



ulm university universität
uulm

Entscheidungsdokumentation bei der Entwicklung innovativer Systeme für produktlinien-basierte Entwicklungsprozesse

**Ramin Tavakoli Kolagari, Alexander Raschke,
Matthias Schneiderhan, Ian Alexander**

Ulmer Informatik-Berichte

**Nr. 2008-04
Februar 2008**

Entscheidungsdokumentation bei der Entwicklung innovativer Systeme für produktlinien-basierte Entwicklungsprozesse

Erfahrungsbericht zum Praktikum Experimentelles Software Engineering im Sommersemester 2005

Ramin Tavakoli Kolagari¹, Alexander Raschke²,
Matthias Schneiderhan³, Ian Alexander⁴

Zusammenfassung

Die Entwicklung innovativer Systeme ist trotz intensiver Forschung auf diesem Gebiet eine herausfordernde Arbeit. Innovative eingebettete Systeme in der Automobilindustrie stellen hohe Anforderungen an die Entwickler, da diese zumeist Systeme für einen konkreten Bedarf entwickeln, der in dieser Ausprägung bisher nicht existiert und damit auch keine dedizierten Stakeholder befragt werden können. Im Zuge dieser Entwicklungsarbeiten wird ein intensives Domain Engineering betrieben, bei dem viel Wissen über die zu entwickelnde Domäne angesammelt wird, das auch für die Fortentwicklung der Systeme relevant ist. Da in der Industrie Wiederverwendung eine beachtliche Rolle spielt, da durch sie eine Senkung der Entwicklungskosten sowie eine Steigerung der Qualität unterstützt wird, muss man von Beginn einer Entwicklung an die Wiederverwendbarkeit von Wissen oder Komponenten unterstützen. Software-Produktlinien sind eine besonders erfolgreiche Technik, um Wiederverwendung zu gestalten. In dem vorliegenden Bericht beschreiben wir ein Experiment, das zwei unterschiedliche Beschreibungstechniken für das während der frühen Phase der Innovationsentwicklung anfallende Wissen vergleicht, um es für die Einbettung der Innovation in eine Software-Produktlinie zu nutzen. Verglichen werden klassische Sitzungsprotokolle mit Entscheidungsmodellen, die auf der Arbeit von James Dewar und Stephen Toulmin aufsetzen.

Das in diesem Bericht beschriebene Experiment wurde als Praktikum für Studierende im Hauptstudium an der Universität Ulm im Sommersemester 2005 durchgeführt. Die aus dem Experiment gewonnenen Hinweise deuten an, dass eine sinnvolle Verwendung von den von uns vorgestellten Entscheidungsmodellen mit weniger Aufwand zu einem besseren Verständnis der Domäne der Innovation insgesamt führt und dieses Wissen auch leichter anderen Personen zugänglich ist.

¹Fachgebiet Softwaretechnik, FR5-6, Technische Universität Berlin
E-Mail: tavakoli@cs.tu-berlin.de

²Institut für Programmiermethodik und Compilerbau, Universität Ulm
E-Mail: Alexander.Raschke@uni-ulm.de

³Institut für Programmiermethodik und Compilerbau, Universität Ulm
E-Mail: Matthias.Schneiderhan@uni-ulm.de

⁴Scenario Plus Ltd, London, England
E-Mail: ian@scenarioplus.org.uk

Inhaltsverzeichnis

1	Einleitung	3
2	Grundlagen und Ziele	5
2.1	Software-Produktlinien und Innovationen	5
2.2	Toulmin-basierte Entscheidungsmodelle	6
2.3	Werkzeug Statemate	11
2.4	Ziel und Szenario des Experiments	13
3	Durchführung des Experiments	17
3.1	Grobaufbau des Experiments	20
3.2	Einführungswochen	21
3.3	Iterationsübersicht	22
4	Verwendete Dokumente	25
4.1	Aufwandsfassung	26
4.2	Anforderungsdokument	27
4.3	Grobe System-Anforderungen	28
4.4	Testprotokoll	28
4.5	Presseberichte	28
4.6	Kontextdiagramm	28
4.7	Merkmaldiagramm	29
4.8	Projektprotokoll	30
4.9	Szenarien	31
5	Ergebnisse	31
6	Kritische Diskussion des Experiments und Ausblick	41
7	Anhang	45
7.1	Iterationsanforderungen und Presseberichte	45
7.2	Anforderungstabelle	72
7.3	Merkmaldiagramm	75
7.4	Entscheidungsmodelle	76
7.5	Szenarien	79
7.6	Statemate-Modell am Beispiel vom Spurassistenten-System	80
7.7	Projektprotokoll	83
7.8	Aufwände der Gruppen je Iteration	86
7.9	Auswertung der Szenarien der Gruppen je Iteration	88
7.10	Fragebögen zur qualitativen Auswertung	89

1 Einleitung

Die Entwicklung von Innovationen spielt eine wichtige Rolle vor allem in den Industrien, die sich durch technologische Errungenschaften auf dem Markt positionieren. Da die Entwicklung von Innovationen kostenintensiv ist, muss von Anfang an darauf geachtet werden, dass diese Innovationen am Markt bestehen können und damit auch weiterentwickelt werden können [PTK04]. Für den Transport der Information über die verschiedenen Iterationen der Weiterentwicklung gibt es verschiedene Ansätze. Jeder dieser Ansätze besitzt seine Stärken und Schwächen. Bei dem im Folgenden beschriebenen Experiment geht es darum, diese Stärken und Schwächen zu untersuchen. Dies erfordert natürlich eine empirische Herangehensweise, bei der gezielt die Wirkungsweise einzelner Techniken der Informationsbeschaffung und -weiterleitung beobachtet werden können soll. In diesem Bericht präsentieren wir den Aufbau und die Ergebnisse einer solchen Untersuchung.

Häufig ist es in der Industriepraxis nicht möglich, Experimente im laufenden Betrieb durchzuführen. Ebenso sind Vergleiche von unterschiedlichen Techniken nur schwer möglich, da sie meist aus Kosten- und Zeitgründen nicht durchgeführt werden. Ein wesentlicher Punkt sind auch die Risiken, die neue Techniken mit sich bringen: Bei unbekanntem Herangehensweisen sind die Auswirkungen der einzelnen Bestandteile der Technik häufig nicht eindeutig vorhersehbar. Darum eignet es sich, ein Experiment, das neue Techniken in der Praxis vergleichen möchte, mit Studierenden im Rahmen der universitären Ausbildung durchzuführen, wenn sich der Einarbeitungsaufwand in Grenzen hält und die Probanden für das Experiment nicht besondere Vorkenntnisse aufweisen müssen, die man von den Studierenden nicht erwarten kann. Einschränkungen der Empfehlung, derartig geartete Experimente mit Studierenden durchzuführen, werden im letzten Abschnitt dieses Berichts diskutiert.

In dem vorliegenden Bericht beschreiben wir ein Experiment, das zwei unterschiedliche Beschreibungstechniken für das während der frühen Phase der Innovationsentwicklung anfallende Wissen vergleicht, um es für die Einbettung der Innovation in eine Software-Produktlinie zu nutzen. Verglichen werden klassische Sitzungsprotokolle mit Entscheidungsmodellen, die auf der Arbeit von James Dewar und Stephen Toulmin aufsetzen. Einzelheiten dazu werden in den folgenden Abschnitten gegeben.

