

Nutzungspotenziale eines Referenzdatenmodells bei der Erstellung eines fachlichen Klassenmodells

Friedhelm Krebs

SIZ

Informatikzentrum der
Sparkassenorganisation GmbH
Königswinterer Str. 552
D-53227 Bonn
www.siz.de

Friedhelm.Krebs@SIZ.de

Zusammenfassung: In der Analysephase eines Anwendungsentwicklungsprojektes steht die Erarbeitung der fachliche Zusammenhänge im Mittelpunkt, implementierungstechnische Fragestellungen werden deutlich ausgegrenzt. Es konnte festgestellt werden, dass ein begriffsorientiertes Referenzdatenmodell eine Fülle von Inhalten bereit hält, die bei der Erstellung eines fachlichen Klassenmodells genutzt werden können. Für den Übergang wurde ein Ansatz gewählt, der bekannte Abbildungsregeln nutzt, bei dem zusätzlich jedoch der benötigte Datenmodell-ausschnitt hinsichtlich seiner zugrundeliegenden Semantik hinterfragt wird, um ihn bei der Überführung in das fachliche Klassenmodell geeignet anzupassen. So gelang es, einen höheren Nutzungsgrad der im Referenzdatenmodell enthaltenen Fachlichkeit und eine bessere Qualität des daraus entstandenen Klassenmodells zu erreichen, als es bei einer automatischen Überführung des Datenmodell-ausschnitts möglich gewesen wäre.

1 Einleitung

Bankfachliche Anwendungsentwicklung beginnt nur noch selten auf der grünen Wiese. Inzwischen haben sich in vielen Teilbereichen stabile Begriffe und Strukturen herausgebildet, die in Referenzmodellen oder über Analysemuster dokumentiert sind. Vereinheitlichung der Anwendungslandschaft und Austauschbarkeit von Anwendungen sind weiterhin bedeutsame Themen. Unter dem Druck hoher Personalkosten ist es wichtiger denn je, Potenziale aus vorhandenem Wissen und Erfahrungen aus vorhergehenden Anwendungsentwicklungsprojekten möglichst gut auszuschöpfen. Ein Aspekt ist es, in objektorientierten Anwendungsentwicklungsprojekten Inhalte aus Referenzdatenmodellen zu verwenden. Die Herausforderung besteht darin, die fachlichen Inhalte so umzusetzen, dass die Klassenmodellierung unmittelbar darauf aufsetzen kann.

In einem Anwendungsentwicklungsprojekt innerhalb der Sparkassen-Finanzgruppe, in dem zur Zeit ein objektorientiertes Analysemodell erstellt wird, wurden Erfahrungen mit der Nutzung eines Referenzdatenmodells gesammelt. Es zeigt sich, dass neben dem

reichhaltigen Reservoir an bankfachlichen Begriffen und Definitionen auch Strukturen enthalten sind, die für das Analysemodell von Bedeutung sind.

Der folgende Beitrag spiegelt Erfahrungen dieses Projektes wieder und beschreibt das Vorgehen bei der Erstellung des initialen Klassenmodells auf Basis des Referenzdatenmodells.

2 Vorstellung des Projektkontextes

Gegenstand des Projektes ist die Entwicklung eines Kartenverwaltungssystems für verschiedene IT-Dienstleister der Sparkassen-Finanzgruppe. Auftraggeber ist das Informatikzentrum der Sparkassenorganisation (SIZ), dessen grundsätzliche Aufgabe es ist, innerhalb der dezentral organisierten Struktur der Sparkassen-Finanzgruppe Synergieeffekte in der IT zu erzielen. Dies geschieht durch die Erarbeitung von methodischen und technologischen Grundlagen sowie durch Koordination übergreifender Anwendungsentwicklungs-Projekte.

Das Projektteam setzt sich im Wesentlichen zusammen aus einem fachlichen Team, das mit Hilfe der Use-Case-Modellierung den Anwendungsbereich fachlich analysiert, sowie einem Klassenmodellierungs-Team, das aus jeweils 2 Mitarbeitern mit dem Schwerpunkt Klassen- bzw. Datenmodellierung besteht. Die Mitarbeiter entstammen zum großen Teil den beteiligten Häusern, ergänzt durch einige externe Berater.

