

Modellierung von Anforderungen

Dehla Sokenou
GEBIT Solutions
Koenigsallee 75b
14193 Berlin
www.gebit.de
dehla.sokenou (at) gebit.de

Abstract: In der betrieblichen Anwendungsentwicklung werden in vielen Fällen Anforderungen in rein textueller Form erfasst. Nachteile ergeben sich aus der mangelnden Auswertbarkeit dieser textuellen Beschreibungen. Ausserdem gibt es eine große Diskrepanz zwischen den textuellen Anforderungsdokumenten und den Designdiagrammen, die z.B. in UML erstellt werden. Vielfach wird versucht, Anforderungen stattdessen mit Hilfe eines UML-Werkzeugs zu erfassen. Doch diese Werkzeuge unterstützen textuelle Dokumente nur sehr unzureichend. Unser Ansatz überträgt modellgetriebene Ansätze auf das Requirements Engineering und verbindet beide Welten zu einer einheitlichen Sicht auf Anforderungen.

1 Motivation

Gängige Softwaresysteme zur Erfassung von Anforderungen werden überwiegend in internationalen Großunternehmen eingesetzt, bei denen die Erfassung von Anforderungen eng mit gesetzlichen Regelungen verknüpft ist oder es eine enge Zusammenarbeit zwischen Herstellern und Zulieferern gibt (z.B. Pharma-, Verkehrs- oder militärische Industrie).

Anders sieht es dagegen im Bereich der klassischen betrieblichen Softwareentwicklung aus. Hier sind diese Systeme eher selten anzutreffen. Stattdessen werden Anforderungen häufig mit mehr oder weniger strukturell vorgegebenen Word-Dokumenten erfasst, wobei die vorgegebene Struktur ohne Probleme vom Anwender verändert werden kann. Dies führt häufig dazu, dass die so entstandenen Dokumente nicht mehr einheitlich sind und von Außenstehenden nur schwer verstanden werden können. Strukturierungsmechanismen wie Use Case Templates lösen diese Probleme nur bedingt. Auch andere Aspekte sprechen gegen diese Praxis, z.B. dass eine Auswertung oder gar die automatische Ableitung von neuen Artefakten aus diesen Textdokumenten nur sehr schwer bis unmöglich sind.

Ein Fortschritt auf dem Gebiet ist die Verwendung von graphischen Modellierungssprachen (z.B. UML). Typischerweise werden Use Cases verwendet, an denen die Anforderungen in Form von textuellen Beschreibungen hinterlegt sind. Der Medienbruch zur Anwendungsentwicklung wird vermieden und alle Projektbeteiligten sprechen die gleiche Sprache. Auch Verknüpfungen zu anderen Diagrammen sind leicht möglich. Doch ein UML-Werkzeug ist nicht primär auf die Erfassung von Anforderungen ausgerichtet. So wird einerseits der Anwender auf fachlicher Seite nicht ausreichend geführt, anderer-

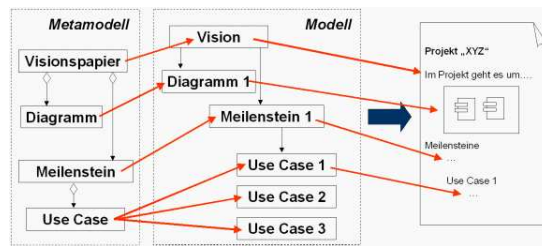


Abbildung 1: Kleiner Ausschnitt aus einem Requirements-Metamodell (links), Dokumentenmodell (Mitte) und Dokument (rechts)

seits können sich vermeintliche Kleinigkeiten wie die Sortierung der Ausgabedokumente zu einem großen Problem ausweiten. So gibt es z.B. bei den Use Cases in einem Diagramm keine ersichtliche Sortierung, die aber bei der Ausgabe in einem Pflichtenheft meist gewünscht ist. Eine Nummerierung der Use Cases als Abhilfe wird spätestens dann zum Problem, wenn ein neuer Use Case zwischen zwei bestehenden eingefügt werden muss.