Zusätzlich zu der Entwicklung der Innovation auf Ebene der Anforderungen sollten die Studierenden auch eine Umsetzung der von ihnen entworfenen Systeme vornehmen. Dabei wurde auf das kommerzielle Werkzeug Statemate von I-Logix zurückgegriffen [I-L]. Bei diesem Werkzeug handelt es sich um ein Werkzeug, mit dem man hierarchische Zustandsautomaten (Statecharts) erstellen kann. Mit Hilfe eines Panels, in dem man die entsprechende Hardware, die angesteuert werden soll, nachmodellieren kann, kann der von den Studierenden erstellte Zustandsautomat auf seine Funktionsweise getestet werden. Die Dokumente, die Modelle und der Automat einer Gruppe werden dann in der nächsten Iteration an eine andere Gruppe weitergereicht. Nach jeder Iteration wurden zusätzliche Informationen und Anforderungen an die Innovation an die Studierenden weitergereicht, welche in der folgenden Iteration in das schon bestehende Modell eingebaut werden sollten.

Im folgenden Abschnitt 2 wird zunächst auf die Grundlagen und Ziele des Experiments eingegangen. Im Anschluss daran werden die Zusammenhänge zwischen innovativen Systemen und Software-Produktlinien erläutert. Die Entscheidungsmodelle nach Toulmin und Dewar sind Gegenstand des darauf folgenden Abschnitts. Dabei skizzieren wir kurz die Historie Argumentations-orientierter Modelle, geben allerdings einen umfassenden Überblick über die von uns vorgeschlagenen Modelle inklusive der Erweiterungen, die wir für notwendig befunden

haben, um sie für die Anwendung in dem von uns betrachteten Umfeld tauglich zu machen. Weiterhin wird beschrieben, wie eine Umsetzung des Modells in dem Werkzeug DOORS aussehen kann. Im Abschnitt *Werkzeug Statemate* (2.3) wird auf das verwendete Modellierungswerkzeug Statemate von I-Logix eingegangen. Die Studierenden mussten zwar keine Implementierung der von ihnen zu entwickelnden Innovation durchführen, sollten aber eine testbare Version ihres Produktes in Statemate als Zustandsautomat erstellen. In diesem Abschnitt wird kurz erläutert, wofür sich Statemate eignet und in welchem Zusammenhang es in dem Experiment verwendet wurde. Zum Ende des Abschnitts 2 folgt eine Beschreibung des Ziels und des zugrunde liegenden Szenarios des Experiments. In diesem Zusammenhang wird auch auf den qualitativen Charakter des Experiments eingegangen. Da sich im Rahmen dieses universitären Experiments mit der gegebenen Anzahl an Studierenden und der unzureichenden Möglichkeit der Kontrolle von Störvariablen keine statistisch signifikanten Ergebnisse erzielen lassen, haben wir versucht, die Studierenden in einer Doppelrolle agieren zu lassen, indem sie sowohl Probanden als auch Forscher waren, die sich selber beobachten und daraus Schlüsse ziehen und Empfehlungen und Verbesserungsvorschläge geben, die auch bei dem weiteren Verlauf des Experiments beachtet wurden.

Im darauf folgenden Abschnitt 3 wird auf die *Durchführung des Experiments* eingegangen. Nach einer kurzen Einleitung in die wichtigsten Einschränkungen und Umgebungsbedingungen folgt der Grobaufbau des Experiments. Hier werden der Aufbau und die zeitliche Abfolge von Trainings- und Experimentphase erläutert. Des weiteren wird die Verteilung der Studierenden auf die Entwicklungsprojekte der Innovationen angegeben. In der *Iterationsübersicht* wird die Experimentphase genauer betrachtet. Eine einzelne Iteration wird hierzu in kleinere Teile zerlegt, die bestimmte Aufgabenteile enthalten. So gliedert sich eine Iteration in eine „Analyse- und Entwurfsphase“ und in eine „Modellierungsphase“. In einer genauen Wochenübersicht über eine Iteration werden die gestellten Anforderungen an die Studierenden verdeutlicht. Dabei spielen die Materialien, die die Studierenden bekamen, ebenso eine Rolle, wie die abzugebenden Dokumente. In der letzten Woche des Experiments sollten die Studierenden dann ein Merkmaldiagramm entwickeln, das eine Übersicht über die Produktvarianten der entstandenen Innovation darstellt.

In Abschnitt 4 werden die von den Studierenden zu *verwendenden Dokumente* dargestellt. Auf Grund der zwei unterschiedlichen Herangehensweisen mussten die beiden Gruppen zum Teil unterschiedliche Dokumente bearbeiten und abgeben.

Die Auswertung der Ergebnisse ist in Abschnitt 5 beschrieben. Hier wird in einer tabellarischen Auswertung auf die erzielten Ergebnisse eingegangen. Hierbei werden alle Dokumente sowie die Aufwandserfassung in Betracht gezogen. Ein Vergleich zwischen den einzelnen Gruppen soll Aufschluss darüber geben, welche Herangehensweise sich in diesem speziellen Umfeld als erfolgreicher erweist. Wie oben beschrieben, lassen sich aus dem Experiment keine statistisch signifikanten Ergebnisse ableiten. Daher steht bei der hier beschriebenen Auswertung vor allem die subjektiv durch die Studierenden empfundene Nützlichkeit sowie der Aufwand bei der Arbeit mit ihren unterschiedlichen Vorgehensweisen im Vordergrund. Ein weiterer Indikator für die erfolgreiche Anwendung einer Methode ist die Aussagekraft der Merkmaldiagramme: Sie geben einen Eindruck vom Verständnis der Innovation als ein Produkt in unterschiedlichen Varianten, was die Basis darstellt für eine Software-Produktlinie.

2 Grundlagen und Ziele

In diesem Abschnitt werden zunächst die für das Praktikum relevanten Grundlagen der verwendeten Methodik und Werkzeuge beschrieben. Darauf aufbauend werden dann die motivierenden Fragestellungen konkretisiert und die Methodik für diesen Versuch dargestellt.

Für unser Experiment sollten die teilnehmenden Studierenden insgesamt drei verschiedene innovative Systeme in unterschiedlichen Iterationen für einen PKW entwickeln (Airbag, Spurassistent, Türschloss). Für unseren Versuch mussten die Studierenden nicht implementieren, es reichte, das funktionale Verhalten des Steuergeräts in Zustandsautomaten mit Hilfe des Werkzeugs Statemate zu modellieren.

In den folgenden Abschnitten werden wir die Grundlagen für Software-Produktlinien einführen. Des Weiteren beschreiben wir die von uns auf Basis der Toulmin-Argumentation entwickelten Entscheidungsmodelle, die ein Teil der Studierenden für die Beschreibung des von ihnen generierten Wissens nutzten. Dann wird das Werkzeug Statemate kurz vorgestellt und in dem Unterabschnitt 2.4 geben wir einen Überblick über die Ziele und das zugrundeliegende Szenario des Experiments.

2.1 Software-Produktlinien und Innovationen

Eine *Software-Produktlinie* ist „a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way“ [CN02]. Die grundlegende Idee von Produktlinien ist damit die Erstellung von Kernelementen für alle bei der Entwicklung anfallenden Artefakte für einen gewissen *Scope* von Produkten, die gemeinsam verwaltet werden. Dies wird typischerweise in einem zweiteiligen Prozess durchgeführt: *Domain Engineering* und *Application Engineering*, die beide durch Management-Aufgaben in Relation gesetzt werden. Die Konzentration auf die Kernelemente, die über alle Produkte im Scope der Produktlinie unverändert gleich bleiben und die Konsistenz des Ansatzes über die Artefakte hinweg sind die beiden wesentlichen Vorteile der Produktlinie. Damit können sich die Management-Aufgaben auf die wesentlichen Aspekte (die Kernelemente) konzentrieren, ohne mit der Verwaltung beispielsweise individueller Features überladen zu sein, wie es in anderen Wiederverwendungsansätzen durchaus der Fall ist. Außerdem ist es bei der Produktentwicklung nicht nötig, einen Pool an möglichen Features für sein System zu explorieren, wie dies der Fall bei komponentenorientierten Ansätzen [AM02, ABM00] ist; stattdessen werden die Kernelemente gemäß den Bedürfnissen der anstehenden Systementwicklung erweitert. Um dies zu ermöglichen, wurde das Konzept der Variationspunkte eingeführt: Variationspunkte können zu jedem Kernelement annotiert werden, und beschreiben damit eine mögliche, angedachte Erweiterung des Systems an diesem speziellen Punkt. Des Weiteren können die Variationspunkte den Typ der Erweiterung beschreiben (beispielsweise „optional“, „Alternative“) und in einer Erläuterung können mögliche Entwurfsentscheidungen oder Abhängigkeitsbeziehungen zu anderen Variationspunkten beschrieben werden.

