

Sichtenintegrierte Anforderungsanalyse mit AutoRAID

Eva Geisberger, Bernhard Schätz

Institut für Informatik
Technische Universität München
Boltzmannstr. 3
D- 85748 Garching

{geisberg|schaetz}@in.tum.de

Zusammenfassung: Wesentliche Aufgabe der Gestaltung eingebetteter Systeme ist die systematische Erarbeitung funktionaler Systemanforderungen und deren entsprechende Integration in die Umgebung des technischen Gesamtsystems. Herausforderung hierbei sind die vielfältigen Abstimmungsaufgaben für eine der Problemstellung angemessenen Systemspezifikation. Dazu definiert die hier vorgestellte modellbasierte Anforderungsanalyse und Systemdefinition mit dem RM-Werkzeug AutoRAID einen strukturierten Modellierungsansatz, der systematisch die zielorientierte Erarbeitung und Abstimmung der verschiedenen Anforderungen mithilfe grundlegender System-sichten zu einer integrierten Spezifikation.

Schlüsselwörter: Anforderungsanalyse • Systemsichten • Systemmodell • Werkzeugunterstützung,

Abstract: When developing embedded systems, it is essential to systematically define functional requirements in agreement with the restrictions imposed by the technical aspects of the overall system. A main challenge is here to obtain an adequate definition of the system by coordinating different views, ranging from customer-oriented goals and requirements constraints of the technical design. According to that challenges we introduce our model based requirements analysis and system design approach and its supporting RM tool AutoRAID. Core concept is a structured modeling approach that systematically supports the stepwise, goal oriented refinement and integration of the various requirements with the help of various system views.

Keywords: Requirements Engineering • Views • Model Based • Tool Support

CR Subject Classification: D.2.1, D.2.2

1 Einführung

Eingebettete Systeme sind SW-/HW-Systeme die das Verhalten technischer Gesamtsysteme steuern. Beispiele sind Motor- oder Airbag-Steuerungen aus dem Automobilbereich oder Fertigungssysteme

in der Automatisierungstechnik. Die Herausforderungen für ihre Gestaltung werden bei der Betrachtung des Beispiels heutiger Systementwicklungen im Automobil deutlich:

- Eingebettete Systeme sind Teil komplexer verteilter System- und Produktlandschaften. Sie sind in HW- und SW-Systeme zu integrieren und bei der Definition der Systemschnittstellen unterschiedliche, oft standardisierte Domänenmodelle zu berücksichtigen.
- Sie haben hohe Qualitätsanforderungen bzgl. Funktionalität, Sicherheit, Echtzeit und Nutzungstransparenz – d.h. das System muss für den Nutzer vorhersehbar und verständlich reagieren.
- Eingebettete Systeme werden zunehmend im Rahmen von Produktlinien entwickelt. D.h. sie sind Weiter- oder Parallelentwicklung; für ihre angemessene Gestaltung sind produktstrategische Fragestellungen zu berücksichtigen ebenso wie die Strukturierung der Systemfunktionalität in Hinblick auf Plattformen und Variationen.

Unter diesen Gesichtspunkten wird deutlich, dass die Erarbeitung einer Anforderungs- und System-spezifikation das Ergebnis systematischer Abstimmungsaufgaben zwischen den verschiedenen Stakeholderanforderungen aus den Kunden-/Nutzergruppen (Nutzer, Marketing, Vertrieb, Service, Produktlinie) und den technischen Disziplinen (Mechanik, Elektronik, Informatik) sein muss. Dazu sind

- die Problemstellung und Zielsetzung der Produktentwicklung zu analysieren,
- die funktionalen Anforderungen und Eigenschaften sowie die Einbettung des Systems mit allen seinen Schnittstellen systematisch herauszuarbeiten und präzise zu beschreiben,
- dabei die vielfältigen Rahmen- und Randbedingungen, die sich aus den unterschiedlichen Zielsetzungen und der Einbettung in Produkt- und Systemlandschaften ableiten lassen, herauszuarbeiten und zu analysieren
- und die sich daraus ergebenden entwurfs- und realisierungsbezogenen Anforderungen - Ein-

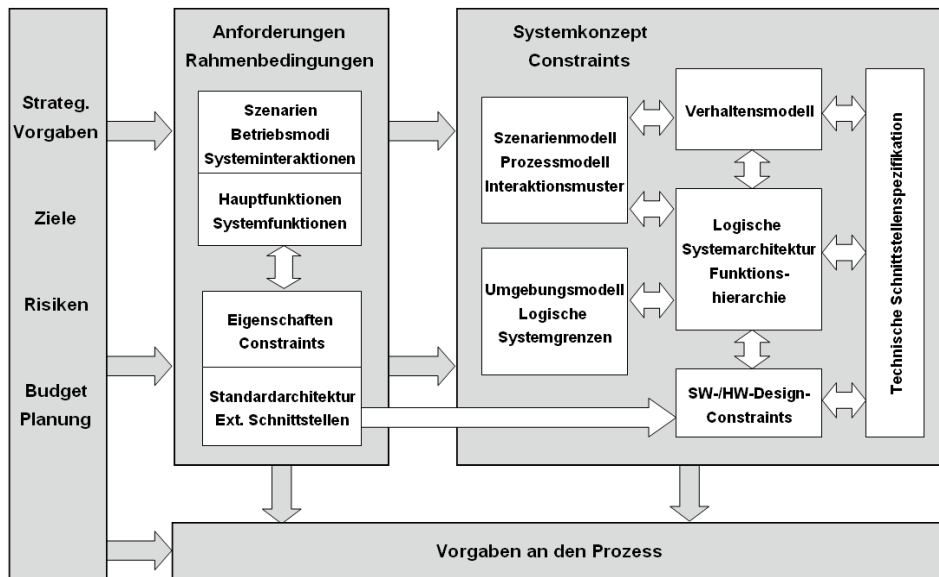


Abb. 1: RE Produktmodell – Modell von Spezifikationsprodukten

schränkungen an die Gestaltung der Funktionen/ des Verhaltens und technische Realisierung - frühzeitig in die Systemspezifikation mit einzubeziehen.

2 Modellbasiertes RE eingebetteter Systeme

Der zentrale Ansatz des modellbasierten RE liegt in der Einführung eines gemeinsamen Modells von zu erarbeitenden Spezifikationsprodukten – dem RE Produktmodell (s. Abb. 1) Es leitet die interdisziplinäre Erarbeitung und Abstimmung der Anforderungen mithilfe grundlegender Modelle und Konsistenzbedingungen. Seine wesentlichen Elemente sind die Definition der *Ziele* und *strategischen Vorgaben*, die daraus zu begründenden Anforderungen und *Rahmenbedingungen* an das zu entwickelnde System aus Kunden-/Nutzersicht und das zu entwerfende präzise abgestimmte *Systemkonzept* mit seinen *Constraints* an die *Schnittstellen* zu Systemumgebung und den weiteren *Entwurf* in den Disziplinen Software, Mechanik und Elektronik.

Das Produktmodell klassifiziert Anforderungen und zu modellierenden Systemansichten. Abbildung 2 zeigt das Klassifikationsschema für Anforderungen. Es garantiert die zielorientierte Erarbeitung und Bewertung von Anforderungen und Entwürfen: *Ziele* begründen die Definition und Gestaltung *funktionaler Anforderungen* und *Rahmenbedingungen* sowie den *detaillierten Systementwurf*.

