



Business rules!

Modellieren von Geschäftsprozessen

Carsten Pflug

Die Analyse und auch das Design geschäftsprozessorientierter Systeme (z. B. für Banken und Versicherungen) birgt viel mehr Hürden als auf den ersten Blick anzunehmen ist. Bei der Entwicklung solcher Systeme wird immer stärker auf Objektorientierung mit Hilfe der Notation UML gesetzt. Doch das Werkzeug alleine lässt noch keine Hürden überwinden, es macht sie lediglich etwas kleiner. In diesem Artikel wird anhand eines Beispiels der Analyseprozess eines solchen Systems durchlaufen und gezeigt, wie die Analyseergebnisse im Design einer Java-Applikation weiterverwendet werden können.

Merkmale geschäftsprozessorientierter Systeme?

► Geschäftsprozessorientierte Systeme begegnen uns fast tagtäglich und sind mittlerweile zu einer Selbstverständlichkeit geworden. Ob Sie bei der Kfz-Zulassungsstelle ein neues Fahrzeug anmelden, eine Reise in die Karibik buchen, eine Versicherung abschließen oder ein Konto eröffnen möchten: Bei all diesen Tätigkeiten treten Sie direkt oder indirekt in Kontakt mit geschäftsprozessorientierten Systemen. Systeme dieser Art dienen in erster Linie dazu, den Menschen bei bestimmten Arbeitsabläufen zu unterstützen oder sogar die Abläufe komplett zu übernehmen.

Eine Gemeinsamkeit aller geschäftsprozessorientierten Systeme ist neben der direkten Schnittstelle zum Menschen in Form einer GUI (Graphical User Interface), dass die Geschäftsprozesse (Business Processes) gänzlich oder zum Teil im System abgebildet sind, um die Benutzer bei ihrer Arbeit zu unterstützen. Prozesse sind eine Abfolge von Tätigkeiten, deren Reihenfolge durch fachliche Regeln bestimmt und somit durch diese gesteuert werden. In den Prozessen werden Daten (z. B. die Prämie der Versicherung, die Angaben zum Kfz) erzeugt, verändert oder persistent in einem Speichermedium (z. B. einer relationalen Datenbank) abgelegt. Hauptaufgabe beim Erstellen derartiger Systeme ist es daher, die Prozesse exakt zu definieren und in dem System abzubilden.

Systeme, welche speziell für die Unterstützung von Geschäftsprozessen entwickelt werden, sind die Workflow-Management-Systeme. Diese Art von Systemen ist speziell dafür ausgelegt, vordefinierte Prozesse mit einer Workflow-Engine abzuarbeiten. Der wohl bekannteste Vertreter ist SAP.

Trotz der vielen Standardprodukte zur Unterstützung von Geschäftsprozessen auf dem Markt ist die Entwicklung geschäftsprozessorientierter Systeme keine vom Aussterben bedrohte Tätigkeit. Heute und auch in absehbarer Zukunft werden viele der Standardprodukte individuell für ein Anwendungsgebiet oder ein Unternehmen angepasst. Doch ist die Entwicklung nicht ganz so einfach, wie man es sich auf den ersten Blick vorstellen mag. Gerade die Beschreibung der Prozesse bedeutet einen nicht unerheblichen Aufwand bei der Entwicklung.

Dieser Artikel beschreibt ein mögliches Vorgehen bei der Analyse geschäftsprozessorientierter Systeme und zeigt, wie die Erkenntnisse der Analyse mit Hilfe der UML dokumentiert werden können. Weiterhin wird ein Einblick in die Möglichkeiten, die Analyseergebnisse direkt im Design weiter zu verwenden, gegeben. Das Vorgehen wird anhand eines einfachen Beispiels aus der Versicherungsbranche illustriert.

Anforderungen in der Analyse spezifizieren ...

