

# Reflexionen zur sprachlichen Konstruktion von Informationssystemen

Ulrich Frank

## 1 Einleitung: Eine kurze Bestandsaufnahme

Die Entwicklung, Einführung und Nutzung von Informationssystemen hat in den zurückliegenden Jahrzehnten eine deutliche Professionalisierung erfahren. Sie macht sich einerseits daran fest, dass eine wachsende Zahl von Entscheidungsträgern einschlägige akademische Abschlüsse aufweist – was sich in einer zunehmenden Differenzierung und Konsolidierung korrespondierender Fachsprachen äußert. Andererseits hat die Leistungsfähigkeit von Entwicklungswerkzeugen, Basissystemen (z. B. Middleware) wie auch von Anwendungssystemen deutlich zugenommen. Diesen Professionalisierungstendenzen zum Trotz kann jedoch kaum die Rede davon sein, dass die gegenwärtige Situation zufrieden stellend ist. Dafür gibt es eine Reihe von Indizien. Projekte zur Entwicklung oder Anpassung von Software leiden immer noch häufig unter einer falschen Einschätzung des erforderlichen Ressourcen- und Zeitbedarfs. Nicht wenige Projekte verfehlen die angestrebten Ergebnisse oder scheitern. Die Kosten der Software-Entwicklung und –wartung sind nach wie vor erheblich. Gleichzeitig lässt die Integration von Anwendungssystemen oft zu wünschen übrig. Das gilt in mehrfacher Hinsicht. So sind die in vielen Unternehmen vorhandenen Altsysteme nicht nur unzureichend integriert, sie sperren sich auch – trotz der Methoden und Werkzeuge der so genannten Enterprise Application Integration – gegen eine enge softwaretechnische Integration, weil sie auf heterogenen Konzepten beruhen. Daneben ist die Unterstützung, die Anwendungssysteme für Geschäftsprozesse und Entscheidungsszenarien bieten – also die organisatorische Integration, in vielen Fällen nicht zufrieden stellend. Der vielfache Gebrauch des Schlagworts „IT-Business-Alignment“ ist dafür ein Indiz unter vielen. Die zunehmende Bedeutung unternehmensübergreifender Kooperationsmuster – etwa in der Logistik oder der Produktentwicklung – empfiehlt darüber hinaus die Integration von Anwendungssystemen über die Grenzen eines Unternehmens hinaus. Vor diesem Hintergrund mag es auf den besorgten IT-Manager beruhigend wirken, dass eine Vielzahl von Dienstleistern mit dem Versprechen antritt, die Not zu lindern. Hier ist etwa an IT-Berater oder die Anbieter von ERP-Systemen oder so genannter Middleware zu denken. Die Freude an solchen Angeboten wird aber zum einen durch eine mitunter verwirrende, von den Marketing-Abteilungen der jeweiligen Unternehmen geprägte Begrifflichkeit getrübt, zum anderen durch Produkte und Leistungen, die zuvor geweckte Erwartungen nicht erfüllen.

Auch im Fundus der zuständigen wissenschaftlichen Disziplinen, also der Informatik und der Wirtschaftsinformatik, findet sich eine Vielzahl von Methoden und Werkzeugen, die darauf zielen, Anwendungssysteme flexibler zu gestalten, die An-

wendungsentwicklung effizienter zu machen und die organisatorische Integration von Informationssystemen zu verbessern. In dem vorliegenden Beitrag werde ich allerdings nicht auf einzelne Ansätze eingehen, sondern stattdessen zwei grundlegende Orientierungen der Forschung fokussieren, die geeignet erscheinen, die Wirtschaftlichkeit der Entwicklung, Anpassung und Nutzung betrieblicher Informationssysteme nachhaltig zu fördern: *Integration* und *Wiederverwendung*.

Eine höhere softwaretechnische Integration verspricht die Vermeidung nicht kontrollierter Redundanz sowie die Überwindung von Medienbrüchen. Eine engere organisatorische Integration des Informationssystems mit dem jeweils relevanten Handlungssystem fördert eine friktionsfreie Nutzung von Software, da Daten wie auch Funktionen und von der Software realisierte Prozesse den korrespondierenden Informationen und Aufgaben im Handlungssystem besser entsprechen. Ein höheres Integrationsniveau verspricht also effizientere Prozesse, bessere Entscheidungsunterstützung und letztlich einen höheren Automatisierungsgrad durch die Ausschöpfung ungenutzter Automatisierungspotentiale.