Funktionale Anforderungen beschreiben die zu spezifizierende Nutzungsfunktionalität des zu-

künftigen Systems, und *Rahmenbedingungen* geben die hierbei einzuhaltenden Qualitätseigenschaften, Standards und Entwurfsmuster vor. Diese „high-level“ Anforderungen sind mithilfe der funktionalen Systemmodellierung zu verfeinern und daraus das Systemkonzept mit den *detaillierten Systemanforderungen* und *Constraints* an die weitere Entwicklung präzise abzustimmen und zu vervollständigen. Zwischen *Anforderungen* und *Rahmenbedingungen* bzw. *Constraints* ist eine methodische Entwurfsbeziehung definiert: *Rahmenbedingungen* (nicht-funktionale Anforderungen) gelten als abzustimmende Entwurfsentscheidungen und müssen aus *Zielen/Vorgaben* begründet sein. Sie bestimmen die Verfeinerung und Gestaltung der *funktionalen Anforderungen* und müssen systematisch auf die funktionalen Modelle des *detaillierten Systementwurfs* mit ihren Modell- und Konsistenzbeziehungen (Beziehungspfeile innerhalb des Systemkonzeptes in Abb. 1 und Abbildung 5) abgebildet werden.

Das funktionale Modellierungskonzept beschreibt eingebettete Systeme und ihren Einsatz mittels fünf wesentlicher „Sichten“: Ablaufsicht (*Prozess- und Szenariomodelle*), Struktursicht (*Umgebungsmodell, Systemgrenzen, Funktionshierarchie*) Interaktionssicht, Datensicht und Verhaltenssicht. Durch die Abbildung dieser Sichten auf ein einheitliches Systemkonzept sind Konsistenzbedingungen zwischen den Sichten definiert, die für die Überprüfung und Abstimmung erarbeiteter Anforderungs- und Systemmodelle genutzt werden können. Anforderungen verschiedener Stakeholder werden sukzessiv auf die Modellierungselemente der Systemsichten abgebildet, entsprechend strukturiert und mithilfe des zugrunde lie-

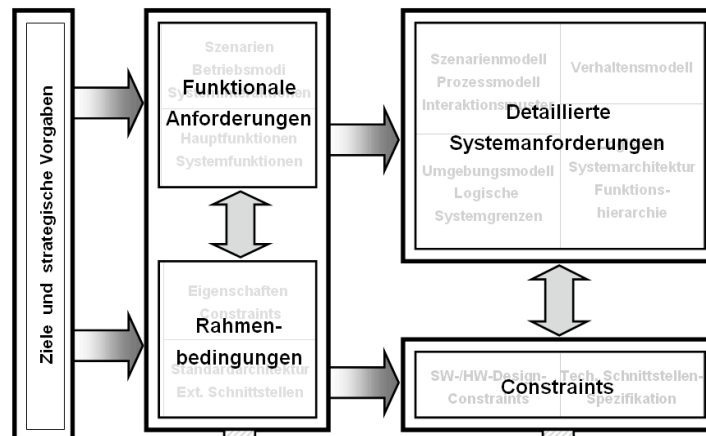


Abb. 2: Klassifikationsschema von Anforderungen

genden Systemmodells und seiner darin verankerten Konsistenzbedingungen analysiert und vervollständigt. Wesentlich für die erfolgreiche und zielorientierte Abstimmung der funktionalen Systemansichten ist der interaktive Einsatz entsprechender anschaulicher Beschreibungstechniken. Mit diesem strukturierten Modellierungsansatz ist eine Verständigungs- und Abstimmungsgrundlage für die interdisziplinäre Erarbeitung, Überprüfung und Validierung einer möglichst konsistenten und vollständigen Anforderungs- und Systemspezifikation gegeben. Das RE Produktmodell erlaubt die gemeinsame zielorientierte und verfolgbare Definition von Anforderungen und dient als Maßstab für Qualität und Fortschrittskontrolle der Spezifikation.

3 Requirements Management - Werkzeug AutoRAID

Die wesentlichen Konzepte des modellbasierten RE sind in dem Datenmodell des Werkzeuges AutoRAID (AutoFocus Requirements Analysis Integrating Development) festgeschrieben (s. Abb. 6 und 8). Es bestimmt die iterativ zu erarbeitenden Ergebnisse der fachübergreifenden Requirements Engineering Aktivitäten *Refinement*, *Classifying*,

Modelling und *Analysis* von Anforderungen und Systementwürfen.

Abbildung 3 zeigt das entsprechende iterative Vorgehen in AutoRAID. Der Schritt *Starting and Getting* bezeichnet den (wiederholten) Input von Anforderungen und die Aktivität *Systemdesign* das Abstimmen und Spezifizieren von Modellen/Entwürfen während des RE-Prozesses.

AutoRAID ist mit dem Spezifikationswerkzeug AutoFocus integriert und nutzt dessen Systemansichten und Konzepte für die strukturierte Analyse im Anforderungsprozess – hierarchische Darstellung der Systemkomponenten, Strukturdiagramme (*SSDs*), Zustandsautomaten (*STDs*), erweiterte Sequenzdiagramme (*EETs*) und Datentypdefinitionen (*DTDs*). Damit können dessen Verifikations- und Simulationsmöglichkeiten für die Überprüfung und Validierung der Anforderungen genutzt werden.

AutoFocus wurde am Lehrstuhl für Systems und Software Engineering der TU München als Forschungsprototyp entwickelt und in Industrieprojekten eingesetzt. Es verbindet Konzepte der Design-Modellierung, der Simulation, Code-/Testfallgenerierung, und erlaubt die Verifikation von Software-Komponenten. Abbildung 5 zeigt die dahinter liegende Konzeptwelt - das verein-