Das Vorgehen im Projekt orientiert sich an einem objektorientierten Vorgehensmodell, dem AE-OO-Modell der Sparkassen-Finanzgruppe [NSK97], aus dem ein phasenorientiertes Vorgehen ausgewählt wurde.

3 Nutzungspotenziale des Referenzdatenmodells

Die fachliche Analyse hat das Ziel, die reine Fachlichkeit des Anwendungsbereiches abzubilden, ohne durch dv-technische Rahmenbedingungen eingeschränkt oder zu früh in ein bestimmtes Lösungsszenario gedrängt zu werden. Unter diesem Blickwinkel ist auch das verwendete Referenzdatenmodell zunächst hinsichtlich seiner Eignung als Referenz für ein Klassenmodell näher zu beleuchten.

3.1 Vorstellung des verwendeten Referenzdatenmodells

Das SKO-Datenmodell [WeKi98] wurde Anfang der 90er Jahre im SIZ entwickelt und wird seitdem ständig fortgeschrieben. Es handelt sich um einen begriffsorientierten Ansatz mit starkem Geschäftsbezug. Basis ist das Financial Services Data Model (FSDM) der IBM. Durch Fortschreibungen ist das bankfachliche Wissen aus einer Vielzahl von Anwendungsentwicklungsprojekten der Sparkassen-Finanzgruppe eingeflossen.

Kennzeichnend für das SKO-Datenmodell sind 3 aufeinander aufbauende Ebenen.

Die A-Ebene des SKO-Datenmodells dient dazu, den Informationshaushalt des Bankgeschäfts auf einem hohen abstrakten Niveau abzubilden. Auf der Basis von 9 Begriffen grenzt sie diesen Informationshaushalt datenseitig ab. Darin enthalten sind Leitbilder des Modells.

Die B-Ebene des SKO-Datenmodells besteht aus einem Begriffssystem, das bankfachliche Begriffe in monohierarchischer Struktur zur Verfügung stellt. Das SKO-Datenmodell

soll über seine Ebenen den gesamten bankfachlichen Datenhaushalt umfassen. Unter dem Begriff „gesamter bankfachlicher Datenhaushalt“ werden hier alle Begriffsdefinitionen einschließlich der Verweise zwischen den Begriffen verstanden. Zur Zeit enthält die B-Ebene 4118 Begriffe.

Die C-Ebene enthält ein konzeptionelles Datenmodell in ER-Notation, das die Anforderungen der 3. Normalform erfüllt. Neben Entitäten, Attributen und Beziehungen enthält die C-Ebene eine Fülle von Domänenwerten zur Beschreibung der Wertebereiche der Attribute. Weiteres Kennzeichen sind Super-/ Subtyp-Beziehungen. In der C-Ebene sind allerdings keine dv-technischen Entscheidungen eingeflossen; es sollen nur bankfachliche Begrifflichkeiten enthalten sein, die im Bereich der Sparkassen-Finanzgruppe als gültig und abgestimmt betrachtet werden können.

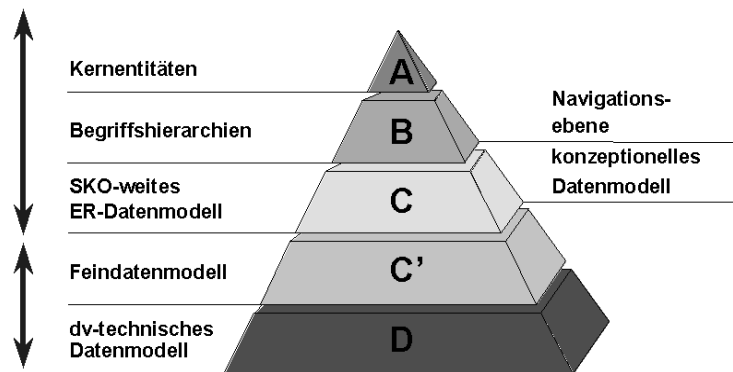


Abbildung 1: Ebenen des SKO-Datenmodells

C'- und D-Ebene beschreiben die Verwendung des SKO-Datenmodells. Sie werden nicht zentral verwaltet. Die C'-Ebene repräsentiert Ausschnitte der C-Ebene für die jeweiligen Anwendungsentwicklungsprojekte, wo sie entsprechend verfeinert werden. Die D-Ebene repräsentiert das zugehörige dv-technische Datenmodell.