Die Analyse dient dazu, die Anforderungen an das System zu spezifizieren und exakt zu dokumentieren. Hierzu werden zwei grundlegende Dimensionen des Systems – die Strukturen und das Verhalten – betrachtet. Meistens wird davon ausgegangen, dass die Geschäftsprozesse, die es gibt und durch das System realisiert werden sollen, alle bekannt sind. Häufig ist dies aber nicht der Fall und es wird auf einem sehr wackeligen Grund gestartet, da normalerweise die Systemanalyse die Ergebnisse der Geschäftsprozessanalyse als Grundlage nimmt. Deswegen steht der Systemanalytiker meist vor der Aufgabe, im Rahmen der Systemanalyse eine Menge an nicht vorliegendem Wissen über die Geschäftsprozesse erst einmal erheben zu müssen.

Die Strukturen

Bei der Betrachtung der Strukturen werden die Daten, die in der Domäne, in dem Anwendungsfeld, existieren, identifiziert und mit deren Beziehungen untereinander in Form eines Klassendiagramms modelliert. Die Klassen beschreiben die Objekte und beinhalten als Attribute all diejenigen Daten, welche in dem System bei den verschiedenen Szenarien, die das Verhalten des Systems beschreiben, verwendet werden. Hierbei kann es sich um Daten handeln, die eingegeben, angezeigt, erzeugt oder auch im Hintergrund geprüft werden sollen.

Bei der Analyse der Strukturen sollten Sie immer die Domäne als Analysegegenstand heranziehen, um ein Modell über die fachlichen Zusammenhänge zu erstellen. Es ist nicht das Ziel der Analyse, ein Datenbankmodell zu erstellen, sondern ein abstraktes fachliches Datenmodell.

Das Verhalten

Parallel zur fachlichen Struktur der Domäne wird das Verhalten des Systems analysiert. Der Einstieg in die Analyse des Verhaltens erfolgt über die Betrachtung der Funktionalitäten des Systems. Funktionalitäten repräsentieren einen ganzen oder einen Teil eines Geschäftsprozesses, der vom System unterstützt wird. Diese Funktionalitäten werden in Form von Use-Cases dokumentiert. Jeder Use-Case beinhaltet einen Prozess, welcher in der Analyse definiert und dokumentiert werden muss. Die Prozesse lassen sich dann sehr einfach und leicht verständlich mit Aktivitätsdiagrammen darstellen. Die Diagramme ermöglichen es, alle Prozesse, welche ablaufen sollen, in der gewünschten Detaillierungstiefe darzustellen. Da die Prozesse den Kern von geschäftsprozessorientierten Systemen ausmachen, ist es wichtig, diese sehr exakt zu modellieren. Das bedeutet, sie sehr detailliert zu beschreiben und immer wieder mit den Fachabteilungen abzusprechen.

Neben den reinen Abläufen können die im Klassendiagramm modellierten Daten, welche verarbeitet werden, in dem Dia-

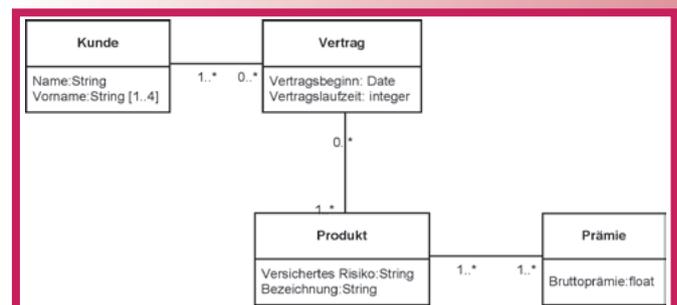


Abb. 1: Stark vereinfachtes Klassendiagramm eines Versicherungssystems



gramm mit Hilfe der Objektknoten dargestellt werden. Zusätzlich sollten die Regeln, welche die Abläufe steuern, in dem Diagramm auftauchen. Der Inhalt der Regeln selbst sollte an einem anderen Ort dokumentiert werden, da sonst die Diagramme mit Informationen überladen werden und nur noch mühselig zu verstehen sind. Für die Regeln eignet sich eine natürlichsprachliche Dokumentation oder eine formalisierte Beschreibung (z. B. Bedingung: Alter < 18, Folge: Prämie = Prämie - 5%). Achten Sie bei der natürlichsprachlichen Dokumentation der Regeln darauf, dass die Regeln eindeutig und klar verständlich beschrieben werden. Tipps und Tricks finden Sie auf www.sophist.de oder in [Rupp04]. In dem Diagramm werden die Regeln lediglich an der passenden Stelle referenziert. Wie die Regeln dargestellt werden, ist dem Analytiker überlassen. Die Referenzierung der Regeln durch Kommentare wird in Abbildung 2 gezeigt.