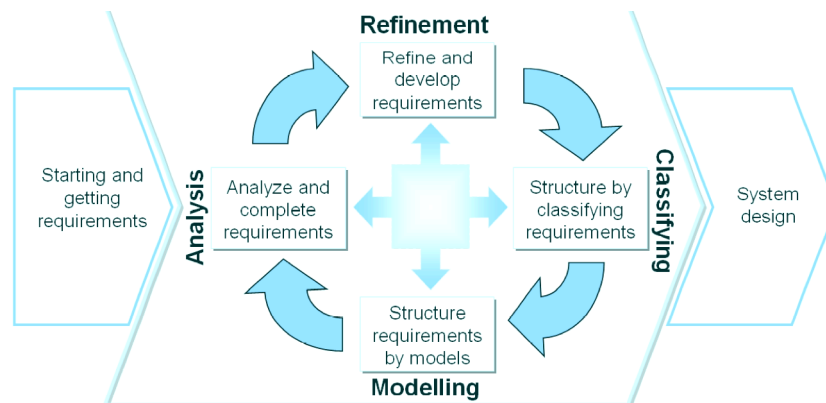


Abb. 3: Iteratives Prozessmodell in AutoRAID

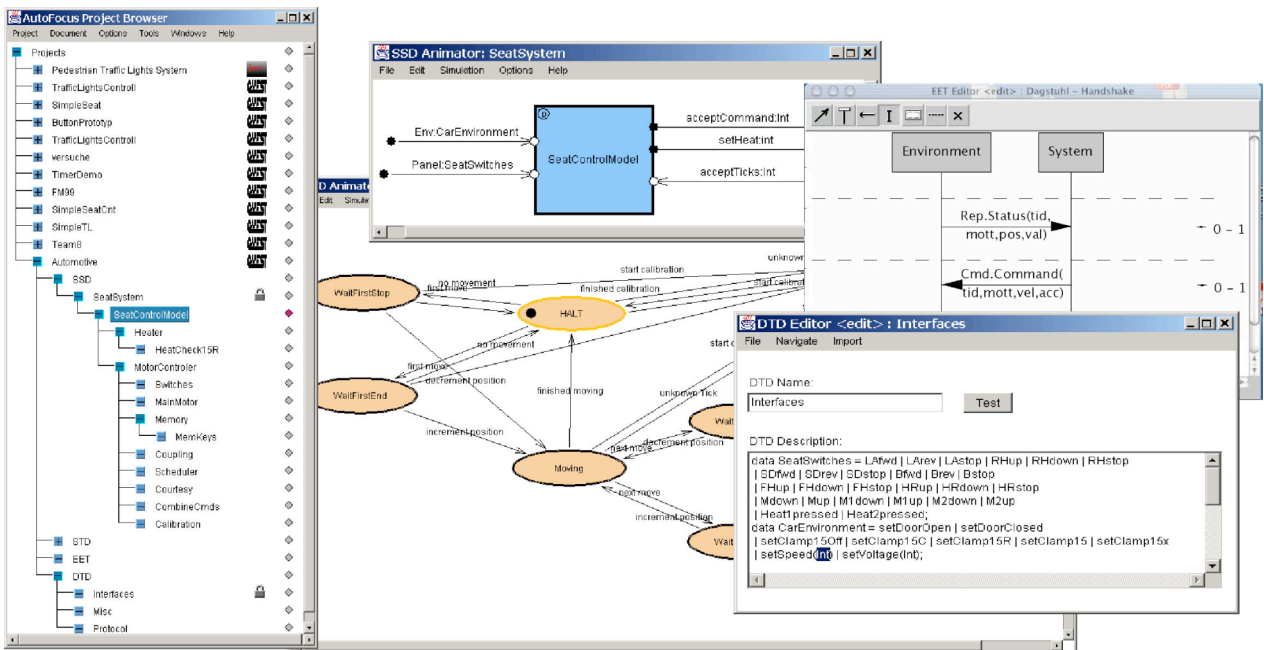


Abb: Modellierung von Systemen mit AutoFocus

fachte Datenmodell des Werkzeuges. Entsprechend der methodischen Schritte des iterativen Prozessmodells wird im Folgenden das modellbasierte RE in AutoRAID skizziert; eine detaillierte Beschreibung des Werkzeuges findet sich in [1].

3.1 Erfassen und Verfeinern von Anforderungen

Das Requirements Engineering beginnt mit der

Erhebung (*Elicitation*) von Anforderungen. Allgemeine Techniken sind strukturierte Workshops oder Interviews [5,8] und der Einsatz von Szenariotechniken [4]. In AutoRAID werden Anforderungen entweder manuell eingegeben, oder mittels „copy&paste“ und Importfunktionen aus strukturierten Dokumenten in die Verwaltung des Werkzeuges übernommen. Hierbei müssen in einem zugehörigen Fenster im Arbeitsbereich die

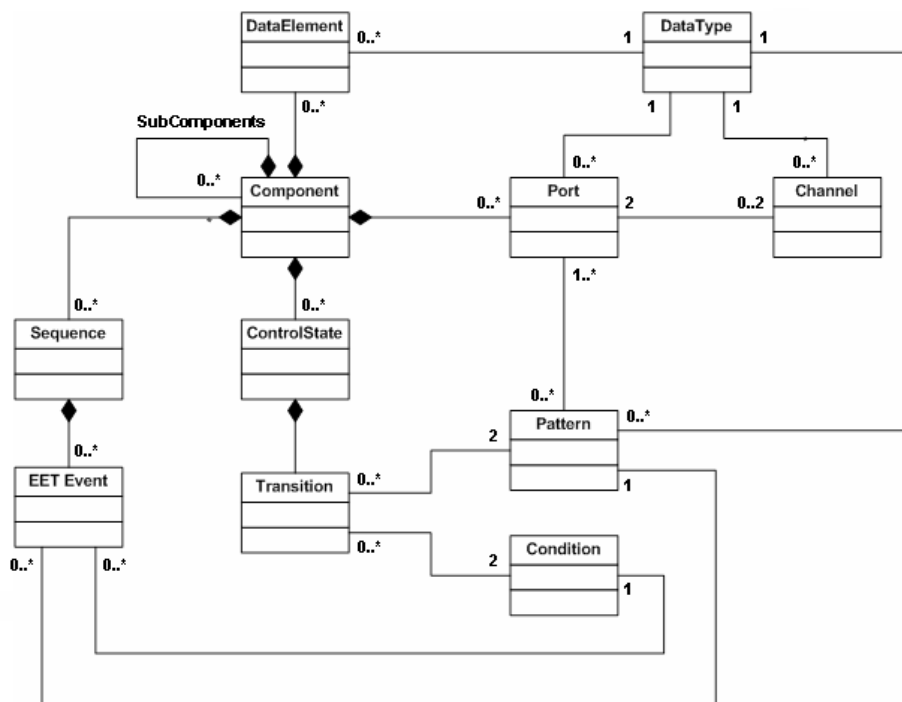


Abb. 5: Beispiele für Modellbeziehungen in AutoFocus

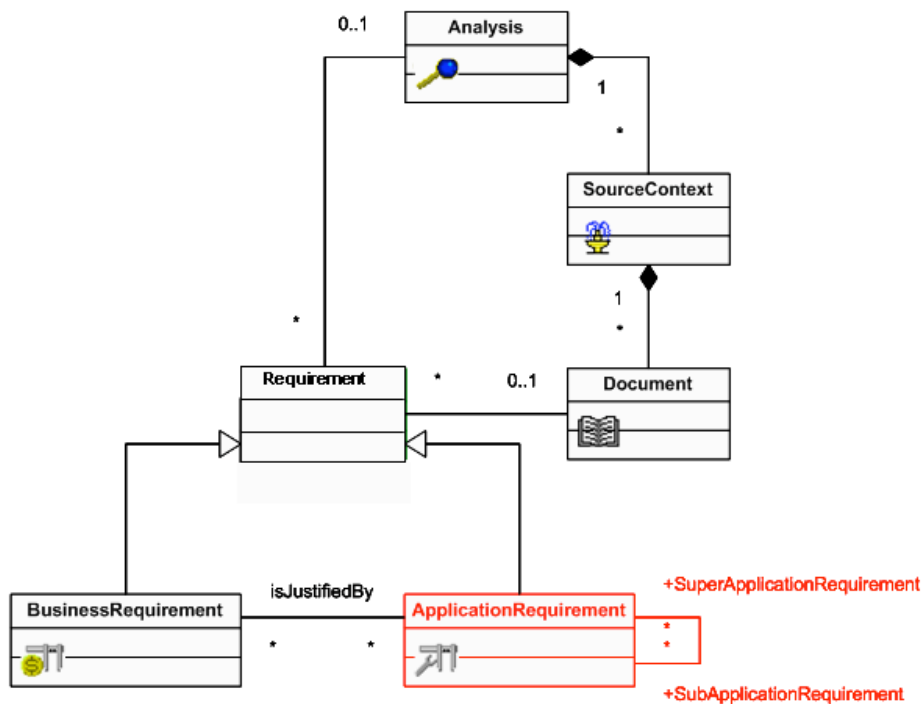


Abb. 6: Datenmodellausschnitt zur Strukturierung von Anforderungen

Anforderungsattribute wie *Titel*, *Beschreibung*, *Status*, *Priority*, usw. festgelegt werden. Abbildung 6 zeigt die entsprechenden Konzepte im Datenmodell und Abbildung 8 ihre Darstellung in der AutoRAID-Bedienoberfläche.

