

Modellbasierte Anforderungsentwicklung mit AutoRAID

Bernhard Schätz, Andreas Fleischmann, Eva Geisberger, Markus Pister

Fakultät für Informatik, Technische Universität München
Boltzmannstr. 3, 85748 Garching
{fleischa|geisberg|pister|schaetz}@in.tum.de

Abstract: Während im Softwareentwurf zunehmend modellbasierte Ansätze zur Anwendung kommen, wird die Anforderungsentwicklung meist unter Verwendung von strukturiertem Text durchgeführt; dies hat zur Folge, dass nur eine schwache Verbindung zwischen Anforderungsanalyse und Entwurf möglich ist. Modellbasierte Anforderungsentwicklung hilft, diese Lücke zu schließen sowie die Qualität von Anforderungsspezifikationen zu verbessern. Der Werkzeugprototyp AutoRAID zeigt, wie ein strukturiertes und integriertes Anforderungsmodell die modellbasierte Anforderungsentwicklung unterstützen kann, und Operationen für die Verfeinerung und Strukturierung von textuellen Anforderungen hin zum Entwurfsmodell angeboten werden können.

1 Einleitung

Wesentliche Voraussetzung für die Unterstützung einer modellbasierten Anforderungsentwicklung ist einerseits die Einführung eines gemeinsamen konzeptuellen Modells für Anforderungen und Entwurfs-elemente, und andererseits der Einsatz dieses konzeptuellen Modells zur schrittweisen Strukturierung und Formalisierung im Laufe des Entwicklungsprozesses. So wird ein schrittweiser, werkzeuggestützter Übergang von der Anforderungsanalyse hin zum Entwurf ermöglicht, der den Bruch zwischen diesen Phasen verringert. Im Folgenden demonstrieren wir anhand des Werkzeugprototypen AutoRAID eine mögliche Unterstützung für diesen Übergang, unter Nutzung einer reviewbasierten Vorgehensweise.

Aufgrund ihrer eher informellen Natur von (textuellen) Anforderungen lassen sich viele Techniken der Qualitätssicherung aus dem Entwurf nicht oder nur schwer für die Qualitätssicherung in der Anforderungsentwicklung einsetzen. Standards wie IEEE 1233-1998 [IE98a] führen daher spezifische Anforderungen an qualitätssichernde Maßnahmen für die Anforderungsentwicklung ein. Im Folgenden zeigen wir, wie mittels des modellbasierten Ansatzes von AutoRAID in [IE98a] adressierte Qualitätsdefizite mittels konstruktiver und analytischer Verfahren angegangen werden können:

- **Inkonsistenz:** Durch das Aufbrechen von unstrukturierten Anforderungen in ein detailliertes Anforderungsmodell mit Verbindungen zwischen den Modellteilen können Inkonsistenzen frühzeitig erkannt werden (z.B. bei der Untergliederung eines Systems in einzelne Interaktionsschritte unter Berücksichtigung der Schnittstelle des Systems).

- **Mehrdeutigkeit:** Durch die Verwendung eines detaillierten konzeptuellen Modells für die Anforderungen können die informellen Anforderungen auf Elemente des Entwurfsmodell abgebildet und so Mehrdeutigkeiten schrittweise eliminiert werden (z.B. im Zuge der Klassifizierung und Detaillierung von Einzelschritten eines Beispielablaufs).
- **Nichtverfolgbarkeit:** Durch die werkzeuggestützte und teilweise automatisierte Vernetzung von Anforderungen mit den Elementen des Entwurfs wird eine werkzeuggestützte Rückverfolgbarkeit sichergestellt (z.B. durch automatische Vernetzung im Zuge der Motivation eines Betriebsmodus aus einer Modusanforderungen heraus).
- **Realisierbarkeit, Verifizierbarkeit:** Durch die Einschränkungen, die durch die Formalisierung über das konzeptuelle Modell bei der Beschreibung von Anforderungen erzwungen werden, wird eine Präzisierung ähnlich wie bei der Verwendung modellbasierter Beschreibungsformen im Entwurf sichergestellt und so die Realisierbarkeit und Verifizierbarkeit von Anforderungen verbessert (z.B. bei der Präzisierung von Interaktionen in einem Beispielablauf).