Wiederverwendung gilt seit langem als eine Art Königsweg wirtschaftlicher Realisierung von Software: Artefakte, die einmal sorgfältig entworfen bzw. implementiert wurden, können beliebig oft wieder verwendet werden. Während dieser Umstand auch eine zentrale Triebfeder der industriellen Produktion ist, kommt bei der Software-Erstellung noch ein wesentlicher Vorteil hinzu: Die Erstellung einzelner Exemplare erfolgt durch Kopieren, verursacht also nur marginale Kosten. Dadurch wird die Realisierung einer ökonomisch unwiderstehlichen Vision möglich: qualitativ hochwertigere Software zu geringeren Kosten. Trotz der überaus attraktiven Verheißungen, die die Wiederverwendung von Software verspricht, sind die bisher erzielten Resultate ambivalent. Auf der einen Seite finden sich beeindruckende Beispiele für die Erfolge der Wiederverwendung informationstechnischer Artefakte. Sie betreffen einerseits generische Basissoftware wie Betriebssysteme, Datenbankmanagement-Systeme oder Middleware, andererseits neu geschaffene Systemklassen, die auf die generische Unterstützung von Aufgaben im Büro- und Verwaltungsbereich gerichtet sind, also etwa Textverarbeitung, Tabellenkalkulation oder Terminverwaltung. Auch bei betrieblicher Anwendungssoftware ist ein hohes Maß an Wiederverwendung, vor allem im Bereich der ERP-Systeme, zu verzeichnen. Dennoch bleiben die Vorteile der Wiederverwendung für einzelne Anwender hier deutlich bescheidener. Das dürfte zum einen auf den mitunter erheblichen Anpassungsaufwand, der mit solchen Systeme verbunden ist, zurückzuführen sein, zum anderen auf Marktstrukturen, die teilweise durch eine eher geringe Wettbewerbsintensität gekennzeichnet sind. Noch ernüchternder erweist sich die Bilanz bei der Wiederverwendung von Komponenten. Die an den Vorzügen der industriellen Produktion orientierte Vorstellung, individuelle Software-Systeme aus hochwertigen Komponenten herstellerunabhängig zusammenzufügen, konnte bisher in nennenswerter Weise lediglich für systemnahe Komponenten realisiert werden.

Um zu untersuchen, wie den angedeuteten Widrigkeiten zum Trotz ein höheres Maß an Integration und Wiederverwendung erreicht werden kann, sind beide Begriffe näher zu analysieren. Nur so können die Voraussetzungen, die jeweils erfüllt sein müssen, differenziert erfasst werden. Dazu werde ich Informationssysteme im Sinne

Erich Ortner als *linguistische Artefakte* betrachten, die sowohl auf formalen als auch materialen Sprachkonzepten beruhen.

## 2 Integration, Wiederverwendung und Semantik

Auch wenn es sich bei Integration und Wiederverwendung um zentrale Begriffe der Wirtschaftsinformatik und der Informatik handelt, sind differenzierte Begriffsanalysen kaum zu finden<sup>1</sup>. Zu den wenigen Ausnahmen gehören Anderson (1981) und Wiederhold (1996). Das Ziel der Integration von Software-Komponenten ist die Herstellung von Kommunikationsfähigkeit. *Kommunikation* erfordert gemeinsame Konzepte, um die in Kommunikationsbeziehungen ausgetauschten Nachrichten mit Bedeutung zu versehen. Man kann zwischen *statischer*, *funktionaler* und *dynamischer* Integration unterscheiden.

Statische Integration fokussiert die Daten der zu integrierenden Komponenten. Sie ermöglicht die gemeinsame Nutzung von Daten – entweder durch den Rückgriff auf einen gemeinsamen Speicher oder durch den Austausch von Daten. Voraussetzung dafür ist ein geeignetes semantisches Referenzsystem, in dem die gemeinsamen Konzepte definiert sind – etwa in Form von Datentypen, Relationstypen oder Klassen. Die Vorteile statischer Integration sind hinlänglich bekannt. Dabei ist einerseits an die Vermeidung nicht kontrollierter Redundanz und damit inkonsistenter Systemzustände, andererseits – damit zusammenhängend – an den durch unzureichende Integration verursachten Aufwand zu denken.

Funktionale Integration wird durch die gemeinsame Nutzung von Funktionen realisiert. Sie impliziert – neben der nötigen Kommunikationsinfrastruktur – ein semantisches Referenzsystem, in dem die Schnittstellen der Funktionen spezifiziert sind. Hier ist z. B. an Funktionsbibliotheken zu denken, die in einem verteilten System unmittelbar oder über einen Mediator verfügbar gemacht werden. Offensichtlich impliziert funktionale Integration statische Integration, da nur so die Schnittstellen der Funktionen in einheitlicher Weise definiert werden können.

Software-Komponenten sind dynamisch integriert, wenn sie nach Maßgabe eines Koordinationsschemas, etwa dem Schema eines Workflow Management Systems – gemeinsam Prozesse ausführen. Um ein solches Schema spezifizieren zu können, ist nicht nur funktionale Integration angezeigt. Darüber hinaus müssen alle für die Koordination erforderlichen Ereignisse in einem semantischen Referenzsystem spezifiziert sein – und die von den beteiligten Komponenten erzeugten Ereignisse müssen diesen Referenzereignissen entsprechen. Auch die organisatorische Integration erfordert gemeinsame Konzepte: Die Begriffe der Diskurswelt des Handlungssystems sollten möglichst mit den Begriffen korrespondieren, auf denen die Software gründet. Dies ist nicht der Fall, wenn etwa das Konzept „Kostenträger“ der Diskurswelt nicht in der Rechnungswesen-Software abgebildet ist – mit der Folge, dass

<sup>1</sup> Eine ausführliche Betrachtung des Integrationsbegriffs findet sich in Frank (2008).

die Anwender mühsame und fehlerträchtige Übersetzungsleistungen vollbringen müssen.