Ursprünglich in der Forschung für Software-Architekturen entwickelt [BCK03, Bos00], finden Software-Produktlinien ihren wesentlichen Gewinn auch in diesem Bereich der Software-Entwicklung, da die Architektur der entscheidende Link zwischen den Anforderungen und dem Code ist. Änderungen an der Architektur sind einer der Haupttreiber für Kosten bei der Software-Entwicklung. Daher ist ein ökonomischer Ansatz für die Einführung von Produktlinien nur dann sinnvoll, wenn sowohl Anforderungen als auch Architektur, Code und Testfälle

adressiert werden. Grundsätzlich ist allerdings die Einführung von Produktlinien mit hohen Investitionskosten verbunden, im Wesentlichen durch Änderungen im Organisationsaufbau und gesteigerten Anforderungen an die Kommunikation.

Um die hohen Investitionskosten bei Software-Produktlinien dennoch besser in den Griff zu bekommen ist es notwendig, von Anfang an für die Gestaltung von Software-Produktlinien vorzusorgen. Eine wesentliche Voraussetzung bei klassischen Software-Produktlinien-Ansätzen ist, dass man das Domain Engineering auf einer Menge an bereits vorhandenen Produkten ausführt. Damit werden innovative Systeme nicht in Produktlinien eingebracht, weil bei der Entwicklung eines innovativen Systems eben keine Vorgängerprodukte zur Verfügung stehen, die mit betrachtet werden können. Damit setzen Software-Produktlinien auf einer Menge von Produkten auf, die in *produktspezifischen Prozessen*, *produktspezifischen Projektstrukturen* und auf Basis von *produktspezifischen Entscheidungen* erstellt wurden; der Aufwand, diese Strukturen in einen Produktlinien-Ansatz zu übertragen, bringt die hohen Investitionskosten mit sich.

Das hier beschriebene Problem ist in der Praxis einer der Gründe, dass Produktlinien noch nicht so umfassend eingesetzt werden, wie es sich die Forschung wünscht und wie auch die bisherigen Erfolgsmeldungen vermuten lassen würden. Innovative Systeme werden, wenn sie am Markt bestehen, weiterentwickelt und für unterschiedliche Produkte in unterschiedlichen Ausprägungen erstellt — sie gehen über in eine Produktfamilie. Um diese Familie verwaltbar zu gestalten, muss sie mit Mitteln, die Produktlinien zur Verfügung stellen, entwickelt werden. Um dies schon bei der frühen Entwicklung von Innovationen zu unterstützen, haben wir ein Modell entwickelt, das die Gründe, die den Entscheidungen, die zu einem Produkt führen, zugrunde liegen, dokumentiert. Damit regen wir die Entwickler dazu an, sich nicht nur mit den *produktspezifischen Entscheidungen* zu beschäftigen, sondern hier schon *produktlinienorientierte Entscheidungen* zu beachten. Im folgenden Abschnitt gehen wir genauer auf die vorgeschlagenen Modelle ein.

2.2 Toulmin-basierte Entscheidungsmodelle

Ein wesentliches Problem in sämtlichen Wiederverwendungsansätzen ist, dass Dokumente, Modelle oder andere Artefakte verloren gehen, weil das Hauptaugenmerk darauf liegt, Produkte möglichst schnell auf den Markt zu bringen. Das Resultat davon ist, dass Informationen bezüglich des Produkts häufig nur in Form des finalen Ergebnisses vorliegen, das wiederum aus den Entwurfsentscheidungen besteht, die im Produkt umgesetzt wurden. Der Weg, wie man zu diesem Ergebnis kam, beispielsweise der Verlauf der Studien, Prototypen, Experimente, Argumente und der Prozess der Entscheidungsfindung, der zu dem Produkt führte, wird zwar häufig nicht absichtlich verworfen, geht aber dennoch zumeist unter und ist schwer zu durchschauen, besonders für Personen, die nicht am ursprünglichen Projekt beteiligt waren.

Wiederverwendung aber bedeutet eine umfassende Sichtung auch der Weiterentwicklung der Dokumente. Per Definition entspricht ein neues Produkt nicht exakt einem bereits existierenden Produkt; um also die Unterschiede zu entwickeln, muss man sich den Gedankenverlauf präzise vor Augen führen, der zur Auswahl oder zur Ablehnung von Entwurfsoptionen führte. Für die Weiterentwicklung von Produkten, die ja durch Wiederverwendung unterstützt werden soll, muss eine erneute Evaluierung im Licht sich geänderter Situationen, neuer Belege, anderer Einschränkungen oder Ziele oder fortgeschrittener Technologie der getroffenen Entscheidungen stattfinden.

Daher sind wir der Meinung, dass Wiederverwendung eine bewusste Erweiterung der

Rückverfolgbarkeit von Anforderungen in die Argumentation erfordert, die zu den getroffenen Entwurfsentscheidungen führte. Natürlich spielt diese Art der Rückverfolgbarkeit vor allem bei Innovationen oder Produkten mit innovativen Anteilen eine essentielle Rolle, da im Falle von etablierten Produkten die zugrunde liegende Argumentation den beteiligten Personen klar ist und sich Auswirkungen von Weiterentwicklungen in der Technologie leichter auf vorhandene Features des Produktes abbilden lassen. An dieser Stelle gibt es also eine Verbindung zwischen Innovationen und Wiederverwendung, die in dieser Form in der wissenschaftlichen Literatur zu diesem Thema kaum untersucht wurde. Da mit Software-Produktlinien [CN02] ein erfolgreicher und systematischer Wiederverwendungsansatz beschrieben wurde, schlagen wir vor, dass die benannte Art der Rückverfolgbarkeit integraler Bestandteil der Produktliniendokumentation sein muss. Da diese Information lange genutzt wird (möglicherweise über mehrere Jahre), ist die Unterstützung durch ein Datenbankwerkzeug unverzichtbar.

1958 argumentierte Stephen Toulmin überzeugend, dass konventionelle (aristotelische) Beschreibungen von Argumentation eine (mathematische oder logische) Beweisebene erforderten, die in der Praxis so gut wie nie vorhanden waren. Aristoteles argumentiert mit logischer Stringenz und kommt so zu einem Vernunftschluss: Alle Menschen sind sterblich; Sokrates ist ein Mensch; also ist Sokrates sterblich. Toulmin schlägt eine andere Struktur der Argumentation vor, wie sie in Abbildung 1 gezeigt wird.

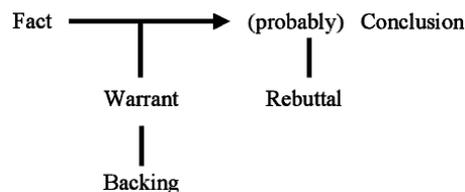


Abbildung 1: Toulmin Argumentation.