Gängige Werkzeuge zur Unterstützung der Entwicklung von Anforderungen bieten aktuell nur geringe Möglichkeiten zur Qualitätssicherung; die wichtigsten Techniken – meist ohne spezifische Werkzeugunterstützung – sind die Verwendung von Formularen sowie Reviews. Im Gegensatz dazu verwendet der modellbasierte Ansatz zur Entwicklung von Anforderungen ein formalisiertes Modell von klassifizierten und präzisierten Anforderungen, wie sie implizit in Ansätzen wie [IE98b] zum Einsatz kommen. So fügt sich dieser modellbasierte Ansatz nahtlos in einen strukturierten Anforderungsprozess ein: Funktionalitäten wie die Strukturierung oder Klassifizierung von Anforderungen, die in reviewbasierten Verfahren wie in [DP04] zum Einsatz kommen, werden werkzeuggestützt den Domänenexperten angeboten. Neben diesen konstruktiven Entwicklungsschritten (z.B. die Identifikation von Einzelschritten in einem Beispielablauf) werden typischerweise in Inspektionen eingesetzte analytische Schritte mechanisiert durchgeführt (z.B. die Absicherung der vollständigen Zuordnung von funktionellen Anforderungen zu Systemkomponenten). In den folgenden Abschnitten werden die in AutoRAID [AR04] realisierten Aspekte vorgestellt:

- *Strukturiertes Anforderungsmodell:* Einführung von domänenspezifischen Anforderungsklassen (z.B. architektonische und modale Anforderungen, Anwendungsfallszenarien)
- *Integration textueller Anforderungen:* Einbettung textueller Anforderungen in das konzeptuelle Modell und pragmatische Verfahren zu deren Nutzung im Entwicklungsprozess (z.B. Import von Anforderungsdokumenten)
- *Werkzeuggestützte Strukturierung und Klassifikation:* Automatisierung der schrittweisen Entwicklung (z.B. Identifikation von Akteuren in funktionalen Anforderungen)

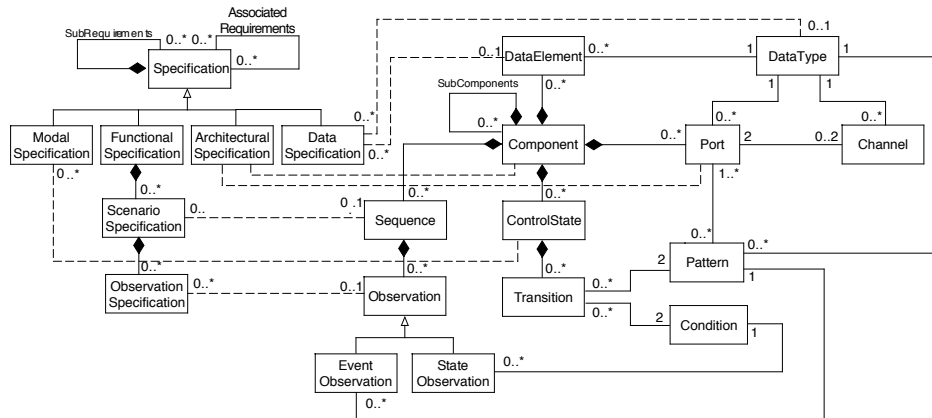


Abbildung 1: Vereinfachtes konzeptuelles Modell für Analyse und Entwurf

- *Kopplung von Anforderungen und Elementen des modellbasierten Entwurfs*: Verbindung der Anforderungselemente mit den Entwurfselementen (z.B. Kopplung von modalen Anforderungen mit Kontrollzuständen)

Als konzeptuelles Modell für den Entwurf wird hier das Modell von AutoFOCUS [Sch04] verwendet, das spezifisch auf die Domäne reaktiver Systeme ausgerichtet ist.

2 Konzeptuelles Modell von AutoRAID

Werkzeuge für die Anforderungsanalyse, die aktuell in der praktischen Anwendung zum Einsatz kommen, verwenden im Allgemeinen vergleichsweise einfache konzeptuelle Modelle zur Darstellung von Anforderungsspezifikationen. Im wesentlichen werden in diesen Modellen lediglich die Konzepte „Anforderung“ sowie die Relationen „Unteranforderung“ sowie „Assoziierte Anforderung“ eingesetzt. Im Gegensatz dazu – wie Abbildung 1 gezeigt – benutzt AutoRAID ein deutlich komplexeres Modell, mit dem Konzept des (informellen) Spezifikationselements als zentrales Konzept. Wie in Abschnitt 3 im Detail erläutert, besteht der im Folgenden beschriebene Ansatz aus den Aktivitäten *Strukturierung*, *Klassifikation*, *Formalisierung* und *Analyse*. Um diese Aktivitäten zu unterstützen, bietet das konzeptuelle Modell von AutoRAID explizite Elemente und Assoziationen.