Deshalb schlagen wir einen Ansatz vor, der beide Welten verbindet: modellgetriebenes Requirements Engineering (MDRE). Anforderungen werden in strukturierter Form erfasst. Die Struktur wird festgelegt durch ein Metamodell, das einmal angelegt wird und vom Endanwender in der Regel nicht mehr geändert werden kann. Die Einheitlichkeit der Dokumente ist damit sichergestellt.

Abschnitt 2 stellt MDRE im Detail vor. Ein Überblick über unsere Erfahrungen mit dem Ansatz wird in Abschnitt 3 gegeben. Ein Vergleich mit anderen Ansätzen findet sich in Abschnitt 4, eine Zusammenfassung in Abschnitt 5.

2 Modellgetriebenes Requirement Engineering

Betrachtet man modellgetriebene Ansätze, so gibt es einige Grundprinzipien, die sich auch auf das Requirements Engineering übertragen lassen:

1. Ein Metamodell gibt die verfügbaren Artefakte, deren Eigenschaften und Beziehungen zu anderen Artefakten vor.
2. Die Systembeschreibung stützt sich auf diese Artefakte und wird in einem Modell durchgeführt, dessen Struktur durch das Metamodell festgelegt ist.
3. Es wird versucht, die Beschreibung möglichst frei von Redundanzen vorzunehmen. Eine Information, die an mehreren Stellen benötigt wird, wird entweder aus dem Modell generiert oder aus anderen Teilen des Modells entnommen.
4. Änderungen am Modell haben direkte Auswirkung in der durch das Modell beschriebenen Anwendung. Dies lässt sich auf unterschiedliche Weise realisieren, z.B. durch Modelltransformationen, Generierung von Quellcode aus dem Modell oder durch unmittelbare Verwendung von Modellinformationen zur Laufzeit.

Übertragen auf das Requirements Engineering bedeutet dies:

1. Ein Metamodell gibt vor, welche Dokumente für die Anforderungen zu verwenden sind. Das können z.B. Glossar, Projektübersicht, Use Case Paket, Use Case Spezifikation, Testfall, Testplan, Change Request usw. sein. Alle Artefakte sind referenzierbar, so dass Verweise auf andere Artefakte, z.B. UML-Diagramme, können definiert werden, z.B. dass eine Use Case Spezifikation einen Verweis auf ein Use Case Diagramm, aber auch einen einzelnen Use Case oder Akteur, definiert werden können. Abbildung 1 zeigt als Beispiel auf der linken Seite einen Ausschnitt aus einem Metamodell des TREND/Analyst¹. Das Metamodell ist wie das UML-Metamodell in UML spezifiziert und kann in generischer Weise für jedes Projekt individuell angepasst werden. Dies kann durch komplette Neudefinition oder auf Basis eines der vordefinierten Metamodelle durch Kopieren oder Vererbung realisiert werden.
2. Anforderungen werden strukturiert gemäß dem Metamodell erfasst. Für jedes Artefakt gibt es eine Vorlage oder eine Eingabemaske, die durch den Typ und die Eigenschaften des Artefakts im Metamodell vorgegeben ist. Dadurch sind Aussehen, Verknüpfung mit anderen Dokumenten und Diagrammen und weitere Eigenschaften fest vorgegeben und für alle Projektbeteiligten gleich. In Abbildung 1 ist dies in der Mitte und auf der rechten Seite dargestellt.
3. Durch Verknüpfungen zu anderen Artefakten kann das Anforderungsmodell frei von Redundanzen gehalten werden.
4. Für den weiteren Softwareentwicklungsprozess lassen sich die Artefakte aus dem Anforderungsmodell weiterverwenden oder neue Artefakte (z.B. Templates für den Code oder für Testfälle) generieren.