Der Kern der geschäftsprozessorientierten Systeme ist die Sammlung an verschiedenen Prozessen, die unter bestimmten Bedingungen abgearbeitet werden und die gewünschten Ergebnisse erzielen. Bisher haben wir die Szenarien als Abläufe, dargestellt mit Aktivitätsdiagrammen, betrachtet. Es ist aber auch möglich, die Abläufe auf Zustandsautomaten abzubilden. Da dies aber nichts an dem vorgestellten Vorgehen ändert, und damit der Artikel nicht zu umfangreich wird, werde ich hier nicht näher auf die Verhaltensmodellierung mit Zustandsautomaten eingehen.

Ein weiteres Artefakt, das bei der Analyse entstehen muss, ist eine Beschreibung der GUI (Graphical User Interface), da sie schließlich als repräsentativer Teil des Systems für dessen Akzeptanz beim Benutzer einen hohen Stellenwert hat. Für die Dokumentation der GUI gibt es viele Möglichkeiten, vom Oberflächenprototypen bis hin zur Modellierung der GUI mit Hilfe eines Zustandsautomaten. Wie dabei im Speziellen vorzugehen ist, wird hier nicht beschrieben, da dies den Rahmen des Artikels sprengen würde.

Sie werden an dieser Stelle feststellen, dass nicht alle Informationen, die in der Analyse dokumentiert werden, in Diagrammen dargestellt werden. Zum Beispiel lassen sich viele nichtfunktio-

nale Anforderungen (z. B. Häufigkeit der Abarbeitung eines Prozesses am Tag) nicht oder nur sehr umständlich in Diagrammen darstellen. Ganz ohne Werkzeug für das *Requirements Management*, um natürlichsprachliche Informationen festzuhalten, wird man bei seiner Systementwicklung doch nicht auskommen.

... und im Design weiterführen

Die Tätigkeiten im Design und bei der Architektur sind sehr umfangreich und vielfältig und haben direkte Auswirkungen auf die Qualität des entstehenden Systems. Dabei muss auf jeden Fall gewährleistet werden, dass die in der Analyse spezifizierten, funktionalen und nicht funktionalen Anforderungen erfüllt werden. Viele der Design- und Architekturtätigkeiten von geschäftsprozessorientierten Systemen entsprechen denen anderer (z. B. technischer) Systeme. Der größte Unterschied liegt in der Aufgabe der Prozessdefinitionserstellung bei geschäftsprozessorientierten Systemen, welche nachher beschrieben wird.

Das Vorgehen des Architekten beim Finden und Definieren von Komponenten unterscheidet sich bei geschäftsprozessorientierten Systemen nicht sonderlich von anderen Systemen. Der einzige wichtige Aspekt ist, dass wir in den Workflow-Management-Systemen eine Workflow-Engine haben, welche sich um die korrekte Abarbeitung der definierten Prozesse kümmert. Diese Workflow-Engine finden wir in Java-Applikationen als Enterprise-JavaBeans, den Komponenten, welche die Geschäftslogik enthalten, wieder. Daher wird hier nun speziell auf das Design der Prozesse eingegangen. Diese Prozesse wurden in der Analyse definiert und mit Hilfe verschiedener UML-Diagramme dokumentiert. Nun ist es an der Zeit, diese Prozesse in dem zu entwickelnden System zu realisieren. Daher ist eine wichtige Aufgabe, die Umwandlung der in der Analyse definierten Prozesse in Prozessdefinitionen. Eine Prozessdefinition ist eine vom Rechner lesbare Form der Prozessbeschreibung.