Nachdem wir die Dimensionen der Integration in Grundzügen skizziert haben, ist noch zu klären, ob und wie ein Niveau der Integration konzeptualisiert werden kann. Ich schlage dazu vor, dass das Integrationsniveau *ceteris paribus* umso höher ist, je größer der semantische Gehalt der gemeinsamen Konzepte ist, die der Integration zugrunde liegen. Wenn etwa zwei Komponenten Nachrichten als Folge von Bytes austauschen, weil sie nur das gemeinsame Konzept „Byte“ kennen, ist das korrespondierende Integrationsniveau geringer als wenn sie z. B. Objekte der Klasse „Kunde“ austauschen. Dabei verwende ich einen Begriff von Semantik, der weitgehend mit dem Begriff von Informationsgehalt übereinstimmt: Die Semantik eines Konzepts beschreibt die Menge zulässiger Interpretationen. Je mehr denkmögliche Interpretationen dadurch ausgeschlossen werden, desto mehr Semantik hat ein Konzept. Die Beispiele in Abb. 1 veranschaulichen den Zusammenhang zwischen Semantik und Integrationsniveau.

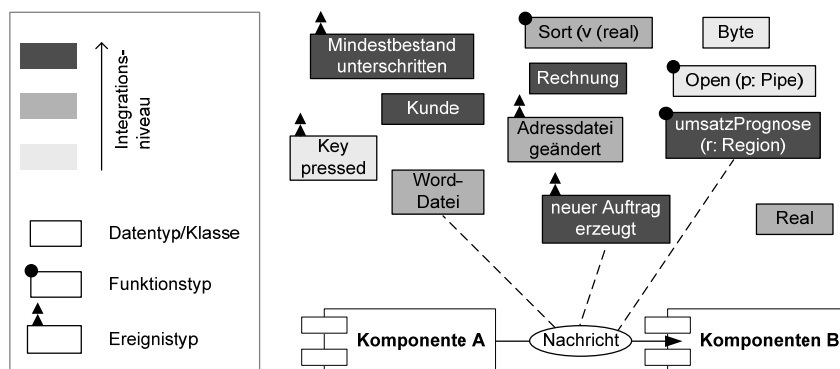


Abb. 1: Integrationsniveau in Abhängigkeit von der Semantik referenzierter Konzepte

Eine nähere Betrachtung der Beispiele in Abb. 1 macht zweierlei deutlich. Zum einen ergibt sich die Semantik eines Konzepts i. d. R. nicht allein durch die Interpretation einer isolierten Struktur, also etwa der Struktur einer Klasse. Vielmehr sind dazu auch die – in der Abbildung nicht dargestellten – Assoziationen zu anderen Konzepten zu berücksichtigen. Gegebenenfalls kann es sich dabei um vielfältige Assoziationen handeln, also gleichsam um ein ganzes Begriffsnetz – ein Umstand, auf den noch einzugehen sein wird und der an einen Satz des späten Wittgenstein erinnert: „Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache.“ (Wittgenstein 1971, § 43.). Daneben ist zu berücksichtigen, dass wir im Kontext von Informationssystemen einerseits bewusst die formale Semantik von Konzepten betonen, da sie eine wesentliche Voraussetzung für die intendierte Implementierung ist. Zum anderen erfordert die angemessene Nutzung der Konzepte auch eine materiale Semantik, die durch die Beziehung zu realweltlichen Klassen und Objekten – oder besser: zu

den Begriffen und Bezeichnen einer Diskurswelt – entsteht. Auch hier findet sich eine Parallele zu Wittgenstein: „Und die *Bedeutung* eines Namens erklärt man manchmal dadurch, dass man auf seinen *Träger zeigt*.“ (ibid). Während die Vorteile eines hohen Integrationsniveaus offensichtlich erscheinen, sollen die Nachteile hier nur kurz angedeutet werden: Je enger verschiedene Komponenten eines Systems integriert sind, desto spezieller also die der Integration zugrunde liegenden Konzepte, desto schwieriger wird die Integration weiterer Komponenten. Auch die Wiederverwendung von Artefakten ist deutlich mit deren Semantik verknüpft. Je mehr Semantik ein informationstechnisches Artefakt hat, anders gewendet: je spezieller es ist, desto größer ist der Nutzen, den es im Fall der Wiederverwendung stiftet. Gleichzeitig aber nimmt die Wiederverwendungsreichweite, also die Zahl möglicher Wiederverwendungsfälle, mit der Semantik eines Artefakts ab.

Vor diesem Hintergrund mag der Schluss nahe liegen, die Semantik informationstechnischer Artefakte im Hinblick auf ihre Integrierbarkeit und ihre Wiederverwendungsreichweite eher gering zu halten. Die Erfolge des relationalen Datenmodells, von HTML oder auch die Verheißungen so genannter lose gekoppelter Systeme könnten hier als Zeugen bemüht werden. Ich halte eine derartige Bescheidenheit für unangemessen. Zum einen ist sie mit dem Risiko verbunden, erhebliche Potentiale zur Steigerung der Leistungsfähigkeit und Wirtschaftlichkeit von Informationssystemen ungenutzt zu lassen. Zum anderen lässt sie außer acht, dass die Entwicklung linguistischer Artefakte, die sowohl ein hohes Maß an Semantik aufweisen als auch eine große Bandbreite möglicher Einsatzfälle abdecken, eine überaus reizvolle intellektuelle Herausforderung darstellt, und einem wesentliche Merkmal wissenschaftlicher Forschung entspricht. Hier sei nur beispielhaft an Popper erinnert, der von wissenschaftlichen Theorien fordert, dass sie einen hohen Informationsgehalt – und damit eine hohe Falsifikationswahrscheinlichkeit – aufweisen und gleichzeitig generell gültig sein sollten. In der Wirtschaftsinformatik wird diese Herausforderung seit geraumer Zeit vor allem mit der Entwicklung von Referenzmodellen adressiert.