MDRE heißt nicht, auf textuelle Beschreibungen zu verzichten und Anforderungen quasi zu "malen". Anforderungen werden immer noch zu großen Teilen in textueller Form erfasst, aber strukturiert und zusätzlich mit (UML-) Diagrammen verknüpft.

Das Metamodell bestimmt nicht nur die Struktur der Dokumente und Diagramme, sondern zusätzlich die Formate für den Export, steuert über Beziehungen zwischen Dokumenten und Diagrammen die Benutzerführung und gibt die Struktur für die Benutzerhilfe vor, in dem für jeden Dokumenttyp das Template einer Hilfedatei für den Endanwender generiert wird. Änderungen am Exportformat können durch Änderungen des Exports im Metamodell leicht im laufenden Projekt vorgenommen werden, ohne dass die Anforderungsartefakte selbst angefasst werden müssen, wie es bei der üblichen Anforderungserfassung der Fall wäre. Zudem unterstützt das Metamodell die korrekte Eingabe durch definierte Felder, die durch den Typ einer Eigenschaft im Metamodell (z.B. Datum) vorgegeben werden. Links zwischen Artefakten sind nur möglich, wenn im Metamodell entsprechende Referenzen definiert sind. Durch die Markierung von Eigenschaften als obligatorisch wird

¹TREND/Analyst ist ein Werkzeug zur modellbasierten Erfassung von Anforderungen, das von GEBIT entwickelt wurde und in einer kostenlosen Community Edition sowie in einer kostenpflichtigen Enterprise Edition erhältlich ist, die zusätzlich zur Community Edition die Re- und Neudefinition des Metamodells erlaubt. Das Standardmetamodell umfasst 32 Artefakte und 64 Links.

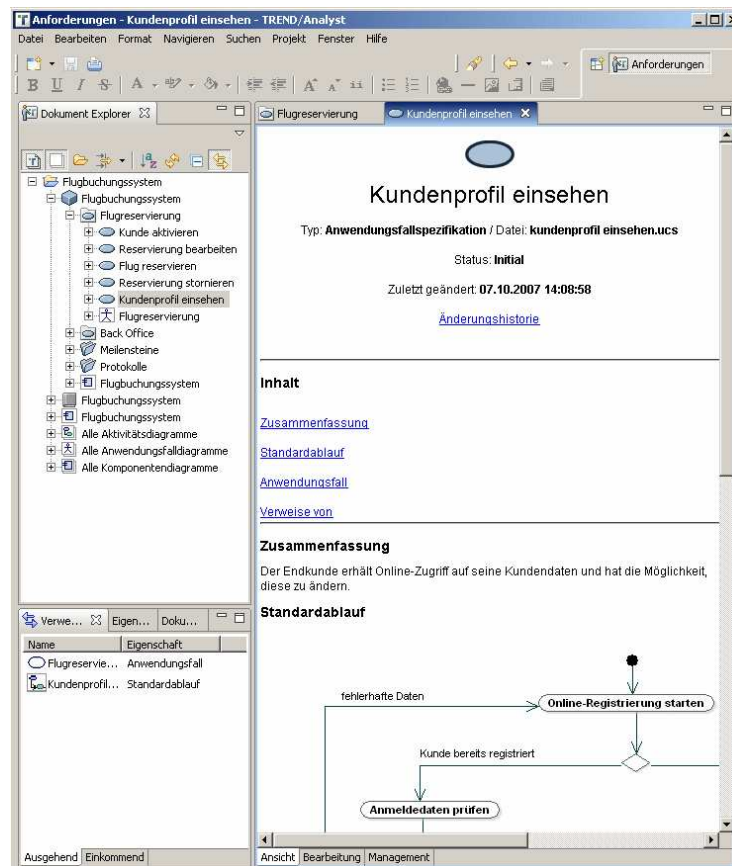


Abbildung 2: Beispiel für ein Anforderungsdokument (Use Case Spezifikation)

geprüft, ob ein Dokument bereits vollständig ist, der Benutzer bekommt einen entsprechenden Hinweis. Ein Dokument ist erst dann vollständig, wenn alle obligatorischen Eigenschaften mit Werten versehen wurden. Auch die referenzielle Integrität wird überprüft. Sollen z.B. noch verwendete Artefakte gelöscht werden, wird der Benutzer gewarnt.