Aufgrund des begriffsorientierten Ansatzes und der Abgrenzung zu dv-technischen Einflüssen verspricht das SKO-Datenmodell eine gute Basis für die objektorientierte Anwendungsentwicklung zu sein. Offen bleibt noch, wie man diese Basis im objektorientierten Kontext nutzt, ohne Einflüsse der strukturierten Vorgehensweise ins Klassenmodell zu übertragen.

3.2 Erstellen eines initialen Klassenmodells

Bevor die endgültige Entscheidung getroffen wurde, das SKO-Datenmodell bei der Klassenmodellierung heranzuziehen, wurde ein initiales Klassenmodell erstellt. Obwohl das SKO-Datenmodell aufgrund seines begriffsorientierten Ansatzes Voraussetzungen erfüllt, die die Anwendung von Abbildungsregeln zwischen Datenmodellen und Klassenmodellen zulassen [Sauer95], wurde hier kein automatisierter Übergang ins Klassenmodell verfolgt. Stattdessen wurde unter Berücksichtigung dieser Regeln, ein manueller Übergang gewählt, bei dem semantische Aspekte im Vordergrund stehen. Grund für diese Entscheidung war der Anspruch, frühzeitig zu erkennen, wenn bestimmte Modellstrukturen der Sichtweise eines Objektmodellierers nicht entsprechen und die fachlichen Inhalte auf andere Weise dargestellt werden sollten. Als wichtigster Unterschied der Sichtweisen wurde gesehen, dass der Fokus der Datenmodellierung Normalisierung ist, um Redundanzen zu vermeiden. Die Objektmodellierung strebt hingegen, auch unter Inkaufnahme von Redundanz, Verkapselung an.

Die Bestimmung des relevanten Modellausschnitts geschah auf der Grundlage von Anforderungsdefinitionen des Projektes. Im Text wurden die benötigten Begriffe identifiziert und in ein Raster aufgenommen. Dieses Raster klassifizierte die Begriffe gemäß der Kernentitäten des SKO-Datenmodells und trennte zusätzlich bereits danach, ob es sich um Kandidaten für Klassen, Attribute, Methoden oder Beziehungen handelt.

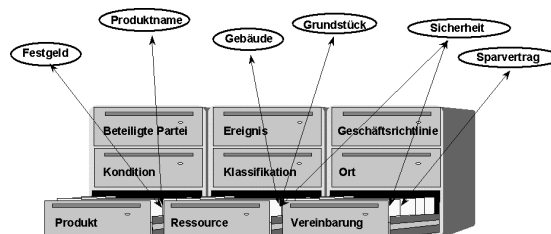


Abbildung 2: Kernentitäten des SKO-Datenmodells

Die Erfahrung bei der Suche nach Klassenkandidaten zeigte, dass die Kernentitäten geeignete Sammeltöpfe darstellen, weil sie den Begriffshaushalt sauber trennen. Voraussetzung ist allerdings, dass die Kernentitäten vertraut sind und sicher zugeordnet werden können. Im aktuellen Projekt entstand der Bedarf, die Begriffe zusätzlich nach einer operativen und einer Wissensebene zu differenzieren.

Nachdem die Suche nach Klassenkandidaten weitestgehend abgeschlossen war, wurden jene Klassen selektiert, die in dem Anwendungsbereich von herausragender Bedeutung sind. Ausgehend von diesen Klassen wurde nun im Datenmodell nach wichtigen Beziehungen gesucht. Hilfreich waren als Einstieg die auf hoher Aggregationsebene zwischen Kernentitäten modellierten Beziehungen. Die im Referenzdatenmodell identifizierten Strukturen wurden hinsichtlich der modellierungstechnischen Motivation hinterfragt. Zum Teil konnten sie unmittelbar übernommen werden, zum Teil wurden sie in anderer Weise, die aus objektorientierter Sicht geeigneter erschien, übertragen. Abschnitt 3.4 beschreibt Kriterien, in welchen Fällen eine Adaption vorgenommen wurde.