Alle Informationen der Anforderungsanalyse eines Projektes werden im *Analysis*-Baum eingehängt. Anforderungen kommen aus einem *Quell-Dokument*, das im Rahmen eines *Erhebungskontexts* (Meeting, Telefonat, usw.) entstanden bzw. eingebracht worden ist. Entsprechend sind diese *Context*-Ereignisse mit ihren Ergebnisdokumenten im *Project*-Baum strukturiert. Eingebrachte Anforderungen werden mit ihrer Identifikationsnummer und Titelbezeichnung (*Title*) unter dem *Requirements*-Baum des *Analysis*-Knoten eingehängt.

Entsprechend der zielorientierten Verfeinerung von Anforderungen können in AutoRAID aus Zielen (*BusinessRequirements*) Anforderungen (*ApplicationRequirements*) abgeleitet, und aus Anforderungen weitere Anforderungen (*SubApplicationRequirements*) gefolgert werden. Umgekehrt ist es möglich, erfasste Anforderungen hinsichtlich ihres Beitrages zur Erreichung gesetzter oder diskutierter Ziele zu analysieren und in entsprechenden Verfeinerungshierarchien zu strukturieren. Hierfür stellt AutoRAID entsprechende Bedienmechanismen bereit.

Abbildung 7 zeigt die entsprechende hierarchische Darstellung im *Requirements*-Baum sowie die verschiedenen Verfeinerungs- und Strukturie-

rungsfunktionen *Edit*, *Classify to*, *Refine*¹, *Create to* und *Associate*. Entsprechend dieser Typisierung wird im „Anforderungsblatt“ (rechts im Arbeitsfenster) das *Type*-Attribut festgelegt. Die Verfeinerungsbeziehungen werden im Anforderungsblatt einer Anforderung automatisch verzeichnet: Unter *Superrequirements* sind die übergeordneten Anforderungen, und unter *Subrequirements* die untergeordneten Anforderungen aufgelistet.

3.2 Klassifizieren und Modellieren

Entsprechend dem Klassifikationsschema von Anforderungen können in AutoRAID Anforderungen als

- *Use-Cases* – noch näher zu spezifizierende Anwendungsprozesse und geforderte Systemfunktionen/Dienste – oder
- *Constraints* – Spezifikation der Systemumgebung oder Forderung nach Systemanforderungen wie Architekturvorgaben, erforderliche Betriebszustände oder Datenschnittstellen verfeinert und spezifiziert werden.

Constraints werden unterschieden nach

- *ArchitecturalConstraints* - Strukturanforderungen an das zu entwickelnde System und seine Umgebung. Hier können die beteiligten Komponenten, ihre Schnittstellen und die

¹ Die Menü-Unterfunktionen von *Refine* sind *Edit Refinement*, *Copy*, *Move*, *Insert* und *Revise*.

Kommunikationskanäle festgehalten werden (*Struktursicht*).

- *ModalConstraints* - Betriebsmodi der Anwendung. Hier können verschiedene Zustände und Übergänge definiert werden (*zustandsorientierte Verhaltenssicht*).
- *DTDConstraints* - Datendefinitionen der Systemkommunikation/Schnittstellen oder zustandbestimmende Variable der Verhaltensmodellierung (*Datensicht*).

Ausgangspunkt für das Erarbeiten von funktionalen Anforderungen und Systementwürfen ist die umfangreiche Modellierung und Analyse der Anwendungs- und Nutzungsprozesse des Systems mithilfe der *Use-Case*- und *Scenario*-Modellierung. Beginnend mit dem informellen Einsatz grafischer Modellierungstechniken wie Aktivitätsdiagramme der UML werden schrittweise die wesentlichen Arbeitsschritte und Nutzungsfunktionen des Systems definiert und strukturiert in AutoRAID modelliert. Dieses Vorgehen und die detaillierte Analyse und Modellierung der erforderlichen Szenarien und Systeminteraktionen wird in Abschnitt 3.3 beschrieben.

Mit der Klassifizierung von Anforderungen als *Use-Case/Scenario*, *Architectural*-, *Modal*- oder *DTD-Constraint* ist der erste Schritt für die detaillierte Systemmodellierung durchgeführt. Nach und nach werden die Komponenten der Systemumgebung festgelegt, das System abgegrenzt, die Schnittstellen skizziert und die zu entwickelnden Systemfunktionen/Dienste identifiziert. Für diese Konstruktion und detaillierte Spezifikation des Komponenten und Systemverhaltens sind in AutoRAID die *Motivation*- und *Association*-Funktionen konzipiert.

Motivation

Mit der *Motivation-Funktion* in AutoRAID werden aus *Constraint*-Anforderungen entsprechende Modellelemente im Modellierungswerkzeug AutoFocus erzeugt. Abbildung 8 zeigt die zugehörige Konstruktionsbeziehungen (*Motivation*) zwischen den (An-) Forderungen nach bestimmten Modellelementen (*Constraints*) und den daraus zu modellierenden Systemelementen (*Component*, *Channel*, *State*, *TransitionSegment*, *Type*) des Spezifikationswerkzeug AutoFocus.

Abbildung 9 zeigt die zugehörige Bedienführung in AutoRAID für die *Motivation* und Konstruktion der Komponente *Engine* aus dem *Architectu-*

ralConstraint Engine. Sie erfolgt durch die Auswahl der Menüfunktion *Motivate – New Component*. Mit Ausführung dieser Funktion wird im Unterbaum *Modelviews - Architectural* eine entsprechende Komponente eingefügt, und gleichzeitig in der Systemmodellierung von AutoFocus das entsprechende Modell konstruiert. D.h. unter dem Entwurfsbaum *Component Car* wird die Komponente *Component Engine* eingehängt, und in der grafischen *SSD*-Modellierung die Komponente *Engine* eingezeichnet.

Die Ableitungsbeziehung zwischen Anforderungen und Modellelementen wird auf beiden Seiten festgehalten:

- In dem Anforderungsblatt der *Constraint-Anforderung ArchitecturalConstraint Engine* werden die aus ihr motivierten Modellelemente unter dem Attributblatt *Motivated Components* aufgelistet (hinteres Fenster im Arbeitsbereich).
- In den Informationen zu dem konstruierten *Modelview Engine* sind die motivierenden Anforderungen unter den Blatt *Motivations* aufgelistet (vorderes Fenster).

Association

Mit der *Association*-Beziehung im AutoRAID-Datenmodell können beliebige Anforderungen zuvor entworfenen (*motivated*) Systemmodellementen (Komponenten, Kanälen, Modes, Szenarien, usw.) zugeordnet und diese damit detailliert spezifiziert werden. Abbildung 9 zeigt eine erste Zuordnung von Anforderungen an die zu entwickelnde Systemkomponente *RevMeterController* (Spezifikationsblatt im Arbeitsfenster). Die Assoziationen sind in dem Attributblatt der *Associations* aufgelistet. Beispielsweise wird von dem *RevMeterController* die Funktion (*UseCase*) *Show_RPM_of_the_Engine* gefordert. Auch können nicht-funktionale Anforderungen wie beispielsweise „Der ‚Reporting‘-Dienst des Steuergeräts CCU darf max. 1 Sek. pro Std. nicht verfügbar sein“ auf diese Weise als Anforderungen an Komponenten (*Components*), Systemfunktionen/Diensten (*Use Cases*) oder Betriebsmodi (*Modes*) zugeordnet werden.

Mit den Möglichkeiten der *Motivation*- und *Association*-Beziehungen werden Anforderungen mithilfe funktionaler Systemsichten herausgearbeitet und entsprechend tiefer strukturiert bzw. modelliert.