2.1 Strukturierung der Anforderungen

Ein der zentralen Schritte in der Anforderungserstellung – wie z.B. durch Werkzeuge DOORS oder Requisite Pro unterstützt – besteht im der Strukturierung von Anforderungen in Unteranforderungen. AutoRAID nutzt dazu das Konzept *Specification* (entsprechend einer Anforderung) und sowie die Assoziationen *SubRequirements* und *AssociatedRequirements*. Dabei haben die Spezifikationselemente die üblichen Attribute (z.B. eindeutige Identifikatoren, Verantwortlicher, Priorität). Anforderungen können unter Verwendung der *SubRequirements*-Assoziation sowie der *AssociatedRequirements*-Assoziation in Unteranforderungen untergliedert und in eine Hierarchie eingeordnet

werden, um die Rückverfolgung von Anforderungen zu ihren übergeordneten Zielen zu ermöglichen.

2.2 Klassifizierung der Anforderungen

Klassifizierung von Anforderungen bildet die Grundlage ihrer Formalisierung. Wie beispielsweise in [IE98b] vorgeschlagen, können Anforderungsspezifikationen klassifiziert werden als:

- *Architektonische Anforderungen*, die den Aufbau eines Systems beschreiben; Komponenten, Schnittstellen sowie Kanäle werden beispielsweise mithilfe dieser Anforderungen erfasst.
- *Modale Anforderungen*, die die Betriebsmodi des spezifizierten Systems beschreiben; verschiedene Kontrollzustände oder Modi des kompletten Systems oder einer Unterkomponenten werden durch diese Anforderungen abgedeckt.
- *Datentypanforderungen*, die jene Datentypen beschreiben, die zur Deklaration von Nachrichten oder zur Deklaration von Datenzuständen genutzt werden.
- *Funktionale Anforderungen*, die mittels Anwendungsfällen in Form von repräsentativen Beobachtungssequenzen über das System und seine Interaktionen mit der Umgebung beschrieben werden.

2.3 Integrierte Anforderungen

Eine wesentliche Funktionalität von AutoRAID ist die Unterstützung eines benutzerfreundlichen Übergangs von der informellen zur modellbasierten Beschreibung des Systems; entsprechend bietet das konzeptuelle Modell – wie in Abbildung 1 dargestellt – eine enge Kopplung zwischen den Konzepten aus der informellen Anforderungsanalyse auf der einen Seite, sowie dem modellbasierten Entwurf auf der anderen Seite. Daher ist jedes klassifizierte informelle Konzept (wie z.B., *architectural specification*) zu einem entsprechenden Konzept des Entwurfs (z.B. *component, channel, port*) assoziiert. Die gestrichelten Assoziationen stellen dabei die Verbindungen zwischen den konzeptuellen Elementen der beiden Modellbereiche dar. Diese Verbindungen von den Anforderungen zu den Entwurfselementen werden dabei sowohl zur Unterstützung der Analyse der Spezifikation und ihrer Vervollständigung als auch zur Abhängigkeitsverfolgung zwischen Analyse und Entwurf verwendet.

3 Werkzeugunterstützung

Abbildung 2 zeigt die Benutzerschnittstelle von AutoRAID. Der Modellbrowser (auf der linken Seite der Abbildung) stellt die Elemente der Anforderungsanalyse (*Analysis*) sowie des Entwurfs (*Component Car*) jeweils baumartig dar. Im Anforderungsbaum werden sowohl generische *Specifications (Requirements)* als auch klassifizierte *Specifications (Use Cases, Constraints)* dargestellt. Im Arbeitsbereich (rechte Seite der Abbildung) ist der Formalisierungsschritt (siehe Abschnitt 3.3) für ein Ereignis (*Event Observation*; Fenster in der vordersten Ebene) dargestellt; dieses Kommunikationsereignis ist wiederum Teil einer Szenariospezifikation (*Scenario Specification*; Fenster in der middle-