### 3 Referenzmodelle

In der Wirtschaftsinformatik versteht man unter Referenzmodellen konzeptuelle Modelle, die für eine größere Zahl von Einsatzfällen Gültigkeit beanspruchen. Sie sind mit einem deskriptiven und/oder einem präskriptiven Anspruch verbunden. Anders gewendet: Ihnen liegt die Annahme zugrunde, dass sie wesentliche faktische Merkmale des intendierten Anwendungsbereichs im Hinblick auf das je vorgegebene Modellierungsziel angemessen beschreiben bzw. dass sie eine Vorlage für eine – im Hinblick auf vorgegebene Ziele – leistungsfähigere Gestaltung des Anwendungsbereichs liefern. Referenzmodelle sind gleichsam durch ein Bekenntnis zu einem höheren semantischen Niveau gekennzeichnet, denn sie spezifizieren einzelne Konzepte nicht isoliert, sondern in einem Kontext, der die Semantik wesentlich prägt. Beispiele für Referenzmodelle sind Datenmodelle, Objektmodelle und Geschäftsprozessmodelle wie auch deren Integration in Form von Unternehmensmo-

dellen. Referenzmodelle bieten also eine Grundlage für statische, funktionale und dynamische Integration.

Die Verheißungen sorgfältig entwickelter und evaluierter Referenzmodellen sind unbestritten. So stellen sie nicht nur selbst wieder verwendbare Artefakte dar; sie fördern zudem die Erstellung wieder verwendbarer Software. Daneben unterstützen sie die Realisierung hoch integrierter betrieblicher Informationssysteme – sowohl intra- wie auch interorganisational. Von besonderer Bedeutung ist dabei der Umstand, dass Referenzmodelle die Chance bieten, Softwarekomponenten unterschiedlicher Anbieter, die jeweils spezifische Teilaspekte abdecken, zu integrieren. Auf diese Weise wird der Wettbewerb zwischen Software-Anbietern gefördert – mit entsprechend positiven Effekten für die Anwender. Darüber hinaus stellen Referenzmodelle im Wortsinn eine einheitliche Sprache dar, die jenseits modischer Schlagworte und interessensspezifischer Sprachschöpfungen eine Grundlage für eine effizientere Kommunikation bietet – nicht nur zwischen Akteuren in der Praxis, sondern auch zwischen Wissenschaft und Praxis. Dabei ist nicht zuletzt an den Einsatz von Referenzmodellen in der Lehre zu denken, da sie den Studierenden eine langfristig tragfähige Grundlage zur Betrachtung und Analyse betrieblicher Informationssysteme bieten – gleichsam als Objekt und Objektivierung der Wirtschaftsinformatik.

Diesen Verheißungen zum Trotz blieben die Erfolge von Referenzmodellen bisher eher bescheiden. Das mag zum einen daran liegen, dass der Aufwand, der mit der Erstellung und Verbreitung von Referenzmodellen verbunden ist, die Möglichkeiten auch größerer Forschungseinrichtungen übersteigt. Dabei ist auch zu berücksichtigen, dass die Entwicklung von Referenzmodellen für Unternehmen mit erheblichen wirtschaftlichen Risiken verbunden ist. Wissenschaftler sehen sich dem Problem gegenüber, dass vor allem Publikationen als Leistungsnachweis zählen, während ein Beitrag zu einem Referenzmodell für die Karriere weniger einträglich ist. Eine ausführliche Analyse der Hemmnisse, die der Entwicklung von Referenzmodellen bisher im Weg standen, findet sich in Frank und Strecker (2007).

Daneben sind mitunter allerdings grundlegende Zweifel hinsichtlich der Machbarkeit von Referenzmodellen zu vernehmen. Sie betreffen vor allem die Varianz der Anforderungen in den jeweils fokussierten Domänen. Dabei ist vordergründig an die zum Teil subtile Vielfalt von Bezeichnern zu denken. Auch wenn Bezeichner keine formale Semantik tragen, kommt ihnen für die Wiederverwendung von Referenzmodellen doch eine große Bedeutung zu. Sie stellen gleichsam die Referenz auf die korrespondierenden Begriffe der jeweiligen Diskurswelt dar und tragen damit wesentlich zur Verständlichkeit von Referenzmodellen bei. Neben Homonymen und Synonymen sind landessprachliche und kulturspezifische Besonderheiten zu berücksichtigen. Dieser Art von Varianz kann durch geeignete Verzeichnisse, die die Menge der Bezeichner dem jeweils relevanten Kontext zuordnet, begegnet werden. Problematischer ist die semantische Varianz der Konzepte, die sich sozialwissenschaftlich mit dem Hinweis auf die Kontingenz sozialer (bzw. sozio-technischer) Systeme belegen lässt und die dem verbreiteten Urteil entspricht, wonach jedes Unternehmen ganz eigentümliche Besonderheiten aufweist. Dieser Einwand kann dadurch entschärft werden, dass die konkrete Organisation von Handlungssystemen und die mit ihr einhergehende Entwicklung korrespondierender Diskurswelten wohl kaum Aus-

