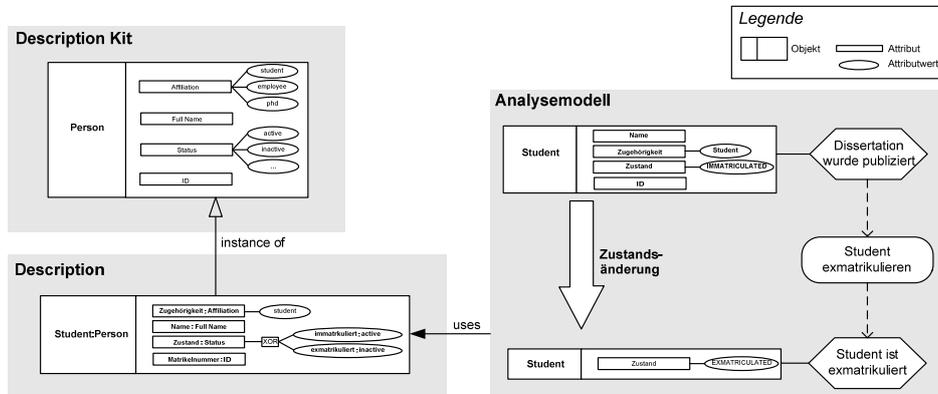


Abb. 3: Erweiterung der EPK als Zustandsübergangsmodell

Dabei wird eine semantische Beziehung zwischen Ereignis und Objekt hergestellt und als betrieblicher Zustand interpretiert. Dieser expliziert sich durch die Sammlung der Zustände der zugeordneten Objekte. Zu automatisierende Geschäftslogik wird damit als Menge von Zustandsübergängen von Objekten dokumentiert.

#### 4.2 Entwurfsmodellierung mit Descriptions

Mit der E<sup>3</sup>-Methode [Gr04] wurde ein Modell für Services entworfen, das die Abbildung einer Serviceschnittstelle mit Hilfe grafischer Zeichen vorsieht und die Methodensignatur an Descriptions koppelt.

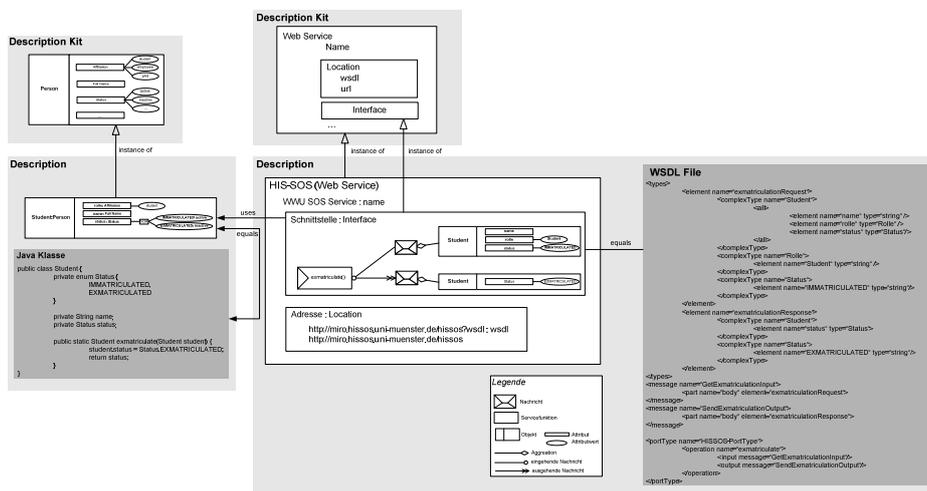


Abb. 4: Modellierung von Web Services mit dem Web Service Description Kit

Bei der Definition der Nachrichtenparameter kommen die Description Kits von Objekt zum Einsatz. Dadurch, dass bei der Modellierung von Analyse- und Entwurfsmodellen in beiden Fällen entsprechende Descriptions benutzt werden, wird die Brücke zwischen beiden Domänen geschlossen. Es wird angenommen, dass die Servicefunktion eines Web Services sich dann für einen bestimmten betrieblichen Kontext eignet, wenn das auslösende respektive resultierende Ereignis einer Prozessfunktion im Analysemodell Descriptions mit einem Zustand umfasst, der dem im Modell der Servicefunktion entspricht. Die Konfiguration wählt automatisch passende Servicefunktionen aus und schlägt dem Modellierer gegebenenfalls eine Anpassung des Soll-Modells vor. Als Ergebnis ist das EPK Modell idealerweise mit der vorliegenden SOA abgeglichen. Ein angepasstes Soll-Modell kann als Referenzmodell – fachliche Lösung mit unterliegender Implementierung – auf andere Organisationen übertragen werden.