3.3 Alternatives Vorgehen auf Basis eines Geschäftsprozessmodells

Parallel zum genannten Projekt wurden Überlegungen unternommen, wie die Erstellung des initialen Klassenmodells verbessert werden kann, wenn als Basis ein Geschäftsprozessmodell zur Verfügung steht. Dabei wurde vorausgesetzt, dass das Geschäftsprozessmodell in ARIS-Notation vorliegt. Quellen für Klassenkandidaten sind hier die Bezeichner der Funktionen und Ereignisse sowie die in den Funktionszuordnungsdiagrammen verwendeten Fachbegriffe. Diese Quellen vereinfachen die Identifizierung von Klassenkandidaten, im Vergleich zur Suche in Texten. Der Weg zum Klassenmodell wird zusätzlich vereinfacht, wenn bei der Geschäftsprozessmodellierung ein definierter Begriffshaushalt verwendet wurde, der mit dem Referenzdatenmodell in Beziehung steht, z.B. eine Nomenklatur [Wegge97]. Schließlich verbessert sich die fachliche Qualität des Klassenmodells grundsätzlich, wenn die zugrundeliegenden Prozesse bekannt sind.

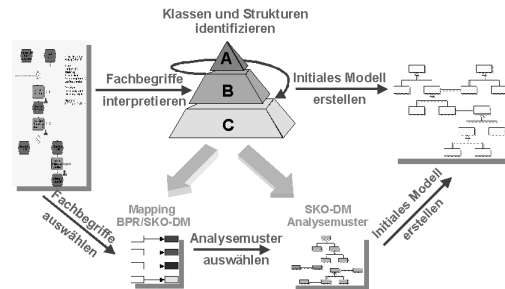


Abbildung 3: Initiales Klassenmodell auf Basis eines Geschäftsprozessmodells

3.4 Verwendbarkeit und Aufbereitung der Datenmodellstrukturen

Fokus der Überlegungen zur Verwendbarkeit der Datenmodellstrukturen ist, inwiefern es nutzbringend war, die Semantik des Datenmodells bei der Überführung in die objektorientierte Sichtweise zu berücksichtigen. Es schließt sich die Frage an, ob im SKO-Datenmodell identifizierte Muster ins Klassenmodell übernommen werden konnten.

Unterschiede ergaben sich bei der Notation von Beziehungen. Im SKO-Datenmodell werden Beziehungen über je einen Bezeichner für die Hin- und für die Rückrichtung bezeichnet. Die UML ist hier mächtiger als der im SKO-Datenmodell verwendete ER-Dialekt. Sie bietet verschiedene Möglichkeiten zur Beschreibung der Beziehung an [BRJ99]. So konnte bei der Erstellung des initialen Klassenmodells individuell entschieden werden, auf welche Weise die Beziehung am geeignetsten beschrieben werden kann. In den häufigsten Fällen wurde Hin- oder Rückrichtung als Rollenbezeichner an einem Beziehungsende umgesetzt, nur selten wurden beide Richtungen benötigt. Einige Beziehungen konnten durch Aggregation ausgedrückt werden. In den übrigen Fällen wurde ein geeigneter Name für die Beziehung selbst gesucht.

Häufig stößt man in Datenmodellen auf Art-Attribute. Mit ihrer Hilfe werden Entitäten typisiert. Im Klassenmodell wurden verschiedene objektorientierte Techniken verwendet, um dies auszudrücken. In den häufigsten Fällen wurden sie als Instanzen der Klasse in Kollaborationsdiagrammen beschrieben. Sofern es die Anschaulichkeit des Klassenmodells förderte, wurden sie als Klassen gemäß dem Singleton-Pattern [GHJV95] abgebildet.