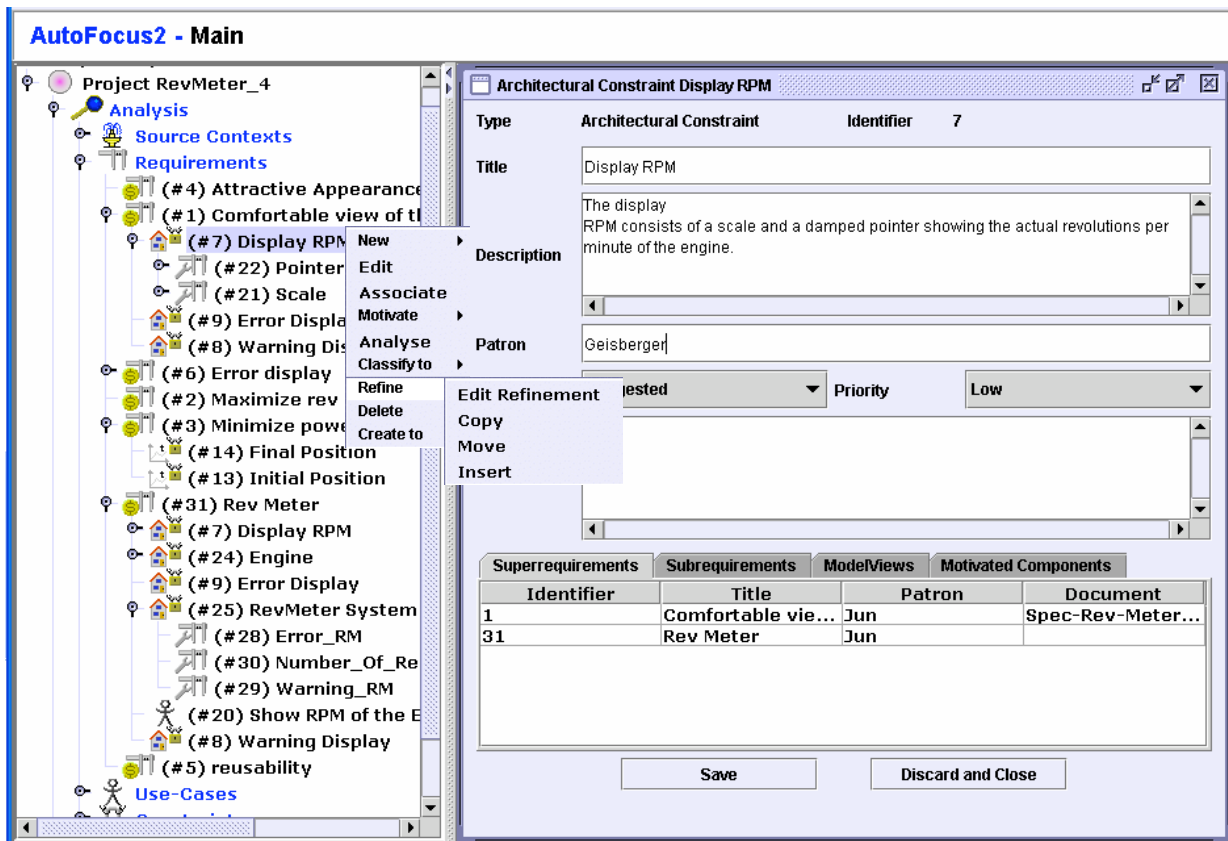


Abb.7: Schrittweise Verfeinerung von Anforderungen

Use-Case- und Szenario-Analyse

Wie eingangs beschrieben ist das grundlegende Mittel zur Erarbeitung und Verfeinerung funktionaler Anforderungen die systematische Prozessanalyse. Es werden

- die wesentlichen Arbeitsschritte und Aufgaben der Anwendung identifiziert (hierarchische *UseCase*-Strukturierung in AutoRAID),
- die relevanten Komponenten des Gesamtsystems identifiziert (*Motivate Architectural Constraints*),
- die erforderliche Kommunikation und Aufgabenverteilung in Gesamtsystem analysiert (Interaktionsanalyse: *CommunicationObservation*),
- die erforderlichen Betriebsmodi und Systemzustände erfasst (*Motivate Modal Constraints*) und
- die (Steuerungs-) Aufgaben – die Systemfunktionen - des zu entwickelnden Systems herausgearbeitet und strukturiert (hierarchischer *Use-Case*-Baum).

Abbildung 10 zeigt die entsprechende Darstellung von *Use-Cases* und *Scenarios* in AutoRAID. Einem Anwendungsfall können mit der Menüfunk-

tion *new Scenario* mehrere repräsentative Szenarien zugeordnet werden.

3.4 Analyse, Validieren und Vervollständigung

Ziel der Analyse ist es, Inkonsistenzen und Lücken in der Spezifikation zu erkennen und Anforderungen und Systemkonzepte in Hinblick auf die Produktziele und Kunden-/Nutzerbedürfnisse zu bewerten und entsprechend zu überarbeiten. Aufgrund der definierten Modell- und Konsistenzbeziehungen innerhalb und zwischen den Elementen des RE Produktmodells und den Systemansichten können die erstellten Modelle systematisch analysiert, überarbeitet und vervollständigt werden.

Wesentliches Mittel der Verfeinerung Vervollständigung des funktionalen Systementwurfs ist in AutoRAID die detaillierte Szenarioanalyse.

Szenario-Analyse mithilfe von Modellviews (Observations)

Jeder Schritt (*SequenceStep*) im informell beschriebenen Szenario (Abb. 10) wird hinsichtlich der Analyseaspekte Interaktion (*CommunicationObservation*), Betriebsmodus (*ModeObservation*) und Systemzustand *StateObservation* betrachtet, analysiert und die zugehörigen Systemansichten/Modelle erarbeitet und präzisiert.