## **Ausblick**

Der vorgestellte Ansatz leistet einen ersten Schritt hin zu einer modellgetriebenen Konfiguration einer SOA durch Geschäftsprozessmodelle. Mit der Produktivschaltung von cubetto® Toolset [Cu08] in naher Zukunft und dem dann folgenden Einbezug der dezentralen IT-Betreuungsorganisationen der Universität wird sich auch die Akzeptanz der Methode abzeichnen. Die konsequente Verwendung des Modellierungswerkzeugs im Projekt MIRO wird helfen, es als Mittel der Wahl für das Dokumentieren und Verwalten der SOA zu verwenden und als Teil des integrierten Informationsmanagements zu betrachten.

## **Literaturverzeichnis**

- [Be07] Becker, J. et al.: Bausteinbasierte Modellierung von Prozesslandschaften mit der PICTURE-Methode am Beispiel der Universitätsverwaltung Münster. In *Wirtschaftsinformatik* 49 (4), 2007; S. 267
- [Be95] Becker, J.: Grundsätze ordnungsgemäßer Modellierung. In *Wirtschaftsinformatik* 37 (5), 1995; S. 435
- [BHT07] Böhm, B.; Held, W.; Tröger, B.: Integrated Information Management at the University of Munster. In *Changing Infrastructures for Academic Services*. Bad Honnef, 2007.
- [Cu08] Cubetto Toolset: Semture GmbH, [www.semture.de](http://www.semture.de), Dresden, 2008.
- [Fr99] Frank, U.: Conceptual Modelling as the Core of the Information System Discipline – Perspectives and Epistemological Challenges. In *Proceedings of the 5<sup>th</sup> Americas Conference on Information Systems, AMCIS'99*, 1999; S. 695 – 697.
- [Ge07] Gehlert, A.: Migration fachkonzeptueller Modelle. Logos Berlin, 2007

- [Gr04] Greiffenberg, S.: Methodenentwicklung in Wirtschaft und Verwaltung. Dr. Kovac, Hamburg, 2004
- [HS06] Hadar, I.; Soffer, P.: Variations in Conceptual Modeling: Classification and Ontological Analysis. In Journal of the AIS 7 (8), 2006; S. 568-592
- [JW08] Juhrisch, M.; Weller, J.: Connecting Business and IT – A Model-driven Web Service based Approach. In Proceedings of the 12<sup>th</sup> Pacific Asia Conference on Information Systems (PACIS 2008), Suzhou, China, 2008
- [JWD08] Juhrisch, M.; Weller, J.; Dietz, G.: Application Access Control using Enterprise Models. In Proceedings of the 12<sup>th</sup> Pacific Asia Conference on Information Systems (PACIS 2008), Suzhou, China, 2008
- [KRT05] Kelly, S.; Rossi, M.; Tolvanen, J. P.: What is Needed in a MetaCASE Environment? In Enterprise Modelling and Information Systems Architectures, 2005; S. 22 – 35.
- [MD03] MDA: Guide Version 1.0.1, Document Number omg/2003-06-01, 2003.
- [OMG02] Object Management Group: Meta Object Facility (MOF) Specification, version 1.4, 2002.
- [Pf07] Pfeiffer, D.: Constructing comparable conceptual models with domain specific languages. In Proceedings of the 15<sup>th</sup> European Conference on Information Systems (ECIS 2007), St. Gallen, 2007
- [Ro03] Rosemann, M.: Preparation of Process Modeling. In (Becker, J.; Kugeler, M.; Rosemann, M. Hrsg.) Process Management, Berlin et al., 2003; S. 41-78
- [Sc00] Scheer, A.-W.: ARIS – Business Process Modeling. Springer, Berlin, 2000
- [St98] Strahinger, S.: Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. In Proceedings of the Modellierung 98 – Astronomical Society of Australia, 1998

## Metastierungsbeziehungen (in Anlehnung an [St98]):

Annahme - Sprachbasierte Metastierung