Ein mögliches Toulmin Argument wäre: „Aktuelle Dieselmotoren sind groß, schmutzig und laut (*Warrant*). Ein Dieselmotor ist vorgeschlagen für unser neues Auto (*Fact*). Daher wird unser neues Auto wohl groß, schmutzig und laut sein (*Conclusion*).“ Eine ähnliche Überlegung könnte zu folgender Schlussfolgerung führen: „... und ist somit nicht angebracht für den anvisierten Markt.“ Nun ist solch ein Argument niemals beweiskräftig, da die Zusicherung (*Warrant*) nur vermutlich zutrifft; durch neue Belege könnte sie untergraben werden, wie beispielsweise durch die Entwicklung von kleinen, saubereren, leisen Dieselmotoren (*Rebuttal*).

Man sieht also, dass das Schlüsselement der Toulmin Argumentation die (ungewisse) Vermutung (*Assumption*) ist, die den Platz der logischen Zusicherung („alle Menschen sind sterblich“) in einem aristotelischen Argument einnimmt. Jemand nimmt an, dass alle Diesel schmutzig sind; dies kann falsch sein, wenn ein Fakt etwas anderes aufzeigt. Tatsächlich können Vermutungen im Extremfall an die Stelle aller fünf von Toulmin vorgeschlagenen Argumentationselemente treten: *Facts*, *Warrants*, *Backings*, *Conclusions* und *Rebuttals* können alle als Vermutungen unterschiedlicher Art behandelt werden. Daher basieren die von uns vorgeschlagenen Entscheidungsmodelle auf Vermutungen und Belegen, wenn diese mehr oder weniger unwiderlegbare Fakten sind.

Teile einer vorhandenen Argumentation sind häufig nützlich, um besondere Vermutungen zu unterstützen. So veranlassen die Ziele (*Goals*) einer Organisation die Generierung von Vermutungen, die wichtig sind für den Entwurf eines Produkts. Weiterhin können praktische

Einschränkungen (*Constraints*) bezüglich Zeit, Geld und Kenntnissen die Fähigkeiten einer Organisation einschränken, risikoreiche Optionen erfolgreich umzusetzen. Einschränkungen können daher zu Vermutungen führen wie: „Wir können keinen neuen Motor vor dem Jahr 1965 entwickeln“, die verwendet werden können, um optimistische Vermutungen darüber zu entkräften, was wir tun können. Somit können Ziele und Einschränkungen auf ihre unterschiedliche Art und Weise mit Vermutungen positiv oder negativ interagieren. Nicht zuletzt haben wir in unser Modell das Element „Hinweisschild“ („*Signpost*“) mit aufgenommen, dass von James Dewar in seinem „*Assumption-Based Planning*“ Ansatz vorgeschlagen wird [Dew02]. Das Hinweisschild beschreibt ein Ereignis, unter dem eine Vermutung aufgehoben wird. Dewar unterscheidet anfällige (wahrscheinlich, dass sie aufgehoben wird) und tragende (kritisch für die zu treffende Entscheidung) Vermutungen als solche, die mit einem Hinweisschild versehen werden sollen.

Wir sind der Meinung, dass Dewars Vorschlag ebenso auf den Bereich der Entwurfsentscheidungen für Software-Produkte wie für Management Planungsentscheidungen anwendbar ist, wofür er es selbst konzipiert hatte. Daher erlauben wir, dass die Vermutungen als anfällig oder tragend attribuiert und sie um Hinweisschilder ergänzt werden dürfen, um das Überprüfen von Entwurfsentscheidungen im Bedarfsfall anzustoßen. Dies ist entscheidend, da es direkt die Wiederverwendung unterstützt. Wenn ein Argument, das besagt, dass eine Entwurfsoption undurchführbar ist, durch ein Hinweisschild als wahrscheinlich aufzuheben beschrieben wird, wird es Zeit, das in Frage stehende Argument zu überprüfen, um zu analysieren, ob die Entwurfsoption nun nutzbar ist. Dieser dynamische Aspekt in der Wiederverwendung scheint in der existierenden Forschung zu diesem Thema nicht beachtet worden zu sein. Statische Nachvollziehbarkeit zu früheren Entscheidungen ist nützlich, da es einen in die Lage versetzt, die Argumentation zu überprüfen; dennoch ist dies nicht genug, da nicht aufgezeigt wird, wann eine erneute Überprüfung der Sachlage lohnenswert ist. Dewars Hinweisschilder erfüllen diese vitale Rolle und flexibilisieren damit ein durchdachtes Argumentationsnetzwerk in Bezug auf zukünftige Entwicklungen, womit auch Informationen, die bei der Erstellung von Innovationen gesammelt werden, in den Wiederverwendungsprozess eingebracht werden können, ohne zu einem Bruch im Entscheidungsnetzwerk zu führen; man hat also ein Werkzeug in der Hand, mit dem man bei der Entwicklung einer Innovation, produktfamilien-orientierte Entscheidungen trifft.

Elemente des Entscheidungsmodells

Die vorgeschlagene Struktur eines Entscheidungsmodells ist als ein abstraktes Meta-Modell in Abbildung 2 illustriert. Es ist unabhängig von einer speziellen Implementierung. Jede Box in der Abbildung repräsentiert eine Klasse von Teilen einer Argumentation. Jeder Pfeil repräsentiert einen Beziehungstyp zwischen einem Paar dieser Klassen. Eine Implementierung eines Entscheidungsmodells als eine Datenbankarchitektur in dem Anforderungsverwaltungswerkzeug DOORS [Tel] ist als ein konkretes Meta-Modell in Abbildung 3 illustriert.

Die Struktur erlaubt die Darstellung eines Netzwerks von Vermutungen, wobei einige eine spezielle Entwurfsentscheidung unterstützen und andere schwächen (wie durch die beiden internen Beziehungen von Vermutungen beschrieben). Die Entwurfsentscheidung selbst ist Teil der (möglichen) Charakteristiken der Innovation, die im Zuge der Entwicklung der Innovation als realistisch eingeschätzt, aber möglicherweise für einen späteren Zeitpunkt zurückgestellt wurde. Ein Entscheidungsmodell besteht typischerweise aus einem Netzwerk von Vermutungen in einer Baumstruktur, die eine einzige Entwurfsentscheidung unterstützen

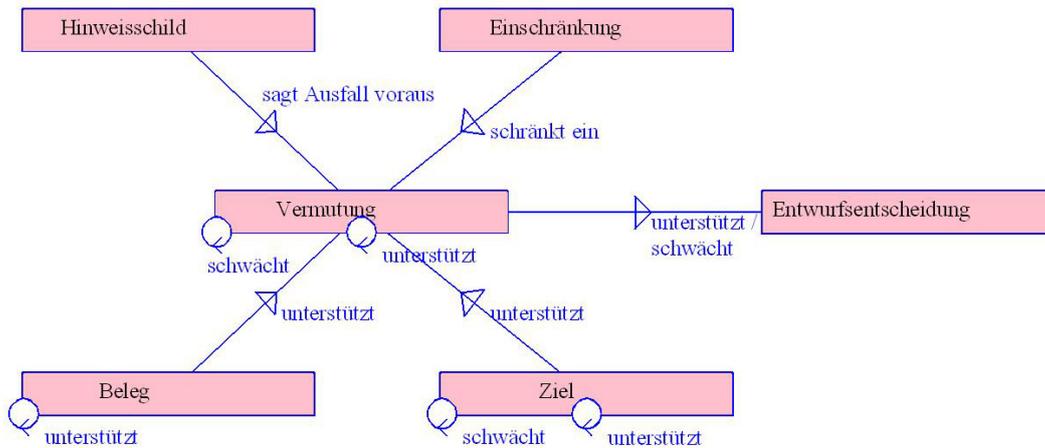


Abbildung 2: Meta-Modell eines Entscheidungsmodells.

oder schwächen. Alternative (oder weitere) Entwurfsentscheidungen werden normalerweise in einem eigenen Entscheidungsmodell diskutiert. Wenn eine Entwurfsentscheidung aus einem Baum an Teilcharakteristiken besteht, so bedeutet die Auswahl dieser Entwurfsentscheidung die Auswahl des gesamten Baumes.