druck einer rationalen Planung, sondern auch Ergebnis einer zum Teil willkürlichen Entwicklung ist. Wenn Varianz nicht nur ein Reflex auf notwendige Differenzierung ist, gibt es Potentiale für eine Vereinheitlichung von Terminologien. Dabei ist auch zu berücksichtigen, dass der effektive Einsatz von Informationstechnologie häufig eine Reorganisation tradierter Handlungsmuster nahe legt. Es geht also nicht nur um die Reduktion faktischer Varianz, sondern auch darum, durch präskriptive Entwürfe zukünftiger Varianz entgegen zu wirken. Die Verbreitung so genannter betrieblicher Standardsoftware und der durch sie geprägten Begriffe ist ein Beleg dafür, dass ein solcher Ansatz funktionieren kann. Dessen ungeachtet verbleiben Zweifel daran, dass Referenzmodelle mächtig genug sein können, eine im Zeitverlauf stabile Orientierung für die Gestaltung betrieblicher Informationssysteme zu bieten. Die Fortschritte in der Informations- und Kommunikationstechnologie sowie die zunehmende einschlägige Qualifizierung der involvierten Akteure werden neue Produktarten, neue Muster kooperativen Handelns sowie neue Geschäftsmodelle hervorbringen, deren Beschreibung neue Konzepte erfordert. Ein Ansatz, diesem Einwand zu begegnen, stellen Referenzmodelle auf einem höheren Abstraktionsniveau dar. Sie erfordern allerdings komplexere, mehrschichtige Spracharchitekturen.

#### 4 Mehrschichtige Spracharchitekturen

Referenzmodelle bzw. korrespondierende Schemata ermöglichen eine Kommunikation zwischen zwei Komponenten durch die Referenz auf gemeinsame Konzepte. Beide Komponenten müssen also über eine entsprechende Referenz verfügen. Menschliche Kommunikation kann auch ohne eine solche Einschränkung gelingen: Wenn auf einen Begriff verwiesen wird, der einem der Kommunikationspartner nicht bekannt ist, besteht die Möglichkeit, diesen Begriff durch Rückgriff auf gemeinsame Begriffe zu beschreiben. In ähnlicher Weise kann dies auch für die Kommunikation zwischen Komponenten realisiert werden, wenn sie über eine gemeinsame Sprache verfügen, die die Spezifikation neuer Konzepte erlaubt. Betrachten wir dazu das folgende Beispiel: Zwei Komponenten tauschen Produktdaten aus. Die Semantik der jeweils übermittelten Daten ergibt sich durch Verweis auf die in einem gemeinsamen Referenzmodell spezifizierte Produktklasse. Nun wird im Kontext von Komponente A eine neue Produktklasse „Lawnmower“ kreiert. Diese Klasse ist nicht Bestandteil des Referenzmodells, so dass Komponente B die empfangenen Daten nicht angemessen interpretieren könnte. Ein erster Ansatz zur Beschreibung der neuen Produktklasse ist in Abb. 2 dargestellt. Dazu wird auf ein gemeinsames Metamodell (also ein Referenzmodell auf einem höheren Abstraktionsniveau) verwiesen, das die Spezifikation von Objektmodellen – und damit: von Klassen – erlaubt.

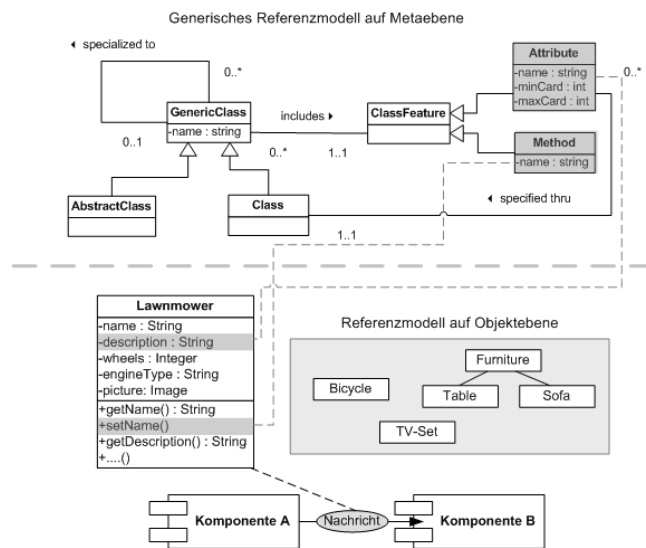


Abb. 2: Spezifikation neuer Konzepte mittels einer generischen Sprache

Ein solches Modell ist vergleichbar mit dem Metamodell, das der Spezifikation objektorientierter Modellierungssprachen – wie etwa der UML – dient. Der Verweis auf die gemeinsame Metaebene erlaubt die Beschreibung der neuen Klasse. Eine solche Beschreibung bleibt allerdings unbefriedigend, da letztlich nur spezifiziert wird, dass es sich hier um eine Klasse mit bestimmten Attributen und Methoden handelt. Es wird also völlig davon abstrahiert, dass hier eine Produktklasse vorliegt. Ein generisches Metamodell dieser Art erlaubt zwar die Beschreibung beliebiger neuer Klassen, allerdings auf einem sehr niedrigen semantischen Niveau – mit entsprechenden Konsequenzen für die Systemintegrität. Besser geeignet ist ein Metamodell, das auf einem höheren semantischen Niveau angesiedelt ist – und damit ein höheres Integrationsniveau verspricht. Ein solches Metamodell ist vergleichbar mit der Spezifikation einer domänenspezifischen Modellierungssprache. Das Metamodell in Abb. 3 illustriert dies an einem vereinfachten Beispiel einer Sprache zur Beschreibung von Produktklassen. Eine ausführliche Darstellung eines entsprechenden Metamodells und seiner Nutzung in der Systementwicklung findet sich in Frank (2002). Die dargestellten Konzepte könnten mit weiteren Integritätsbedingungen angereichert werden, um auf diese Weise semantisch noch gehaltvollere Spezifikationen von Produktklassen zu ermöglichen.