Das SKO-Datenmodell enthält einige assoziativen Entitäten, die in Abhängigkeit der abgebildeten Semantik unterschiedlich umgesetzt werden mussten. Folgende Fälle wurden unterschieden:

- Assoziative Entitäten bilden N:M-Beziehungen ab: Sie wurden nicht in Klassenmodell übernommen, die Kardinalitäten wurden entsprechend bei den Beziehungen selbst eingetragen.
- Assoziative Entitäten fassen über Art-Attribute mehrere Beziehungen zusammen: Die Beziehungen wurden ausmodelliert.
- Assoziative Entitäten können als Rollen einer der beiden in Beziehung stehenden Entitäten aufgefasst werden: Sie wurden als Klassen modelliert und mittels Aggregation mit der zugehörigen Klasse in Beziehung gesetzt. Dies kam dem Prinzip der Verkapselung entgegen.
- Assoziative Entitäten enthalten Attribute, die auf eine andere Klasse übertragen werden können: Sie wurden aufgelöst, die Attribute wurden übertragen.
- Assoziative Entitäten enthalten Attribute, die für die Beziehung selbst relevant sind: Sie wurden als assoziative Klassen modelliert.

Gelegentlich wurden im SKO-Datenmodell generische Strukturen identifiziert, die geeignet sind, ein breites Feld von Informationen ähnlicher Semantik abzubilden. Die hier entstandenen mächtigen Datenstrukturen stehen gelegentlich in Konflikt mit dem objektorientierten Prinzip der Verkapselung. Wo es aus Sicht der Datenmodellierung sinnvoll war, möglichst viele ähnliche Sachverhalte mit dieser mächtigen Struktur zu modellieren, erschien es aus Sicht der Klassenmodellierung häufig sinnvoller zu sein, die Information als Attribute in die Klasse selbst aufzunehmen, auch unter Inkaufnahme von Redundanz. Ein Beispiel sind Konditionen.

Die im SKO-Datenmodell modellierten Super-/Subtype-Beziehungen wurden teilweise ins fachliche Klassenmodell übernommen, sofern es sich um statische, stabile Beziehungen handelte. Tendenziell wurde jedoch vorsichtig mit Vererbungsbeziehungen umgegangen, um zu vermeiden, dass aufwendig zu revidierende Strukturen zu früh festgelegt werden. Hier sind Methoden ein Aspekt, der auch einen Einfluss auf die Gestalt der Vererbungshierarchie haben kann, der im Datenmodell jedoch keine Rolle spielt.

Im Referenzdatenmodell sind einige Muster dokumentiert, die sich auch in anderen Anwendungsszenarien als stabil erwiesen haben. Ein Beispiel ist „Beteiligte Partei“. Mit diesem Muster können alle Parteien abgebildet werden, mit denen das Kreditinstitut in eine Kundenbeziehung tritt, bzw. über die das Kreditinstitut Informationen führen möchte. Ein weiteres Beispiel ist das Muster „Produkt“, das sowohl Bestandsprodukte als auch Dienstleistungen abdeckt. Es handelt sich in beiden Fällen um statische Strukturen, die nicht mit Anforderungen der objektorientierten Modellierung kollidieren. Bei der Übertragung ins Klassenmodell konnten auch die meisten Attribute übernommen werden.

Zusammenfassend zeigt sich an verschiedenen Stellen des Modells, dass syntaktisch gleiche Konstrukte des Datenmodells aufgrund ihrer Semantik auf unterschiedliche Weise ins Klassenmodell übertragen werden mussten. Es fällt auch auf, dass in nur wenigen Fällen die Ausdrucksmächtigkeit der Modellierungssprache Ursache dafür gewesen ist. Dies verstärkt den Anspruch, beim Übergang vom Daten- ins Klassenmodell die abge-

bildete Fachlichkeit genauer zu betrachten, um den unterschiedlichen Anforderungen der Paradigmen gerecht werden zu können.

4 Toolunterstützung

Der zunächst aufwendig erscheinende Weg, den identifizierten Datenmodellausschnitt nicht automatisch in ein initiales Klassenmodell abzubilden, sondern stattdessen Strukturen semantisch genauer zu hinterfragen und manuell in objektorientierte Strukturen zu überführen, konnte durch die praktische Handhabbarkeit der Modellierungs-Tools teilweise wieder relativiert werden.