Die verbleibenden Teile der Struktur stellen weitere Erläuterungen bereit, entweder in der Form von Teilen von Belegen, die konkrete Unterstützung für spezifische Vermutungen darstellen, oder in Form eines Netzwerks von Zielen, die weitere Unterstützung bieten. Ziele sind die Dinge, die das Management beabsichtigt zu erreichen, mithin eine regelmäßig zu erörternde Unterstützung einiger Vermutungen über die Domäne. Weiterhin kann sich das Management über gewisse Einschränkungen bewusst sein (auch über die Möglichkeit des Erreichens dieser Ziele, was aber nicht in diesem Rahmen diskutiert wird), die einige Vermutungen unterstützen, beispielsweise, dass eine Entwurfsentscheidung zu kostenintensiv zu pflegen wäre. Sobald eine Vermutung explizit gemacht wurde, kann man sie mit einem Risiko in Beziehung setzen: Die Vermutung könnte nicht zutreffen. Diese Risiken können entschärft werden durch die Identifikation von passenden Hinweisschildern, die den Ausfall der assoziierten Vermutungen vorhersagen (typischerweise eine 1:1 Beziehung).

Das gesamte abstrakte Meta-Modell in Abbildung 2 wurde in einer einfachen DOORS-Architektur implementiert (Abbildung 3). Jede Box in der Abbildung repräsentiert eine Klasse einer DOORS-Datenstruktur mit (in Klammern) der Klasse an abstrakter Information, die darin enthalten ist. Jeder Pfeil repräsentiert eine konkrete Beziehung in der Datenbank. Die Beziehung „beinhaltet“ bezieht sich auf die in DOORS zur Verfügung stehende Objekthierarchie, die typischerweise als eine Hierarchie nummerierter Abschnitte (wie 1, 1.1, 1.2, etcetera) dargestellt wird. „Interagiert mit“ bezieht sich auf einen *Linkset*, also einer DOORS-internen Struktur zur Abbildung von Objekt-zu-Objekt Links.

Alle textuellen Daten werden in einem *DOORS Formal Module* gehalten, das entfernt vergleichbar ist mit einem Dokument, das eine Hierarchie an Überschriften und Paragraphen enthält. Die eingebaute Hierarchie der DOORS-Objekte in diesem formalen Modul wird verwendet, um eine Menge an Entscheidungsmodellen (ein DOORS-Level 1 Objekt, korrespondiert zu einem Dokumentkapitel) und die Teile der Argumentation von irgendeinem Typ zu repräsentieren (Vermutung, Ziel, Einschränkung, Beleg, Hinweisschild), die sie enthalten

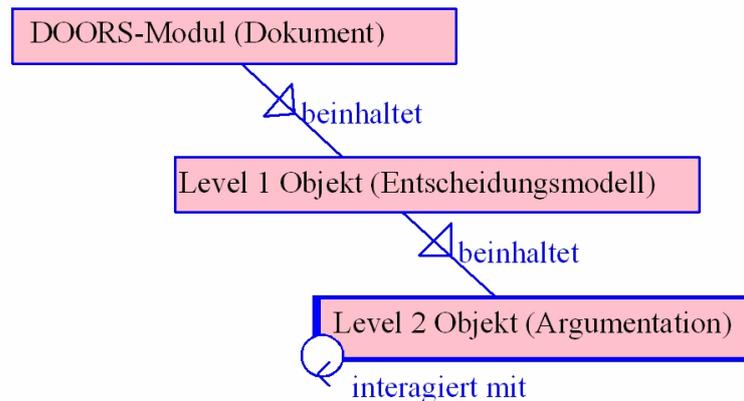


Abbildung 3: Meta-Modell der DOORS Implementierung.

(eine Liste von DOORS-Level 2 Objekten, korrespondiert zu einem Paragraphen). Die Beziehungen zwischen Teilen der Argumentation werden repräsentiert als DOORS Links des Typs „interagiert mit“, mit einem Link Attribut, um die tatsächliche Beziehung festzuhalten (unterstützt, schwächt, etcetera). Kapselung wird direkt durch Verwendung der strukturellen Hierarchie der Objekte in DOORS repräsentiert.

Mehr als ein solches Entscheidungsmodellmodul kann in einem gegebenen Datenbankordner erstellt werden, wenn dies gewünscht ist, obgleich eine beliebige Anzahl an Entscheidungsmodellen in einem Modul definiert werden kann, was nicht notwendig sein sollte, bis auf den Fall, dass eine Unterteilung in Gruppen von Modellen gewünscht ist. Eine beliebige Anzahl von Nachvollziehbarkeitlinks kann zu oder von anderen Modulen erstellt werden. Diese Links können bearbeitet und analysiert werden, womit die gesamten Möglichkeiten der Datenbank angewendet werden können, bis auf die Ebene eines einzelnen Teils der Argumentation. Ein grafischer Editor zur Konstruktion und Modifikation solcher Entscheidungsmodelle wurde erstellt (Abbildung 4). Der Editor ist frei verfügbar von <http://www.szenarioplus.org.uk>.

Um die Anwendung der Entscheidungsmodelle im Automobilbereich zu illustrieren, stellen wir uns vor, wir arbeiten für Ruritanian Autos im Jahr 1963. Wir wünschen uns ein innovatives, kleines Auto mit einem begrenzten Budget zu produzieren. Wir könnten einen innovativen Motor als ein attraktives Feature einführen, aber welchen? Abbildung 5 zeigt das top-level Entscheidungsmodell. Es indiziert eine Menge an detaillierteren Modellen, wovon eines in Abbildung 4 gezeigt ist (was in der Entscheidung mündet, dass die Entwicklung eines kleinen, sauberen und leisen Dieselmotors zu kostenintensiv für die Firma im Jahre 1963 wäre).

Diese zeigen die prinzipiellen Charakteristiken eines Entscheidungsmodells, nämlich Ketten von Vermutungen, die zu einem Ergebnis/einer Entscheidung führen (der Bedarf für einen neuen Motor), unterstützt durch Ziele und beschränkt durch einige praktische Einschränkungen. Einige der Vermutungen scheinen sowohl tragend als auch verletzlich zu sein und diese sind letztlich abhängig von nur zwei Vermutungen: „Öl ist billig“ und „Arbeitslosigkeit ist gering“. Diese beiden werden daher mit einem Hinweisschild versehen, um den Automobilhersteller dazu anzuregen, seine Entscheidung von neuem zu evaluieren, wenn eine von beiden ausfallen sollte. Die Entscheidung könnte durch jede der fünf alternativen Ent-