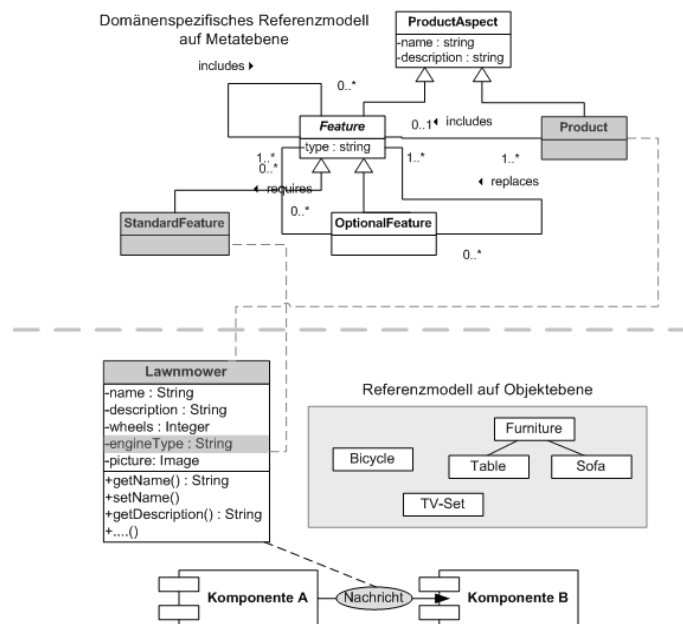


Abb. 3: Spezifikation neuer Konzepte mittels einer domänenspezifischen Sprache

Während das Beispiel in Abb. 3 auf Metamodelle zur Beschreibung statischer Abstraktionen gerichtet ist, können Metamodelle auch dazu verwendet werden, Funktionen zu spezifizieren – sowohl auf der Ebene der Interfaces als auch auf der Ebene des Kontrollflusses. Mehrschichtige Spracharchitekturen, die Referenzmodelle auf verschiedenen Abstraktionsebenen integrieren, bieten ein höheres Abstraktionsniveau und damit eine deutlich bessere Chance Varianz zu überwinden. Gleichzeitig stellt ihre Entwicklung, Wartung und Nutzung deutlich höhere Anforderungen an die jeweils betroffenen Gruppen. Dabei ist nicht zuletzt daran zu denken, dass gegenwärtige Systemarchitekturen eine weitere Abstraktionsebene über der Typ- oder Klassenebene in der Regel nicht vorsehen. Man ist also gezwungen, die Typ- oder Klassenebenen semantisch zu überladen – was nicht nur das Verständnis erschwert, sondern auch die Handhabung deutlich aufwändiger macht. Vor diesem Hintergrund ist die Etablierung mehrschichtiger Architekturen, wie sie in Forschungsprototypen und wenigen kommerziellen Entwicklungsumgebungen schon seit geraumer Zeit existieren, ein Erfolg versprechender Ansatz zur Realisierung besonders flexibler, hoch integrierter betrieblicher Informationssysteme. Dabei ist auch zu berücksichtigen, dass domänenspezifische Modellierungssprachen den Systementwurf erheblich unterstützen können – ein Umstand, der seit einiger Zeit im Zusammenhang mit dem Schlagwort „Model Driven Architecture“ diskutiert wird. So erhöhen sie die Produktivität durch die Wiederverwendung domänenspezifischer Konzepte. Gleichzeitig tragen sie zu einer höheren Qualität bei, da sie den Spielraum möglicher Instanzierung gegenüber generischen Modellierungssprachen deutlich einengen – also we-

niger unsinnige Modelle zulassen. Nicht zuletzt fördern sie durch die Einführung einer speziellen grafischen Notation die Verständlichkeit von Modellen. Während sich die Ausführungen über Referenzmodelle auf der Objektebene wie auch auf der Metaebene auf die konzeptuelle Modellierung beziehen, gelten sie in ähnlicher Weise für Ontologien, wobei in Ontologien nicht immer eine explizite Differenzierung von Objekt- und Metaebene vorgenommen wird.