Zum Auffinden von Entitäten und Beziehungen wurde ein Navigations-Tool für das Referenzdatenmodell verwendet. Die Navigationsmöglichkeiten sind äußerst mächtig und erforderten auch in dem umfangreichen Referenzmodell keine Wartezeiten. Das Überführen der soeben identifizierten Klassenkandidaten, Beziehungen oder Attribute in das objektorientierte Modellierungs-Tool war wenig zeitaufwendig, weil das Anlegen der Elemente in diesem Tool sehr unkompliziert und zügig erfolgen kann und das Copy&Paste der Bezeichner und Definitionen mit äußerst wenigen Navigationsschritten möglich ist.

Der manuelle Weg erschien schließlich kaum aufwendiger, verglichen mit der Option, die gewünschten Elemente im Datenmodellierungs-Tool zu markieren und automatisch zu überführen. Der wesentliche Vorteil bei der manuellen Vorgehensweise besteht darin, dass die im Datenmodell identifizierte Semantik unmittelbar in gewünschter Weise umgesetzt und angepasst werden kann.

Es bleibt zu untersuchen, ob das Vorgehen über einen hybriden Ansatz weiter optimiert werden kann.

5 Zusammenfassung und Fazit

Der Bericht stellt Erfahrungen dar, die bei der Nutzung eines Referenzdatenmodells zum Entwurf eines fachlichen Klassenmodells in der Analysephase eines Anwendungsentwicklungsprojektes gesammelt wurden.

Es wurde ein initiales Klassenmodell erstellt mit einem Grundstock an Klassen und Strukturen, der im Verlauf der fachlichen Analyse erweitert und durch fachliche Informationen angereichert wird.

Im Referenzdatenmodell konnten zahlreiche fachliche Begriffe und Strukturen identifiziert werden, die im Klassenmodell Verwendung finden. Wesentlicher Grund hierfür ist, dass dem genutzten Referenzdatenmodell ein begriffsorientierter Ansatz unterliegt. Ursachen für die Unterschiede zum initialen Klassenmodell waren nur in wenigen Fällen in Notationsunterschieden zu suchen, vielmehr liegt es an den Sichtweisen der Paradigmen.

Durch Nutzung des Referenzdatenmodells konnte Zeit gewonnen werden, weil Definitionen übernommen und Diskussionen über Namensgebung zum Teil eingegrenzt werden konnten. Die Qualität des Zielmodells wurde gesteigert, weil etablierte Muster einfließen konnten. Der Nutzungsgrad des Referenzdatenmodells konnte erhöht werden, weil beim manuellen Überführen der Inhalte in das Klassenmodell die Semantik hinterfragt und dann dem Paradigma entsprechend übertragen wurde. Schließlich fördert die Ver-

wendung eines Referenzmodells grundsätzlich die Vereinheitlichung von Begriffshalten und Strukturen.

Das gewählte Vorgehen hat sich bewährt. Der Mehraufwand durch das manuelle Vorgehen ist vertretbar. Optimierung ist möglich über angepasste Toolunterstützung und indem als fachliche Basis für den Modellausschnitt ein Geschäftsprozessmodell, statt einer ausschließlich textuellen Definition der Anforderungen verwendet wird.

Literaturverzeichnis

- [BRJ99] Booch, G.; Rumbaugh, J.; Jacobson, I.: The Unified Modeling Language - User Guide. Addison-Wesley 1999.
- [Fowl97] Fowler, M.: Analysis Patterns. Reusable Object Models. Addison-Wesley 1997.
- [GHJV95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley 1995.
- [NSK97] Noack, J.; Schienmann, B.; Kittlaus, H.-B.: Ein Leitfaden für die objektorientierte Anwendungsentwicklung in der Sparkassenorganisation. In: Objektspektrum 6 (1997), S. 52-59.
- [Sauer95] Sauer, P.: Ableitung eines konzeptuellen Objektmodells aus einem konzeptuellen Datenschema – ein evolutionärer Ansatz der Objektmodellierung. In: Information Management (1995) 2, S. 56-65
- [Wegge97] Wegge, D.: Nomenklatur in elektronischen Archivsystemen. In: Betriebswirtschaftliche Blätter 46 (1997) 4.
- [WeKi98] Wegge, D.; Kittlaus, H.-B.: The SIZ Banking Data Model. To be published in: Bernus, P.; Mertins, K.; Schmidt, G. (Eds.): International Handbook on Information Systems - Handbook on Architectures of Information Systems, Springer 1998.