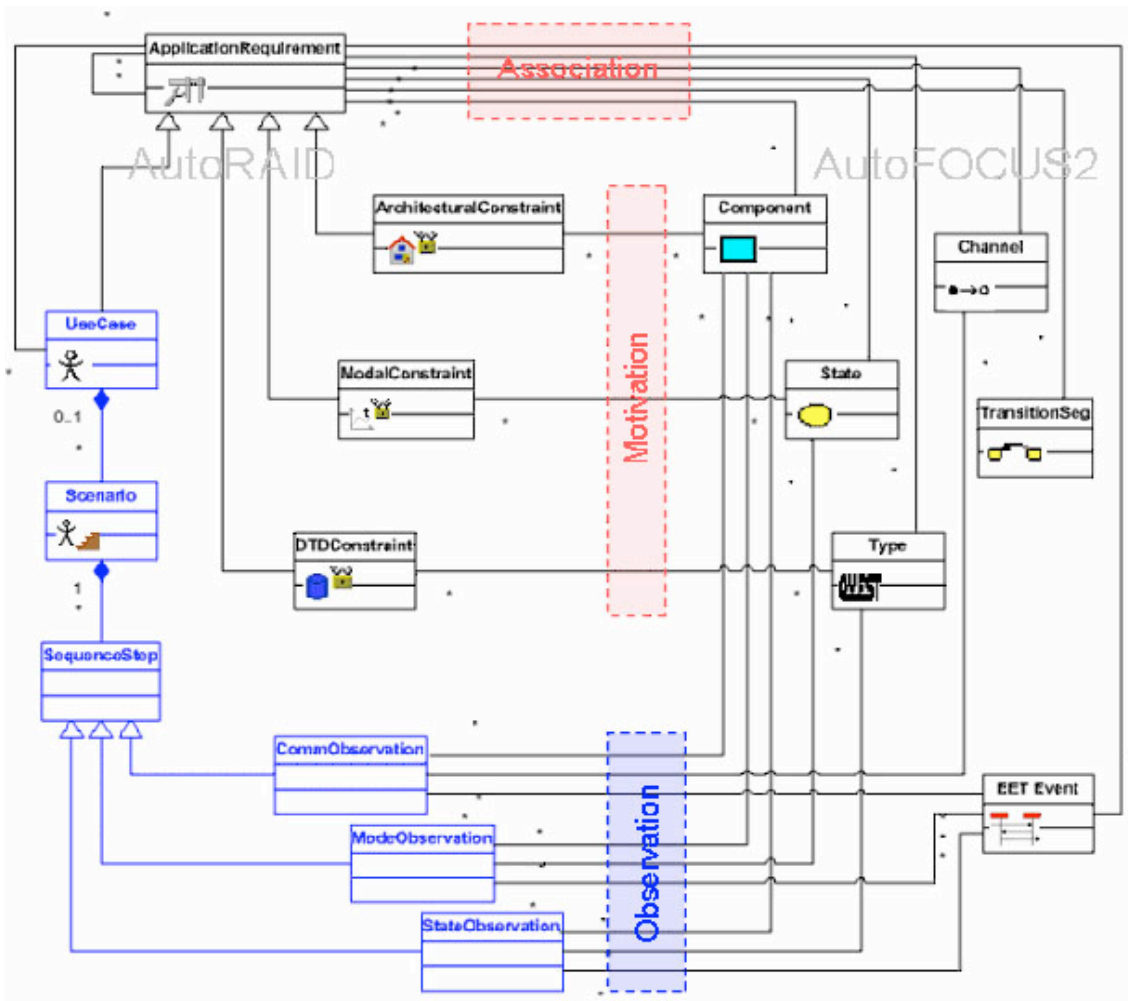


Abb. 8: Integration von Anforderungen und Constraints in AutoRAID

Abbildung 11 zeigt diesen Modellierungsschritt anhand der *CommunicationObservation* und für den Szenarioschritt 7. Mit der Auswahl der *Step Details*-Funktion erscheint das Fenster *Sequencestep* mit der Option der Kommunikationsanalyse (*Type: CommunicationObservation*). Hier werden Sender- und Empfängerkomponenten (*Sender, Receiver*), Kanal (*Channel*) sowie die Nachricht (*Signal*) eines Kommunikationsschritts festgelegt.

Dazu werden bereits vorhandene *Components* oder *Channels* ausgewählt, oder über neu zu erstellende *ArchitecturalConstraints* konstruiert und anschließend ausgewählt. Ergänzend werden mit der Auswahl der Menüfunktionen *ModeObservation* oder *StateObservation* die erforderlichen Mode-Bedingungen bzw. Zustandsvariablen für Interaktionsschritte – spezifiziert.

Sind die einzelnen *Sequencestep*-Modelle – und damit auch die entsprechenden Modellelemente in AutoFocus – durch die *Observation*-Analyse fest-

gelegt², kann mit der Ausführung der *Generate MSC* - Funktion die grafische Darstellung des spezifizierten Szenarios in Form von AutoFocus-EETs erzeugt (s. Abb. 12) und für die weitere Analyse oder Testfallgenerierung genutzt werden. Ähnlich der *Motivation*-Funktion wird mit der Generierung der Interaktionsmodelle für jeden Szenarioschritt (*Sequencestep*) eine entsprechendes Modellelement *EET Event* in AutoFocus erzeugt. Anforderungen an diese Interaktionsereignisse können dann auf Basis der *Association*-Beziehung zwischen *ApplicationRequirement* und *EET Event* zugeordnet, strukturiert und konkret beschrieben werden.

Folgende Analysemethoden werden mit dem AutoRAID-Ansatz unterstützt:

- Anhand der Modellbeziehungen können gezielt „Lücken“ in der Modellierung gefunden und Fragen zur Vervollständigung der Spezifikation diskutiert werden.

² Es genügt auch die teilweise Festlegung der *Observation*-Modelle. In diesem Fall werden mit der *Generate MSC*-Funktion unvollständige *EET*-Diagramme dargestellt. Sie demonstrieren „Lücken“ und sind anschaulicher Ausgangspunkt für die weitere Spezifikation.

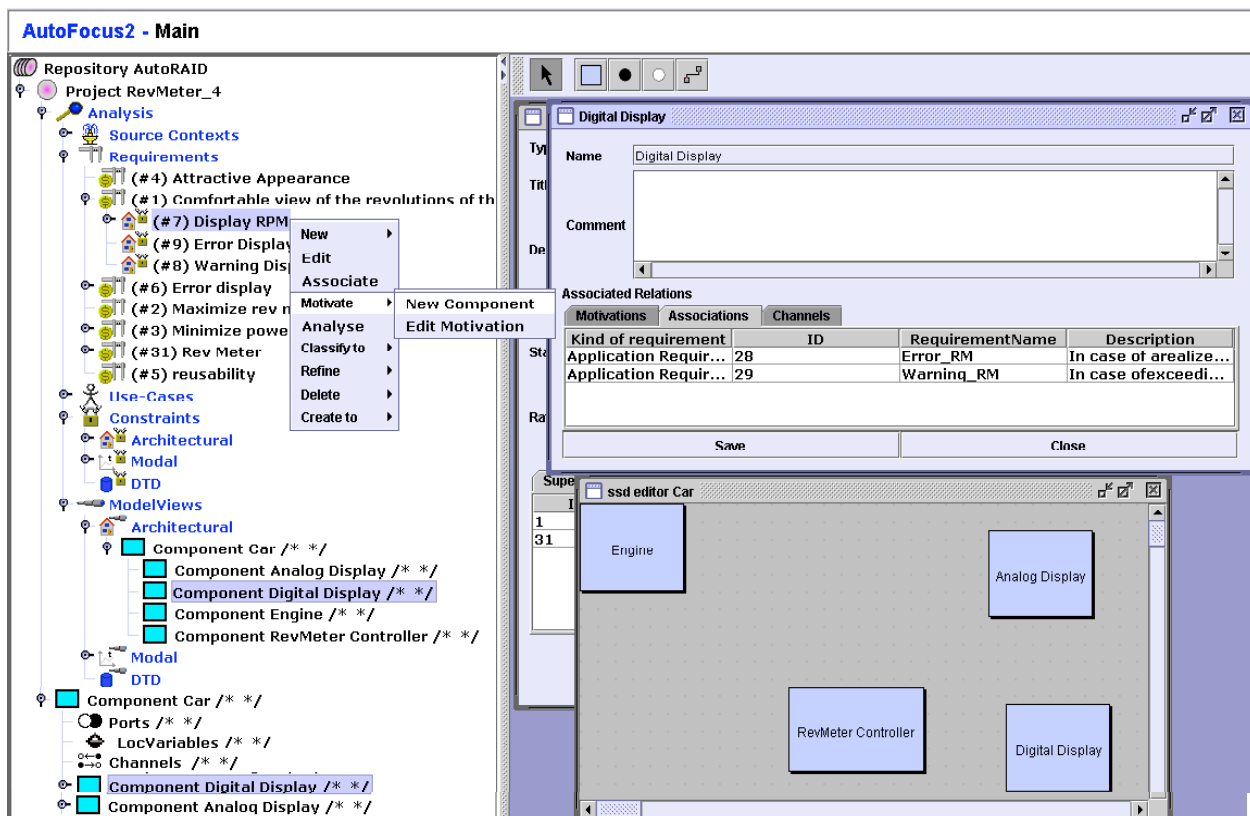


Abb. 9: Motivation- und Association-Funktionen in AutoRAID

- Anhand der in AutoFocus definierten Modellbeziehungen und den Assoziationsbeziehungen zu Anforderungen können Inkonsistenzen im Modell und damit auch Widersprüche in den Anforderungen systematisch erkannt, analysiert und entsprechend der Projektzielsetzungen überarbeitet werden.
- Anhand der zielorientierten Verfeinerungsbeziehung kann der Nutzen von Anforderungen und Entwürfen hinterfragt werden.