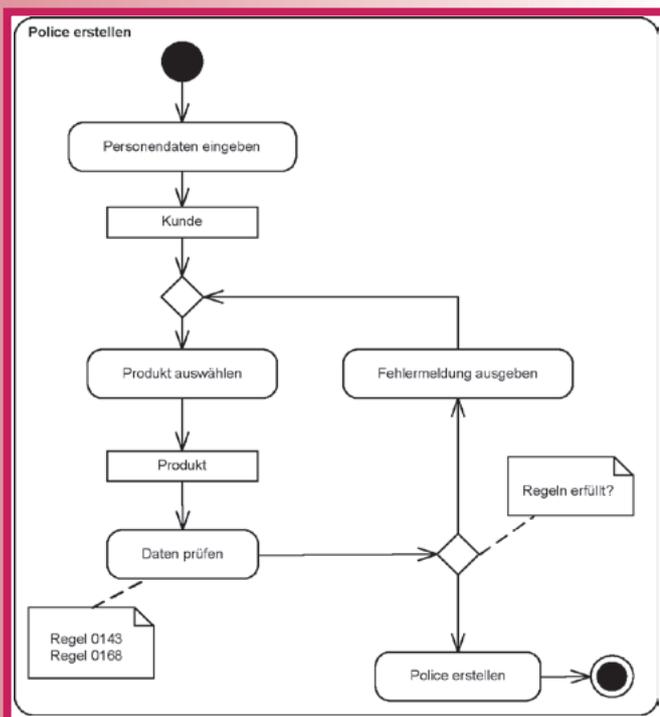


Abb. 2: Ein grober Ablauf mit Aktivitätsdiagramm inklusive Objektknoten und Referenzen auf Regeln

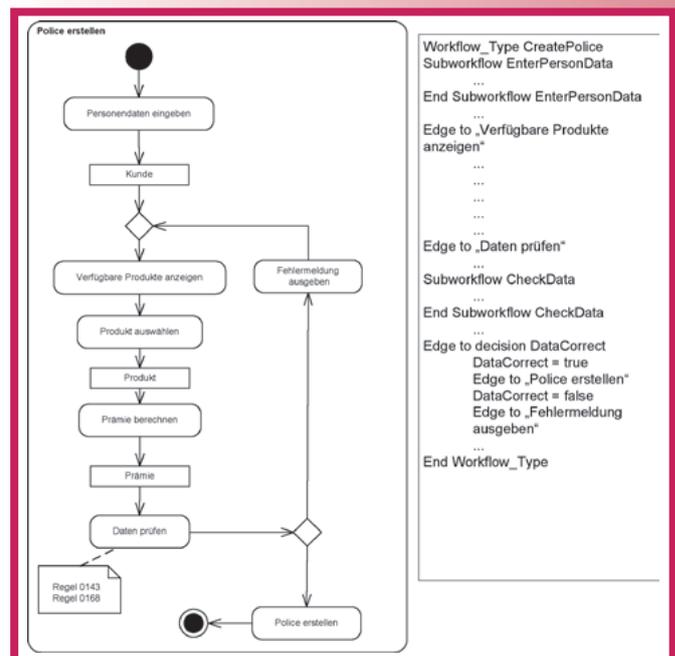


Abb. 3: Ableitung einer Prozessdefinition aus dem Aktivitätsdiagramm der Analyse. Der Ablauf wurde an einer Stelle beispielhaft verfeinert. Die Prozessdefinition spiegelt keine echte Prozessdefinitionssprache wider, sondern zeigt nur den schematischen Aufbau einer solchen Beschreibung



Für jeden Prozess wird eine Prozessdefinition erstellt. Für die Prozessdefinitionen gibt es eine Vielzahl an Prozessdefinitionssprachen. Dabei wird die Wahl der Sprache häufig von dem verwendeten Workflow-Management-System vorgegeben. Zwei Vertreter sind die *Business Process Execution Language* [BPEL] und die *XML Process Definition Language* [XPDL]. Die *Object Management Group* (OMG), die auch die UML-Standards herausgibt, arbeitet zur Zeit mit der *Business Process Management Initiative* [BPMI] zusammen, um eine standardisierte Prozessdefinitionssprache zu erstellen.