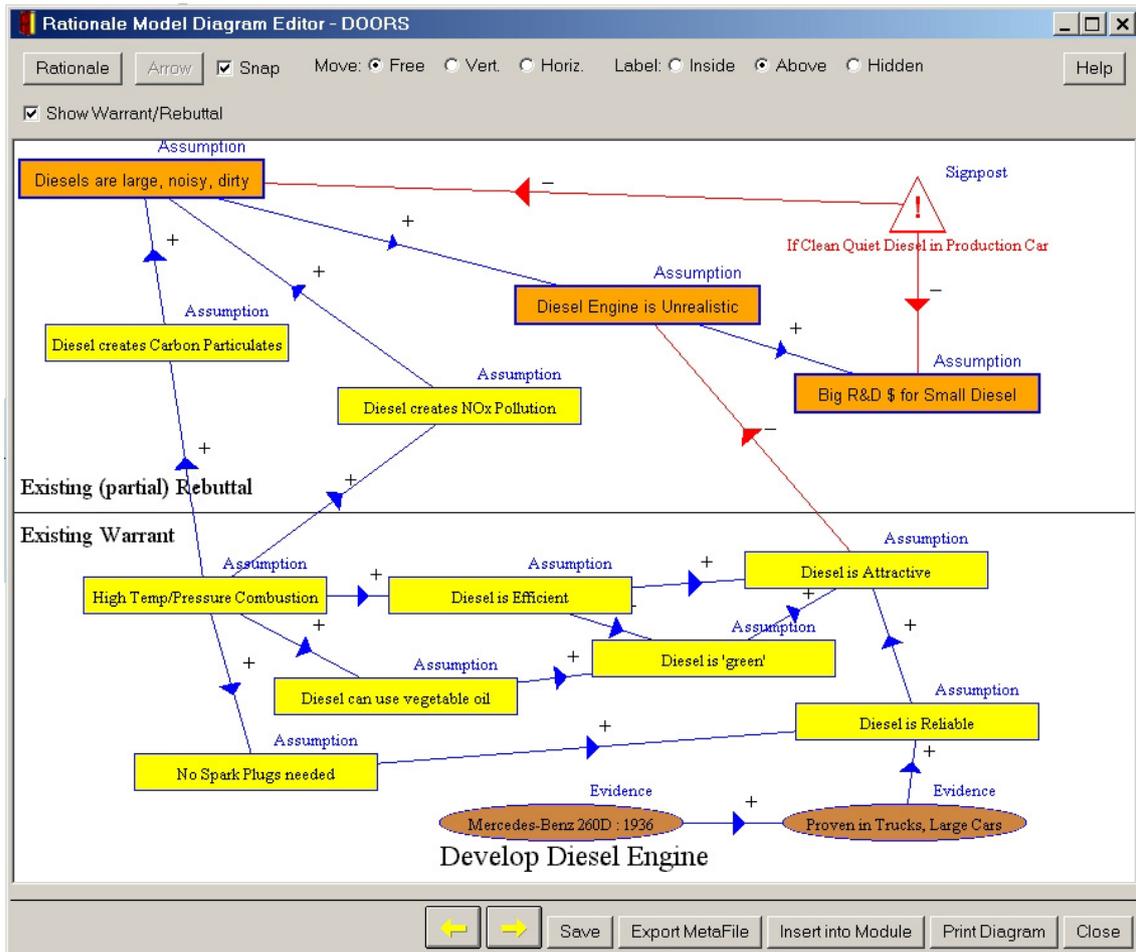
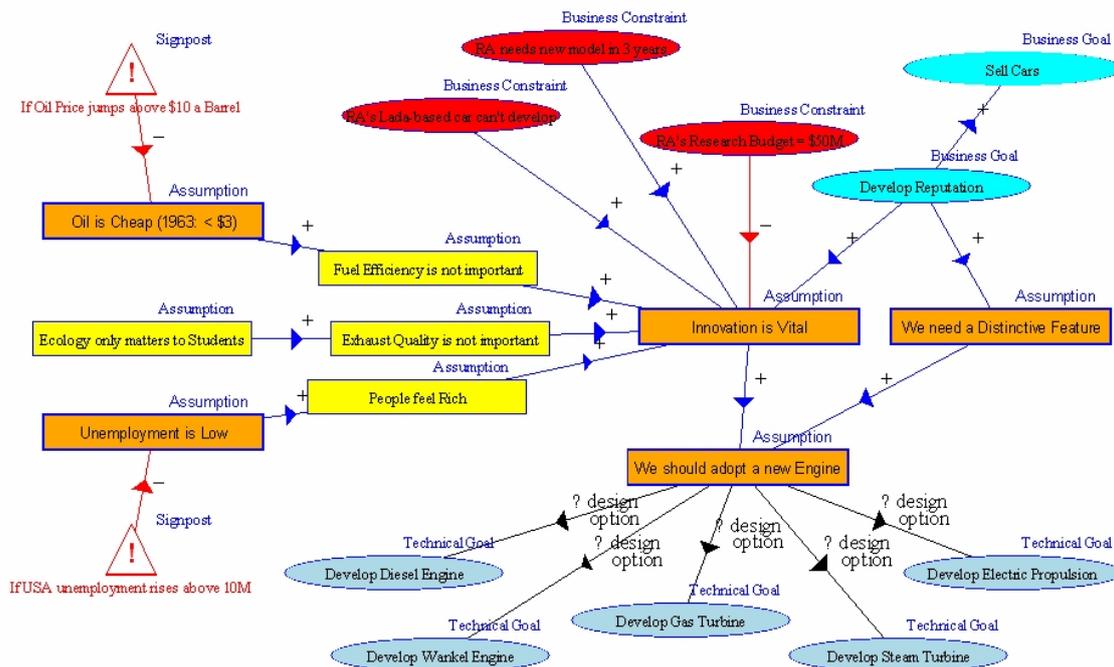


Abbildung 4: Der DOORS Editor für Entscheidungsmodelle.

wurfsentscheidungen umgesetzt werden. Jede dieser Entwurfsentscheidungen ist Thema eines Untermodells, wie das zur Entwicklung eines Dieselmotors in Abbildung 4. Diese Illustration zeigt die Division des Diagramms in eine positive (Toulmin „warrant“) und negative (Toulmin „rebuttal“) Region, wobei jede hauptsächlich aus einer Kette von Gründen, repräsentiert durch Vermutungen, besteht, obgleich relevante Belege auch gegeben sind.

2.3 Werkzeug Statemate

Für die Modellierung und die Simulation eingebetteter Systeme haben sich eine Reihe von Werkzeugen am Markt etabliert (beispielsweise I-Logix Statemate [I-L] oder auch Telelogic DOORS/Analyst [Tel]). In unserem Praktikum haben wir uns für das Werkzeug Statemate entschieden. Dieses Werkzeug wird von vielen Unternehmen im Bereich der eingebetteten Systeme für den Entwurf und das Testen verwendet, insbesondere auch im Automobil-Bereich. Hierarchische Zustandsautomaten (Statecharts) bilden das zentrale Modellierungselement. Sie erweitern einfache Zustandsautomaten um die Möglichkeit der Parallelisierung und der Hierarchisierung und ermöglichen dadurch eine effizientere und elegantere Modellierung, indem



The 1963 Scenario: Ruritanian Autos (RA) Assumptions and Rationale

Abbildung 5: Das Top-Level Entscheidungsmodell für das Szenario von Ruritanian Auto.

Zustände und Zustandsübergänge eingespart werden können. Diese Zustandsautomaten sind außerdem mit einer ausführbaren Semantik hinterlegt, so dass erstellte Diagramme simuliert werden können.

Statemate bietet dafür eine gut bedienbare Simulationsumgebung: der Benutzer hat die Möglichkeit, so genannte Panels (graphische Oberflächen) zu erstellen, mit den Statecharts zu verknüpfen und hierdurch eine prototypische Benutzeroberfläche zu erstellen. In Verbindung mit den zugrunde liegenden Statecharts bietet diese Simulationsumgebung eine hervorragende Möglichkeit, ein System zu testen und Fehler frühzeitig zu entdecken. Es ist möglich, auf spielerische Art die erstellte Spezifikation auf Korrektheit und Vollständigkeit aufbauend auf Testfällen zu untersuchen. Neben diesen Zustandsautomaten bietet Statemate sogenannte „Activitycharts“ (nicht zu verwechseln mit UML 2 Aktivitätsdiagrammen!) an, um die funktionale Sicht im System-Kontext, also die System-Architektur, darzustellen.