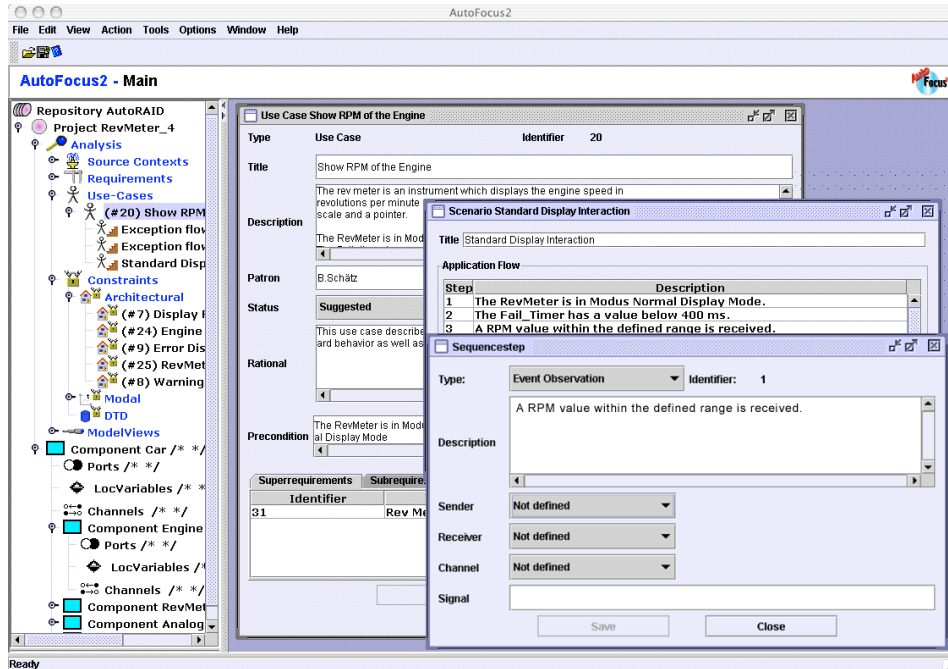


Abbildung 2: Definition eines Ereignisses im Szenario “Standard Display Interaction”

ren Ebene), das wiederum Teil einer funktionalen Spezifikation (*Functional Specification*; Fenster in der hinteren Ebene) ist. Die Anforderungsbearbeitung mittels AutoRAID basiert auf einem review-orientiert Entwicklungsprozess, dessen einzelne Schritte im folgenden beschrieben werden.

3.1 Erstellung und Import von Anforderungen

Zur Erzeugung von Anforderungen bietet AutoRAID die Möglichkeit der manuellen Erzeugung oder den Import von Anforderungsdokumenten. Bei der Erzeugung werden Anforderungen werden vom Benutzer zusätzlich zur Beschreibung Attribute wie *Titel*, *Status*, oder *Priorität* definiert; zusätzlich werden Anforderungen mit eindeutigen Bezeichnern versehen. Um Anforderungen durch Import aus einem vorhandenen Anforderungsdokument zu erstellen, bietet AutoRAID eine komfortable ‘select-and-create’-Funktionalität zur schnellen Identifikation und Isolation von Anforderungen. Um die Rückverfolgbarkeit sicherzustellen, werden Verbindungen zwischen dem importierten Dokumententext und den erstellten Anforderungen im Quelltext dargestellt. Zum Zweck der Dokumentationserstellung erlaubt AutoRAID die Erzeugung von strukturierten Dokumenten aus den bearbeiteten Anforderungen.

3.2 Strukturierung von Anforderungen

Um eine hierarchische Strukturierung von Anforderungen zu erarbeiten, können Anforderungen hierarchisch untergliedert werden. AutoRAID bietet dazu die folgenden Möglichkeiten:

- Ein *Specification*-Element kann direkt mit einer oder mehreren anderen *Specification*-Elementen verbunden werden, um Assoziationen zu untergeordneten Anforderungen (*SubRequirements*) oder abhängigen Anforderungen (*associated Requirements*) darzustellen.
- Aus einem bereits existierenden *Specification*-Element kann ein neues *Specification*-Element abgeleitet werden und mit diesem automatisch per *SubRequirement*-Assoziation verbunden werden; diese Werkzeugfunktionalität dient – wegen der Häufigkeit dieses Schrittes im Entwicklungsprozess – der komfortablen Untergliederung von zusammengesetzten Anforderungen in Unteranforderungen.

3.3 Klassifizierung und Formalisierung von Anforderungen

Der Klassifikationsschritt dient dazu, generische Anforderungen entsprechend ihrer Kategorie einzuordnen. Im Entwicklungsprozess werden dazu Anforderungen soweit in Unteranforderungen aufgebrochen, bis diese ausreichend strukturiert sind, um in eine der jeweiligen Kategorien (*architectural, modal, data, functional*) eingeordnet zu werden. Im Zuge der Klassifizierung werden die kategorisierten Anforderungen weiter präzisiert. So werden z.B. funktionale Anforderungen (*Functional Specification*) durch die Beschreibung in Form eines Beispielablaufs (*Scenario*) detailliert; dieser wiederum wird durch die Strukturierung in einzelnen Ablaufschritte (*Observation Specification*) weiter formalisiert. Ein einzelner Schritt wird wieder schließlich klassifiziert, z.B., in Form eines Kommunikationsereignisses (*Event Observation*) und durch Angabe von Sender, Empfänger und übermitteltem Signal präzisiert.