## 5 Herausforderungen

Den Vorzügen eines hohen Integrationsniveaus stehen erhebliche Herausforderungen gegenüber. Sie betreffen zum einen die kulturellen, ökonomischen und politischen Hindernisse, denen sich die Entwicklung und Verbreitung von Referenzmodellen gegenübersteht. Aus der Perspektive der Anwender impliziert die Nutzung von Referenzmodellen vielfach nicht nur eine Anpassung tradierter Handlungsmuster, sondern auch die Übernahme der damit verbundenen Begrifflichkeit. Systementwickler, die Referenzmodelle wieder verwenden, müssen ebenfalls zur Anpassung bereit sein – was u. U. die Überwindung des berüchtigten „not invented here“-Syndroms einschließt. Aus ökonomischer bzw. politischer Perspektive ist zu berücksichtigen, dass die Entwicklung und Verbreitung von Referenzmodellen mit einem erheblichen Aufwand verbunden ist, der die finanziellen Möglichkeiten vieler Unternehmen übersteigt. Gleichzeitig ist es fraglich, ob öffentlich verfügbare Referenzmodelle mit den Interessen großer Unternehmen vereinbar sind. So bringen entsprechende Investitionen selbst für solche Unternehmen erhebliche Risiken mit sich. Das liegt zum einen an der Komplexität und den Unwägbarkeiten des Gegenstands, zum anderen daran, dass die Wirkungen, die Referenzmodelle auf das Geschäftsmodell marktbeherrschender Anbieter haben, kaum vorhersehbar sind.

Daneben ist an Herausforderungen zu denken, die das Thema für die wissenschaftliche Forschung bereithält. Sie können differenziert werden in die Stützung ontologischer und epistemologischer Grundannahmen sowie die Realisierung leistungsfähiger Formen der Organisation von Forschung. Die Machbarkeit von Referenzmodellen basiert auf einer *ontologischen* Hypothese: Die beobachtbare Varianz faktischer Ausprägungen und Nutzungsformen von Informationssystemen ist weniger Ausdruck wesentlicher Unterschiede, sondern ein Resultat kultureller, in Teilen willkürlicher Entwicklungsprozesse. Eine weit reichende Vereinheitlichung ist deshalb möglich ohne die Funktionsfähigkeit der betroffenen Handlungssysteme einzuschränken. Diese ontologische Hypothese allein ist allerdings nicht hinreichend, um die Machbarkeit von Referenzmodellen zu stützen. Vielmehr muss sie durch eine *epistemologische* Hypothese ergänzt werden, die besagt, dass es auch möglich ist, die vermuteten Gemeinsamkeiten zu erkennen – bzw. zu konstruieren. Beide Hypothesen können weder bewiesen noch widerlegt werden. Deshalb ist es m. E. angeraten, sie nicht einfach nur zu unterstellen – oder sie abzulehnen, sondern sich bei der Entwicklung von Referenzmodellen aktiv und engagiert mit ihnen auseinanderzusetzen. Dabei geht es nicht allein um die grundsätzliche Frage der Machbarkeit. Viel-

mehr spielen ontologische und epistemologische Annahmen auch bei Entwurfsentscheidungen eine wichtige Rolle. Diese betreffen etwa die Frage nach dem angemessenen Abstraktionsniveau eines Referenzmodells. Je weiter ein Referenzmodell von spezifischen Merkmalen einzelner Anwendungsfälle abstrahiert – also je anwendungsferner es ist, desto größer ist tendenziell seine Wiederverwendungsreichweite, desto geringer der Wiederverwendungsnutzen im Einzelfall. Aus wissenschaftlicher Sicht ist es besonders reizvoll, eine Abstraktion anzustreben, die eine hohe Wiederverwendungsreichweite ermöglicht und gleichzeitig eine komfortable und sichere Anpassung an spezielle Bedürfnisse unterstützt. Dabei ist allerdings zu berücksichtigen, dass damit eine erhebliche Herausforderung verbunden ist; vor allem, wenn die betrachtete Domäne durch ein hohes Maß an Kontingenzen und Varianz gekennzeichnet ist. Damit zusammenhängend stellt sich die Frage nach einer angemessenen Differenzierung von Sprache und Sprachanwendung, also von Metamodell und Modell. Immer dann, wenn die vorhandene und zukünftig mögliche Varianz in der Menge der intendierten Einsatzbereiche so groß ist, dass sie durch die Abstraktionskonzepte eines Referenzmodells auf Objektebene nicht sinnvoll abgedeckt werden kann, bietet sich die Erstellung eines Metamodells an. Das Metamodell sollte dabei die wesentlichen Merkmale der intendierten Einsatzbereiche abbilden, um die Erstellung semantisch gehaltvoller Modelle zu unterstützen.

Die Erstellung und Evaluation umfangreicher Referenzmodelle liegt jenseits der Kapazitäten auch größerer Forschungsgruppen an Universitäten. Deshalb ist eine Bündelung der verfügbaren Kräfte unerlässlich. Damit ist allerdings kein Plädoyer für Großforschungseinrichtungen verbunden. Vielmehr sind neue Formen wissenschaftlicher Kooperation zu entwickeln, die unterstützende Methoden und Werkzeuge, aber auch geeignete Anreizsysteme für Wissenschaftler beinhalten. Dabei ist zu berücksichtigen, dass die Entwicklung und Evaluation von Referenzmodellen auch die Beteiligung prospektiver Anwender aus der Praxis empfiehlt. Es sollten also Kooperationsmuster entwickelt werden, die auch einen effektiven Austausch mit engagierten Praktikern unterstützen. Der beachtliche Erfolg, den einige „Open Source Software“-Projekte erzielt haben, legt es nahe, ein entsprechendes Kooperationsmodell für die gemeinsame Entwicklung und Evaluation offener Referenzmodelle anzupassen. In Frank und Strecker (2007) wird eine entsprechende Initiative dargestellt. Im Portal dieser „Open Model“-Initiative (<http://openmodels.org/>) finden sich diverse frei verfügbare Modelle, die in einheitlicher Form dokumentiert sind.