Wir haben mit diesem Werkzeug positive Erfahrungen in vorangegangenen Praktika gemacht: Obwohl die Studierenden das Werkzeug selber anfangs nicht kennen, so kennen sie dennoch Statecharts [Har87, HN96] (die die wesentliche Modellierungsmethode dieses Werkzeugs darstellen) von ihrem Grundstudium, so dass die Einführungsphase in das Werkzeug typischerweise recht kurz ist (üblicherweise benötigen wir eine vierwöchige Einführungsphase in dem Praktikum, um neben Statemate die zu verwendenden Prozesse und Methoden zu erklären und mit Übungen zu vertiefen). Obwohl Statemate normalerweise zur Erstellung von Spezifikationen eingesetzt wird, waren die erstellten Statecharts in unserem Setting außerdem bereits das fertige Produkt. Mit diesem Werkzeug ist es für die Studierenden möglich, in kurzer Zeit lauffähige Modelle zu erstellen, ohne zu viel Zeit in Programmierung und Fehlersuche

zu investieren. Erfahrungen aus vorherigen Praktika zeigen, dass Studierende Modellierung tatsächlich der textuellen Beschreibung von Anforderungen oder der Implementierung vorziehen.

Ein weiterer Grund für den Einsatz dieses Werkzeugs ist, dass keiner der teilnehmenden Studierenden Erfahrungen damit hat. Dadurch konnten die größten Unterschiede bezüglich des Vorwissens der Teilnehmer in Hinblick auf das Experiment ausgeglichen werden. Dies verhindert auch eine übermäßige Frustration der Studierenden, die mit der Programmierung im Allgemeinen und speziell von eingebetteten Systemen noch nicht so vertraut sind.

2.4 Ziel und Szenario des Experiments

Im folgenden Abschnitt geben wir einen Überblick über die Ziele und Fragestellungen des Experiments, über qualitatives Forschen („Action Research“) und über das dem Experiment zugrunde liegende Szenario. Die Gestaltung des Experiments und dessen praktische Umsetzung ist motiviert durch die hier beschriebenen Ziele und Fragestellungen, die schon konkret die Art und Weise der Auswertung vorweg nehmen.

Ziele und Fragestellungen des Experiments

Betrachtet man klassisches Vorgehen bei der Entwicklung von Innovationen [PTK04], so stellt man fest, dass die Entwickler eine intensive Recherche der Domäne betreiben, in der die Innovation angesiedelt ist; diese Recherche fällt dann besonders intensiv aus, wenn mit der Entwicklung der Innovation auch die Entwicklung der Domäne selbst, in der die Innovation eingebettet ist, verbunden ist, wie es der Fall ist bei technischen Innovationen, die es bis zu diesem Zeitpunkt noch gar nicht gab. Bei der Recherche wird viel Wissen über das künftige Produkt angesammelt, aber auch Wissen über die Weiterentwicklung des Produktes. So ist es häufig möglich, bis zu fünf Jahre in die Zukunft der Innovation zu schauen und Weiterentwicklungen vorweg zu nehmen. Dieses Wissen ist sehr wertvoll, da es für die Gestaltung des Produktes als ein wiederverwendbares und leicht erweiterbares Konstrukt genutzt werden kann. Somit lässt sich, so sind wir der Meinung, die Hürde der hohen Investitionskosten bei der Einführung von Software-Produktlinien [CN02] senken, da ein gewisses *Domain Engineering* vorweggenommen wurde: In der Literatur zum Thema Software-Produktlinien wird ein solches Domain Engineering, das in die Zukunft blickt, proaktives Domain Engineering genannt, im Gegensatz zum reaktiven Domain Engineering, das im Nachhinein auf Basis einer ganzen Reihe an Produkten Variabilität und Gemeinsamkeit extrahiert. Das Problem in der Praxis ist in diesem Zusammenhang, dass das mühsam erarbeitete Wissen verloren geht, zugunsten einer schnellen und machbaren Ein-Produkt-Lösung. Das Wissen ist bestenfalls in Projektprotokollen dokumentiert, die allerdings nach Abschluss eines Projekts und besonders für diejenigen, die nicht am Projekt beteiligt waren, praktisch nicht auffindbar sind.

Wir schlagen keinen revolutionären Ansatz vor: So fordern wir beispielsweise nicht, dass man sämtliche Erweiterungen und Variationen, die mit dem Produkt verbunden sind, von Anfang an in der Produktentwicklung betrachtet und auf allen Artefaktebenen adäquat vorhält. Wir schlagen vielmehr einen „sanften“ Weg vor, allein für den Fall, dass das erarbeitete Wissen nicht verloren gehen soll. Dabei setzen wir bei der Dokumentation an: Die textuelle Dokumentation in den Projektprotokollen, in denen sehr viel mehr steht, als künftige Erweiterungen des Produktes, die nicht nach inhaltlichen sondern vielmehr nach zeitlichen Gesichtspunkten sortiert sind und die — was das größte Problem darstellt — nach Abschluss

des Projekts praktisch nicht mehr zur Verfügung stehen, soll ersetzt werden durch die grafische Notation der oben vorgestellten Entscheidungsmodelle. Diese können in DOORS, also in dem Werkzeug, in dem auch die Spezifikationen beschrieben sind, vorgehalten werden und sind somit weniger gefährdet, verloren zu gehen. Des Weiteren sind sie übersichtlich, haben einen klaren inhaltlichen Fokus auf die Erweiterung und Variation des Produktes und sind intuitiv zugänglich. Darüber hinaus stellen sie auch kaum einen Mehraufwand dar, da das in ihnen abgebildete Wissen vorhanden ist und andernfalls textuell beschrieben wurde und es damit zu keinen Änderungen in den Prozessschritten kommt.

Interessant war für uns nun die Frage, ob die vorgeschlagenen Entscheidungsmodelle tatsächlich in der von uns vorgestellten Art und Weise die Hinführung der Innovation in den Rahmen einer Software-Produktlinie unterstützen. Zu beachten ist hierbei, dass wir den von uns vorgestellten Ansatz mit den Entscheidungsmodellen nicht mit einem anderen, möglicherweise auch argumentationsorientiertem Ansatz vergleichen: Erstens ist uns nach eingehender Sichtung der wissenschaftlichen Literatur zu diesem Thema kein anderes Modell, das einen ähnlichen Ansatz verfolgt, bekannt. Wir hätten also das von uns vorgeschlagene Entscheidungsmodell mit Modellen vergleichen müssen, die als Vorläufer des Modells anzusehen sind (Toulmin [Tou58] und Dewar [Dew02]), was keinen Sinn macht. Zweitens wollen wir mit dem Experiment nicht klären, *welches* Argumentationsmodell besser geeignet ist für die Unterstützung des proaktiven Domain Engineering bei Innovationen, sondern *ob* ein Argumentationsmodell dafür besser geeignet ist als die momentane Praxis in der Entwicklung (Dokumentation des Wissens in Projektprotokollen). Um diese Frage zu klären, ist zunächst zu betrachten, in wie weit die Entscheidungsmodelle intuitiv anwendbar sind und ob der Erstellungsaufwand durch sie steigt. Neben der Untersuchung dieser technischen Fragestellungen hat uns auch die Frage interessiert, ob das so dokumentierte Wissen auch das Verständnis des Produktes und seiner Varianten und somit der Domäne unterstützt. Weitere Fragen, die in diesem Zusammenhang mit auftreten, die allerdings keine zentrale Rolle gespielt haben, war die Qualität, Vollständigkeit und Kundenorientierung der erstellten Artefakte; dies spielte eine weniger zentrale Rolle, weil wir nicht unterschiedliche Entwicklungs*methodiken* (die Entwicklungsmethodik war für alle Gruppen in allen Iterationen gleich), sondern nur einen Unterschied in der Dokumentation eines Sachverhalts untersuchten. Natürlich kann man hier einwenden, dass diese unterschiedliche Dokumentation durchaus einen Einfluss auf die Qualität und Vollständigkeit der zu erstellenden Artefakte haben kann, da die Studierenden mit unterschiedlicher Motivation an innovative Herangehensweisen gehen als an bereits bekanntes. Trotzdem dieser Einwand nicht von der Hand zu weisen ist, haben wir den Eindruck, der auch von den Angaben der Studierenden in regelmäßig stattfindenden Interviews bestätigt wurde, dass sich die Motivation der Studierenden in den unterschiedlichen Gruppen nicht besonders unterschied. Vielmehr war für alle Studierenden praktisch alles, was sie in diesem Praktikum lernten, neu und damit sah sich weder die eine noch die andere Gruppe benachteiligt. Was die Übertragbarkeit der Ergebnisse des Experiments auf ein anderes Umfeld angeht (externe Validität, [Pre01]), so muss man grundsätzlich vorsichtig sein, wenn man ein Experiment in universitärem Rahmen durchführt. Natürlich gibt es Vorbehalte von Entwicklern, neue Vorgehensweisen umzusetzen, wenn man mit den bisherigen Ansätzen schon erfolgreich war. Besondere Skepsis wird vor allem den Ansätzen entgegengebracht, die Mehraufwand oder geänderte Entwicklungsmethodik in einem Projekt mit sich bringen und ein anderes Projekt dadurch Vorteile hat (was hier, wie so häufig, wenn man Wiederverwendungsansätze betrachtet, der Fall ist); in diesem Fall muss vor allem die Angst vor dem neuen Ansatz genommen werden, die Vorteile müssen klar ersichtlich sein und er sollte nicht oder nur mit