Obwohl hier formularartige und textbasierte Beschreibungstechniken zum Einsatz kommen, liegt die Anforderungsspezifikation auf dieser Ebene bereits in einer formalen Form vor, die durch die Präzision und die einschränkenden Anforderungen des detaillierten konzeptuellen Modells bestimmt wird. So ist z.B. die Definition eines Kommunikationsereignisses über seine Sender- und Empfängerattribute an Komponenten gebunden, die aus entsprechenden architektonischen Anforderungen abgeleitet wurden.

Daher können dieser Ebene die Spezifikationselemente bereits äquivalent mittels graphischer Beschreibungsmittel dargestellt werden; beispielsweise wird ein präzisierter Beispielablauf alternativ durch ein (automatisch generiertes) Sequenzdiagramm dargestellt. Durch diese unterschiedlichen Sichtweisen ermöglicht es AutoRAID, auf der einen Seite eine detaillierte Beschreibung des Systems aus der Sicht der Anforderungsanalyse zu erstellen und entsprechende Informationen wie Begründungen von Entwicklungsentscheidungen zu dokumentieren; auf der anderen Seite kann diese detaillierte Information auch aus dem Blickwinkel des Entwurfs dargestellt werden, mit dem Schwerpunkt auf einer kompakten und übersichtlichen Darstellung der Information. Schließlich können durch die Verwendung eines gemeinsamen konzeptuellen Modells Elemente des Ent-

wurfs unmittelbar zu den zugrunde liegenden Elementen des Anforderungsmodells rückverfolgt werden.

3.4 Analyse der Anforderungen

Wie im vorherigen Abschnitt beschrieben, unterstützt AutoRAID die Analyse von Anforderungen bereits durch den Einsatz des konzeptuellen Modells gekoppelt mit einer reviewbasierten Vorgehensweise: über die vorgegebene detaillierte Struktur des konzeptuellen Modells wird so konstruktiv eine Präzisierung erzwungen und so eine implizite Analyse der Anforderungen bewirkt. Darüber hinaus ermöglicht aber die modellbasierte Strukturierung und Klassifizierung von Anforderungen auch den Einsatz von analytischen Verfahren in Form von Konsistenzanforderungen; diese können eingesetzt werden, um mögliche Schwächen des Modells zu identifizieren. Typische Bedingungen für diese Anforderungen sind:

- Jedes übergeordnete Ziel (*business requirement*) wird durch mindestens eine fachliche Anforderung (*application requirement*) präzisiert.
- Jede fachliche Anforderungen ist entweder durch eine weitere fachliche Anforderung präzisiert oder einer Anforderungsklasse zugeordnet.

AutoRAID erlaubt die Definition solcher Konsistenzbedingungen sowie deren automatische Überprüfung; bei verletzten Bedingungen werden die verletzenden Anforderungs- oder Entwurfs-elemente zur weiteren Bearbeitung herausgefiltert.

4 Zusammenfassung

AutoRAID bietet Schritte zur Strukturierung, Klassifikation und Analyse von Anforderungen; es unterstützt den Entscheidungsprozess durch Vorwärts- und Rückwärtsverketzung zwischen Anforderungen und Entwurf; es ermöglicht einen schrittweisen Übergang von informellen Spezifikationen hin zu einem modellbasierten Design. Aktuelle Arbeiten im Rahmen von AutoRAID untersuchen die Einbindung anderer domänenspezifischer Anforderungsklassen (z.B. Workflows) und die Unterstützung von Produktlinien. Die Praxistauglichkeit des Ansatzes wird aktuell in industriellen Fallstudien im Automobilumfeld (Karosserieelektronik) und im Luftfahrtbereich (Flugsteuerung) untersucht.

5 Literatur

- [AR04] Website of AutoRAID, <http://wwwwbroy.in.tum.de/~autoraidd/>
- [DP04] Denger, C. Paech, B. "An Integrated Quality Assurance Approach for Use Case Based Requirements". In: Rumpe, B. Hesse, W. (eds). Modellierung 2004. Springer, 2004.
- [IE98a] Software Engineering Committee of the IEEE Computer Society. IEEE Guide for Developing System Requirements Specifications. IEEE Standard 1233-1998. IEEE CS.
- [IE98b] Software Engineering Committee of the IEEE Computer Society. IEEE Guide for Software Requirements Specifications (ANSI). IEEE Standard 830-1998. IEEE CS
- [Sch04] Schätz, B.: Mastering the Complexity of Embedded Systems. In: Formal Techniques for Embedded Distributed Systems, Kordon F., Lemoine, M. (eds.). Kluwer, 2004.