Änderungen am Metamodell für Dokumente und Diagramme sind in begrenztem Umfang möglich, z.B. ist das Hinzufügen einer Eigenschaft oder einer Verknüpfung problemlos möglich. Das Löschen einer Eigenschaft ist problematischer, da Information verloren geht. Änderungen von Eigenschaft sind aktuell noch nicht in geeigneter Weise möglich.

Einzelne Eigenschaften von Dokumenten und Diagrammen können für die Auswertung herangezogen werden, z.B. der Status einzelner Use Cases für eine Auswertung des Projektstatus oder die Zuordnung von Use Cases zu Meilensteinen.

Ein Beispiel für ein Anforderungsdokument, das mit unserem Werkzeug für modellgetriebenes Requirements Engineering erstellt wurde, zeigt Abbildung 2. Die Zusammen-

fassung ist eine im Metamodell definierte textuelle Eigenschaft des Dokumenttyps Use Case Spezifikation, die Eigenschaft Standardablauf definiert eine Verknüpfung zu einem UML-Diagramm vom Typ Aktivität.

3 Erfahrungen

Unsere Erfahrungen mit modellgetriebenem Requirements Engineering und der Anwendung des Werkzeugs haben gezeigt, dass dieser Ansatz nicht nur die strukturierte Anforderungserfassung unterstützt und zu einheitlichen und vollständigen Anforderungsdokumenten führt, sondern dass viele Erweiterungen möglich sind.

Um einige Beispiele zu nennen:

- Ein Bericht über Projektstatus kann "on demand" durch Generierung des entsprechenden Exportdokuments erzeugt werden.
- Die Integration mit einem Issue Tracking System ermöglicht das Tracking des Projektstatus auch in der Implementierungsphase.
- Durch Integration eines gemeinsamen Repositories können Business Analysten und Softwareentwickler auf denselben Dokumenten arbeiten. Änderungen an den Anforderungen werden so schneller kommuniziert. Die gemeinsame Arbeit auf einer gemeinsamen Basis ist besonders bei schnell ändernden Anforderungen wie in agilen Projekten von Vorteil.
- Teile der Anforderungen lassen sich direkt in das Design und bei modellgetriebener Entwicklung auch in die Implementierung übernehmen. So lassen sich z.B. die ersten Entwürfe für Klassendiagramme zu Implementierungsklassen verfeinern und Use Cases als Strukturierungseinheiten auch in der Implementierung verwenden.

Die generische Anpassbarkeit der Struktur der Anforderungsdokumente und Exportformate durch ein Metamodell macht den Einsatz des Werkzeugs sehr universell. Das Werkzeug wurde bereits erfolgreich in Kundenprojekten eingesetzt. Dabei stand zunächst meist der Entwurf eines Metamodells im Vordergrund, das auf die eigenen Bedürfnisse passt. Dies wurde meist in Zusammenarbeit von Business Analysten und Softwareentwicklern realisiert. Nach der Definition sind in der Regel nur noch wenige Anpassungen am Metamodell nötig, dieses wird dann im Projekt oder auch unternehmensweit verwendet. Bisher ist kein Fall aufgetreten, in dem sich eine Struktur nicht mit Hilfe des Metamodells abbilden ließ, so kann z.B. das Aussehen des Exports an das bisher verwendete Format angepasst werden, so dass die Anwendung von modellgetriebener Anforderungserfassung zu keinem sichtbaren Bruch zur bisherigen Anforderungserfassung führt. Die Projektgröße variierte von 1 PM bis 70 PJ. Alle Projekte wurden "in time & budget" abgeschlossen und die Projektdokumentation war auch am Ende des Projekts noch aktuell.