## 6 Abschließende Bemerkungen

Informationssysteme sind darauf gerichtet, tatsächliche oder intendierte Handlungssysteme wirksam zu unterstützen. Gleichzeitig sind Informationssysteme sprachliche Konstruktionen. Sie basieren auf formalen Konzepten, die mit den Begriffen der jeweils relevanten Diskurswelt korrespondieren. Es handelt sich dabei allerdings nicht einfach um eine Abbildung der materialen Konzepte der Diskurswelt auf die formalen Konzepte der Systementwicklung. Sollen die Potentiale der Informationstechno-

logie ausgeschöpft werden, sind häufig eine Neuorganisation der betroffenen Handlungssysteme sowie eine Adaption der korrespondierenden Diskurswelten angezeigt. Die Entwicklung leistungsfähiger und wirtschaftlicher Informationssysteme empfiehlt also, wie Ortner bereits 1983 feststellte, eine "Rekonstruktion oder gar einen Neuaufbau von Sprache" (Ortner 1983, S. 7). Damit sind – wie in diesem Beitrag skizziert – erhebliche wissenschaftliche und ökonomische Herausforderungen verbunden. Um diesen Herausforderungen wirksam zu begegnen, ist es m. E. erforderlich, die zentrale Bedeutung von Sprache sowohl für die Forschung als auch für die Realisierung von Informationssystemen zu betonen. Diese Erkenntnis impliziert eine entsprechende Ausrichtung der Forschung in Informatik und Wirtschaftsinformatik

Aus wissenschaftlicher Sicht ist die spezifische Problematik einer Rekonstruktion von Sprache – mit den Mitteln der je verfügbaren Sprache – kritisch zu reflektieren. Hier sei an ein weiteres berühmtes Wittgenstein-Zitat erinnert: „Die Grenzen meiner Sprache bedeuten die Grenzen meiner Welt.“ (Wittgenstein 1963, 5.6) Gleichzeitig ist zu empfehlen, dass Gesellschaft und Politik die Rekonstruktion von Sprache nicht nur als eine zentrale Orientierung der einschlägigen Forschung ansehen, sondern als einen wichtigen Bestandteil einer kulturellen Entwicklung, die von erheblicher Bedeutung für die Leistungsfähigkeit von Wirtschaftssystemen ist – allerdings auch als Bedrohung empfunden werden mag. Um mit dieser Herausforderung sachkundig und differenziert umgehen zu können, ist neben einer Befähigung zur Abstraktion eine dedizierte sprachliche Kompetenz erforderlich. Es ist Erich Ortner und Hartmut Wedekind deshalb nachdrücklich zuzustimmen, wenn sie die Vermittlung dieser Kompetenzen im Informatik-Unterricht unserer Schulen fordern (so in Ordner und Wedekind 2004, Wedekind und Ordner 2004) – jenseits der heute vorherrschenden, oft halbherzig durchgeführten Einführung in die Programmierung.

## 7 Literatur

Anderson (1981)

Anderson, N. H.: Foundation of information integration theory. Academic Press: New York 1981

Frank (2002)

Frank, U.: Modeling Products for Versatile E-Commerce Platforms - Essential Requirements and Generic Design Alternatives. In: Arisawa, H.; Kambayashi, Y.; Kumar, V.; Mayr, H.C.; Hunt, I. (Eds.): Conceptual Modeling for New Information System Technologies. Berlin, Heidelberg, New York: Springer 2002, S. 444- 456

Frank und Strecker (2007)

Frank, U.; Strecker, S.: Open Reference Models – Community-driven Collaboration to Promote Development and Dissemination of Reference Models. In: Enterprise Modelling and Information Systems Architectures. Vol. 2, No. 2, 2007, pp. 32-41

Frank (2008)

Frank, U.: Integration - Reflections on a Pivotal Concept for Designing and Evaluating Information Systems. In: Kaschek, R.; Kop, C.; Steinberger, C.; Fliedl, G.: Uniscon 2008 Proceedings. Lecture Notes in Business Information Processing. Springer: Berlin, Heidelberg 2008, S. 11-22

Ortner (1983)

- Ortner, E.: Aspekte einer Konstruktionsprache für den Datenbankentwurf. Darmstadt 1983
- Ortner und Wedekind (2004)
- Ortner, E.; Wedekind, H.: Sprachkompetenz und Sprachperformanz als Basis organisationaler Intelligenz. In: Multikonferenz Wirtschaftsinformatik (MKWI), 9.-11. März 2004, Band 2. Duisburg-Essen 2004, S. 435-448
- Wedekind und Ortner (2004)
- Wedekind, H.; Ortner, E.: Toward Universal Literacy: From Computer Science Upward. In: Communications of the ACM, Volume 47, Issue 6 (June 2004), S. 101-104
- Wiederhold (1996)
- Wiederhold, G. (Ed.): Intelligent Integration of Information. Kluwer Academic Publishers: Boston 1996
- Wittgenstein (1963)
- Wittgenstein, L.: Tractatus logico-philosophicus. Suhrkamp: Frankfurt/M. 1963
- Wittgenstein (1971)
- Wittgenstein, L.: Philosophische Untersuchungen. Suhrkamp: Frankfurt/M. 1971