Anhand der problemlöseorientierten Klassifizierung von Anforderungen kann die Problemstellung systematisch analysiert (Situations- und Zielanalyse) und mögliche Lösungskonzepte zielorientiert bewertet werden. Anforderungen können von Entwürfen/ Vorgaben unterschieden, und damit der Nutzen einzelner Lösungskonzepte in Relation zur Machbarkeit und Aufwand für deren Realisierung bewertet werden.

Mit jeder Iteration erhöht sich der Strukturierungsgrad der Spezifikation und damit die Effektivität dieser Analysetechniken. Die Spezifikation wird somit schrittweise in einen konsistenten und vollständigen Zustand gebracht.

Damit ist in AutoRAID eine durchgängige Modellierung und Kopplung von Anforderungen mit dem Systementwurf gegeben. Aus Zielen und Anforderungen können sukzessive Modelle und

Systemkonzepte abgeleitet bzw. entworfen werden (*Forward-Tracing*), und umgekehrt, können und müssen Systementwürfe – also Lösungskonzepte – hinsichtlich ihres Nutzens bewertet, in das Gesamtkonzept eingeordnet und validiert werden (*Backward-Tracing*).

4 Vergleichbare Arbeiten

Kern des Ansatzes ist die Erarbeitung, Strukturierung, Analyse und Validierung/Vervollständigung der Nutzungs- und Systemanforderungen mit Hilfe von grundlegenden domänenspezifischen Modellen, sowie die konsequente Ableitung der Anforderungen und Systemdefinitionen aus Zielen (*Forward- und Backward-Tracing*). Durch die Umsetzung dieses Konzeptes in dem Werkzeug AutoRAID und seine Integration in das fundierte Spezifikationswerkzeug AutoFocus, können dessen Verifikationsmöglichkeiten für die Validierung und Vervollständigung der Anforderungen genutzt werden.

Ansätze zur Strukturierung der Anforderungen mit Hilfe von funktionalen Modellen sind in den Arbeiten von Kotonya und Sommerville zu VORD [9], Leite und Freeman („Requirements Validation Through Viewpoint Resolution“) [7], sowie Lamsweerde et al. zu KAOS [6] zu finden.

Kern von VORD ist die Analyse und Strukturierung

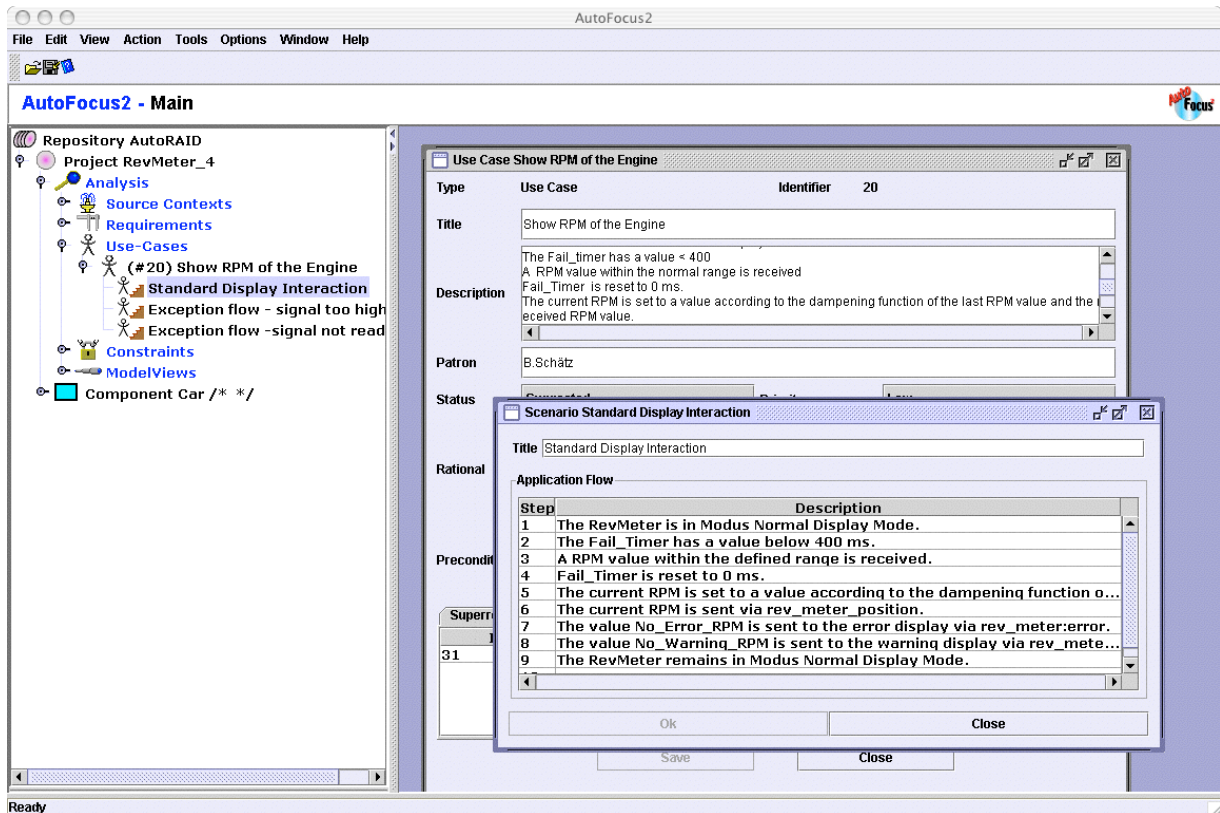


Abb. 10: UseCase- und Szenario-Modellierung in AutoRAID

nung von Anforderungen nach dem Nutzungsgesichtspunkt – nach „Services“. Es werden die Dienste des Systems mittels Szenarien beschrieben, und nicht-funktionale Anforderungen diesen Diensten zugeordnet. Diese Dienste wiederum können mittels „event traces“ und Zustandsautomaten spezifiziert wer-

den. Die Werkzeugunterstützung zu VORD erlaubt es, Konflikte zwischen Anforderungen verschiedener operationaler Viewpoints (Service-Spezifikationen) systematisch zu erkennen und diese aufzulösen. Eine Validierung der Anforderungen durch die Abbildung auf mathematisch fundierte Modelle und eine entsprechende tool-