Das Vorgehen ist aber unabhängig von der Prozessdefinitionssprache. Auf den Punkt gebracht, ist das Prinzip der Prozessdefinition relativ einfach. Die Prozesse werden 1:1 aus den Aktivitätsdiagrammen in eine formalisierte und computerlesbare Form umgewandelt. Falls Sie Ihre Prozesse mittels Zustandsdiagrammen modelliert haben, können Sie das genauso machen. Voraussetzung für die Ableitung der Prozessdefinitionen sind natürlich sehr detaillierte Prozessbeschreibungen. Wie eine solche Prozessdefinition aussehen könnte, wird anhand des Versicherungsbeispiels in Abbildung 3 dargestellt.

Die Prozessbeschreibungen können dann in den Rechner eingelesen und von der Workflow-Engine des Workflow-Management-Systems abgearbeitet werden. Bei einer Java-Applikation werden die Prozesse durch Enterprise-JavaBeans (EJB) realisiert. Die Prozessdefinitionen werden in einer *RepositoryBean* abgelegt. Diese *RepositoryBean* beinhaltet neben den Prozessbeschreibungen auch alle Daten, die für die Prozesse benötigt werden. Für die Abarbeitung der einzelnen Aktionen in einem Prozess ist eine *WorkflowBean* verantwortlich, welche dafür die Daten der *RepositoryBean* verwendet und gegebenenfalls ändert.

Durch diese Vorgehensweise ist es dem Designer möglich, die in der Analyse definierten Prozesse vollständig im System abzubilden. Dadurch wird der wichtige Kern geschäftsprozessorientierter Systeme den Anforderungen entsprechend umgesetzt.

Fazit

Durch die Beschreibung geschäftsprozessorientierter Systeme mit der UML können Analyseergebnisse im Design weiterverwendet werden. Dadurch werden Transformationsverluste von der Analyse zum Design vermieden und die korrekte Umsetzung der Anforderungen in das Design unterstützt. Allerdings setzt der Weg über die Prozessbeschreibung eine sehr detaillierte Modellierung der Abläufe in der Analyse voraus, was in vielen Projekten im Konflikt zu dem engen Terminplan stehen kann. Wichtig bei dieser Vorgehensweise ist, dass sich das Analyse- und das Designteam auf einheitliche Modellierungskonventionen festlegen. Methodisch steht dann der Entwicklung geschäftsprozessorientierter Systeme nichts mehr im Wege.

Literatur und Links

[Böhm97] M. Böhm, Modellierung von Workflows, in: S. Jablonski, M. Böhm, W. Schulze (Hrsg): *Workflow-Management – Entwicklung von Anwendungen und Systemen, Facetten einer neuen Technologie*, dpunkt, 1997

[BPEL] Business Process Execution Language, [www-106-ibm.com/developersworks/webservices/Library/ws-bpel](http://www-106.ibm.com/developersworks/webservices/Library/ws-bpel)

[BPMI] Business Process Management Initiative, www.BPMI.org

[Rupp04] C. Rupp, *Requirements-Engineering und -Management*, Hanser, 2004

[Rupp05] C. Rupp, J. Hahn, S. Queins u.a., *UML 2 glasklar*, Hanser, 2005

[XPDL] XML Process Definition Language, www.simonstl.com/projekts/xpdl



Carsten Pflug ist Trainer und Berater der SOPHIST GmbH und hat sich in der Firma insbesondere auf die Objektorientierung mit der UML spezialisiert. Dabei vermittelt er in verschiedenen Projekten Know-how in den Fachgebieten der objektorientierten Analyse und dem Design.
E-Mail: carsten.pflug@sophist.de.