Ein Nachteil des Ansatzes ist, dass Anforderungen nicht mehr unstrukturiert erfasst und erst später geordnet werden können. Dies bedeutet für manche Anwender eine große Umstellung der bisherigen Arbeitsweise und kann die Akzeptanz verringern. Unsere Praxiserfahrung zeigt jedoch, dass die Vorteile des Ansatzes diesen Nachteil überwiegen.

4 Verwandte Ansätze

Verschiedene andere Ansätze ermöglichen ebenfalls die Integration von Requirements und (UML-) Modellen. Zum einen gibt es Erweiterungen zu klassischen Requirements-Werkzeugen wie DOORS/Analyst, die die Modellierung und Referenzierung von Diagrammen unterstützen. Im Gegensatz zu unserem Ansatz ist jedoch in der Regel nur ein Typ von Artefakt (eventuell mit geerbten Typen) vorgesehen, so dass z.B. Eigenschaften in jedem Requirement vorhanden sind und nicht flexibel einzelnen Typen von Artefakten zugewiesen werden können. Links zwischen Requirements sind meist nicht einer Eigenschaft zugeordnet und damit nicht benannt. Die Struktur ergibt sich nicht durch ein strukturiertes Metamodell wie in TREND/Analyst, sondern durch die Hierarchie der Requirements.

Im Gegensatz dazu vereinfachen Werkzeuge wie Caliber Analyst die Modellierung von UML-Diagrammen für Anforderungsermittler durch die werkzeugunterstützte textuelle Eingabe von Diagrammelementen. Das dahinterliegende Metamodell ist nicht flexibel wie in TREND/Analyst.

Diese Werkzeuge werden meist in Unternehmen eingesetzt, in denen die Erfassung von Anforderungen eng mit gesetzlichen Regelungen verbunden ist, z.B. der Automobilbranche. Auch neue Ansätze der modellbasierte Anforderungsanalyse, z.B. AutoRAID oder IMMOS, finden sich verbreitet in diesen Branchen. Diese lassen sich in der Regel jedoch nicht ohne weiteres auf die betriebliche Anwendungsentwicklung übertragen, da hier die Struktur der Dokumente zwischen einzelnen Unternehmen stark variiert. Diese Ansätze setzen eine feste oder nur begrenzt anpassbare Struktur der Anforderungen voraus, während die Struktur in TREND/Analyst generisch definiert werden kann.

5 Zusammenfassung und Ausblick

Mit dem modellbasierten Requirements Engineering werden Prinzipien der modellbasierten Entwicklung konsequent auf die Anforderungsermittlung ausgedehnt. Vorteil ist die homogene Werkzeuglandschaft, die von Anforderungsermittlern und Entwicklern gemeinsam benutzt wird und so die Verfolgbarkeit von Anforderungen im gesamten Prozess ermöglicht. Durch die Generierung von Dokumenten wie Pflichtenheft oder Projektfortschrittsdokument sind diese immer auf dem aktuellen Stand. Bei Änderung des Formats dieser generierten Dokumente oder in begrenzter Form auch der Struktur der Anforderungen müssen diese nicht mehr manuell angepasst werden, sondern die Anpassung erfolgt nur im Metamodell bzw. in der Beschreibung der Transformation.

Viele Nachteile der bisherigen Erfassung von Anforderungen werden durch die modellgetriebene Technik beseitigt, was nach unseren Erfahrungen zu der großen Akzeptanz unter den Anwendern führt. Dazu trägt auch bei, dass bewährte Techniken wie die starke Konzentration auf textuelle Bestandteile und die ursprüngliche Dokumentstruktur bewahrt werden. Ein solcher Ansatz kann die Qualität der Anforderungen gerade in Bereichen verbessern, in denen Anforderungen bisher informell erfasst wurden.