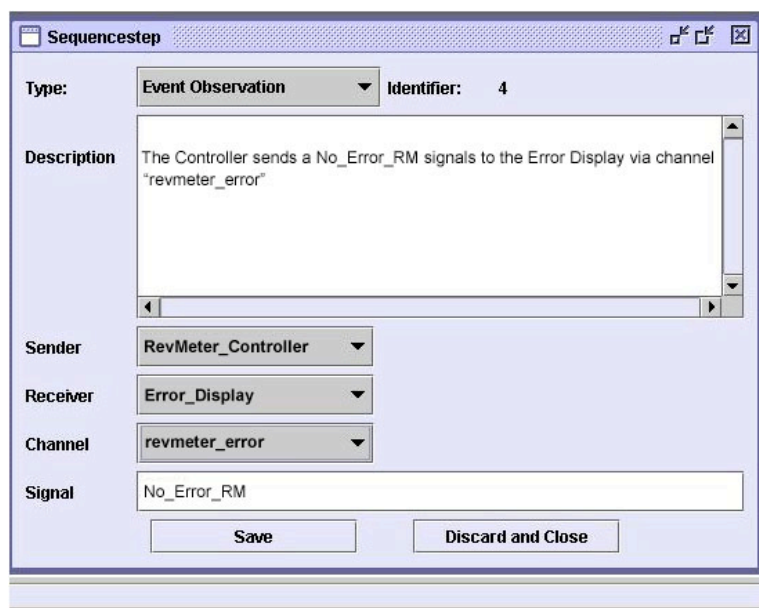


Abb. 11: Detaillierung von Szenarienschritten

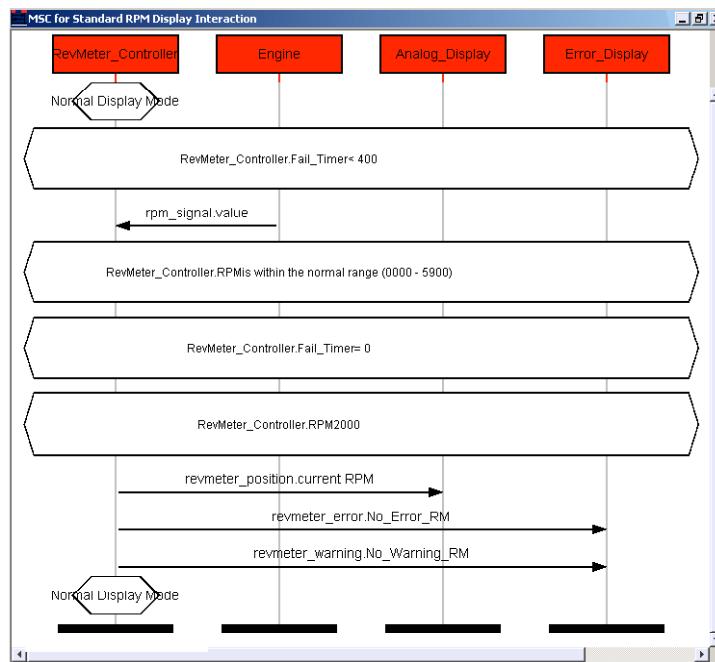


Abb. 12: Formalisiertes Szenario in AutoRAID

gestützte Verifikation ist in VORD bisher nicht gegeben.

Auch Leite und Freeman strukturieren Anforderungen verschiedener Viewpoints nach grundlegenden Modellen – Datensicht, „Actor“-/Prozesssicht, Architektursicht – und geben entsprechende Heuristiken an, mit denen Konflikte zwischen Anforderungen entdeckt und überarbeitet werden können. Eine Abbildung dieses sehr allgemeinen Strukturierungskonzeptes auf präzise oder mathematisch fundierte Aussagen mit entsprechenden Verifikationsmöglichkeiten ist auch hier nicht gegeben.

Goal-orientierte Ansätze wie KAOS [6] analysieren und erarbeiten Anforderungen (Goals) mit Hilfe der Wie- (top-down) und Warum-Fragen (bottom up). Zusätzlich definiert KAOS ein methodisches Konzept zur Verfeinerung von Zielen und Abbildung der gewonnenen detaillierten Anforderungen auf Softwarekomponenten (Agenten). So wird ein Strukturierungskonzept von Anforderungen nach funktionalen (Goals), nicht-funktionalen Anforderungen (Soft-goals) und einer weiteren Betrachtung der Anforderungen bezüglich ihrer Beziehungen untereinander (AND/OR-Strukturen) vorgeschlagen. Sind die Anforderungen an Agenten abgeleitet können diese präzise mit Hilfe temporaler Logik spezifiziert und verifiziert werden. Eine sukzessive Erarbeitung und Strukturierung der Anforderungen mit Hilfe funktionaler Modelle ist in KAOS nicht gegeben, da zwischen der methodischen Strukturierung der (Soft-) Goals und der funktionalen Strukturierung der detaillierten Anforderungen an Agenten ein Bruch besteht.

In [3] wird eine review-basierte Vorgehensweise zur schrittweisen Strukturierung von textuellen Anforderungen mittels tabellarischer Use-Case-Beschreibungen sowie Statechart-Diagrammen beschrieben. AutoRAID vereinfacht einen solchen Review-prozess durch die werkzeuggestützte Unterstützung einzelner Transformationsschritte und bietet Analyseverfahren sowie die Einbeziehung weiterer Aspekte (z.B. Daten, Struktur).

5 Zusammenfassung und Ausblick

AutoRAID unterstützt die modellbasierte Anforderungsanalyse durch einen strukturierten und schrittweisen Übergang von textuellen Anforderungen zum Entwurfsmodell. Wesentliche Funktionalitäten sind dabei ein detailliertes konzeptuelles Modell mit unterschiedlichen Klassen von Anforderungen (z.B. architectural constraints, modal constraints) und werkzeuggestützte Schritte zum Herausarbeiten der Anforderungen (z.B. Klassifikation). Die aktuellen Arbeiten erweitern diese zu einer *tiefen Integration domänenspezifischer Anforderungen*.

Durch die Verwendung eines Produktmodells mit erweiterten, domänenspezifischen Anforderungen (z.B. Zeitanforderungen) und Sichten wird eine tiefere Strukturierung und stärkere Kopplung zwischen den Anforderungen und den Modellsichten in AutoFocus möglich. Ziel ist dabei neben detaillierten generativen Schritten (z.B. Generierung von Testfällen aus strukturierten Anwendungsszenarien) die Unterstützung komplexer Analysetechniken (z.B. Konsistenz der Schnittstellen der architektonischen Anforderungen und der Schnitt-

stellen in den Funktionsanforderungen; Konsistenz von Anwendungsszenarien und Verhaltensbeschreibungen).

Literaturverzeichnis

1. AutoRAID Webseite:
<http://www.broy.in.tum.de/~autoraid/>
2. AutoFocus Webseite: <http://autofocus.in.tum.de>
3. Denger, C. Paech, B. An Integrated Quality Assurance Approach for Use Case Based Requirements. In: Rumpe, B. Hesse, W. (eds). Modellierung 2004. Springer, LNI, 2004.
4. Haumer, P., Jarke, M., Pohl, K., Weidenhaupt, K., Scenarios in System Development: Current Practice, IEEE Software, Vol. 15, April 1998, pp. 34-45
5. Kontonya, G. and Sommerville, I., Requirements Engineering: Processes and Techniques. Wiley & Sons, 1998.
6. Van Lamsweerde, A. Goal-Oriented Requirements Engineering: A Guided Tour. RE'01 - 5th IEEE Int.Symposium on Requirements Engineering, Toronto 2001, pp. 249-263.
7. Leite, L., Freeman, P.: Requirements Validation Through Viewpoint Resolution. IEEE Transactions of Software Engineering, 1991.
8. Sommerville, I., Sawyer, P.: Viewpoints for Requirements Elicitation: A Practical Approach. IEEE Int. Conf. on Requirements Engineering, IEEE Press, 1998.
9. Kotonya, G., Sommerville, I.: Requirements Engineering With Viewpoints. Technical Report, Lancaster Univ., 1995