sehr wenig Mehraufwand verbunden sein. Die externe Validität des Experiments werden wir in Abschnitt 5 ausführlich diskutieren, aber auch schon bei dem Entwurf eines Experiments ist diese zu beachten, da sie neben einer möglichst realitätsnahen Gestaltung des Experiments auch die zu erhebenden Daten beeinflusst. Auf die Gestaltung des Experiments sowie dessen Nähe zur aktuellen Entwicklungsrealität, soweit dies in dem universitären Rahmen möglich ist, gehen wir in Abschnitt 3 ein. Was durch die Betrachtung der externen Validität als Frage zu den bereits beschriebenen hinzukommt, ist die nach der subjektiven Einschätzung der Studierenden. Die Studierenden agieren nicht nur als Probanden, sondern sind auch Forscher und Entwickler, die nun Erfahrungen sammeln müssen mit neuen Vorgehensmodellen. Sie können Aussagen machen zu den Herangehensweisen, Forderungen und Verbesserungsvorschläge einbringen und über die eigene Arbeit reflektieren. Diese Aussagen lassen sich nur in narrativer Form darstellen, sind also nicht in Schaubildern darzustellen. Dennoch sind dies wichtige Informationen, die gerade zur Betrachtung der externen Validität beitragen. Genaueres zu diesem Teil des Experimententwurfs, der qualitative Teil des Experiments, wird in dem folgenden Unterabschnitt genauer beschrieben.

Zusammenfassend haben wir folgende Fragen mit diesem Experiment behandelt:

1. *Intuitivität und Aufwand der beiden Dokumentationsarten*: Wie intuitiv sind die Dokumentationsarten? Wie leicht lassen sich mit ihnen Sachverhalte darstellen, wie schnell kann man sich in ein solches Dokument einarbeiten und wie hoch ist der entsprechende Aufwand?

Methode: Datenauswertung.

2. *Verständnis der Varianten der erstellten Innovation*: Wie ist das Verständnis der Gruppe bezüglich der erstellten Innovation und deren Varianten im Verlauf der Weiterentwicklung?

Methode: Merkmaldiagramm und Anforderungsdokumente, Projektprotokoll und Entscheidungsmodell.

3. *Qualität und Vollständigkeit der erstellten Artefakte*: Wie ist die Qualität der erstellten Artefakte, vor allem State-Mate-Modelle und Testfälle? Diese Frage ist nur am Rande interessant, da wir nicht unterschiedliche Entwicklungsmethodiken verglichen haben, sondern unterschiedliche Dokumentationsmethoden für die Darstellung des generierten Wissens. Dennoch ist diese Frage in Bezug auf die externe Validität notwendig zu betrachten, da gegebenenfalls vorhandene Unterschiede in der Motivation ablesbar sind.

Methode: Einschätzung eines Experten.

4. *Reaktion und Reflektion der Studierenden*: Wie haben die Studierenden selbst auf ihre Dokumentationsmethode reagiert? Welche Verbesserungsvorschläge haben sie gegeben?

Methode: Auswertung der Ergebnisse der regelmäßigen Interviews.

Action Research

Qualitative Forschung oder „*Action Research*“ [Par01, LG01, HR01, Sch01, Tor01, Bru01, Fli02, MW03] ist eine Art Forschung, die eine Reflektion des eigenen Lernens als zentralen Baustein umfasst. In [Sch01] wird *Action Research* charakterisiert als eine Forschung, die sich sowohl durch eine intensive Einbindung des Forschers als auch des Probanden auszeichnet.

Dies trifft in besonderem Maße auf die Experimente zu, die wir in Form eines Praktikums bei der Universität Ulm durchführen. Die Studierenden sind zwar Probanden in einem Experiment, nehmen aber eine aktive und gestaltende Aufgabe wahr, da sie sich mit neuen Methoden befassen und damit gegebene Probleme lösen müssen. Sie sind gehalten, sich mit den Methoden zu beschäftigen und gegebenenfalls Empfehlungen oder Verbesserungsvorschläge abzugeben, die bei gegebenen Möglichkeiten auch umgesetzt werden. Damit agieren die Studierenden nicht nur in der Rolle von Entwicklern, sondern auch in der Rolle von Forschern, die sich mit den gegebenen Methoden auseinandersetzen und Verbesserungen vorschlagen; allerdings resultieren diese Verbesserungsvorschläge nicht aufgrund theoretischer Überlegungen, sondern aufgrund intensiver Beschäftigung und Arbeit mit den Methoden, die teilweise zu konstruktiver Unzufriedenheit führt, aus der wiederum Verbesserungsvorschläge resultieren.

Weiß man, dass man in einem Experiment auch mit einer intensiven Einbindung der Probanden zu rechnen hat (was man natürlich auch entsprechend gestalten kann), so muss man ein Instrumentarium bereit halten, was einen bei der Einbindung der Probanden unterstützt. Dafür werden eine ganze Reihe von Herangehensweisen im *Action Research* vorgeschlagen. Lewin [Lew46] entwickelte im Jahr 1946 eine Theorie des *Action Research* als eine Spirale von Stufen die Planung, Erkundung und Ausführung umfasste. Diese wurde später gemeinhin als Aktion-Reflektion-Kreis verstanden, der Planung, Aktion, Beobachtung und Reflektion umfasste.

Wir haben in Anlehnung an diesen Kreis ein Modell angewendet, das unserer Meinung nach noch besser den Anforderungen unseres Experiments gerecht wird (vgl. Abbildung 6). Da wir uns im Rahmen des Experiments regelmäßig einmal in der Woche in der großen Gruppe getroffen haben, mussten wir diesen Termin nutzen, das gesamte Spektrum an Erfahrung und Ideen, die die Studierenden bei ihrer Arbeit in der Woche angesammelt haben, abzuschöpfen. Dies umfasste nicht nur die Reflektion, also das Überdenken der Arbeiten und Methoden und gegebenenfalls die Formulierung von Verbesserungsvorschlägen, sondern auch die emotionale Reaktion auf diese Arbeiten. Umgesetzt haben wir dieses Modell in wöchentlich stattfindenden Interviews, wobei die beiden Großgruppen, die jeweils eine andere Form der Dokumentation des Domänenwissens verfolgten, getrennt befragt wurden. Die Fragen zu den Interviews wurden jeweils vor dem Treffen von den Betreuern festgelegt und waren an den speziellen Arbeiten der Studierenden ausgerichtet.

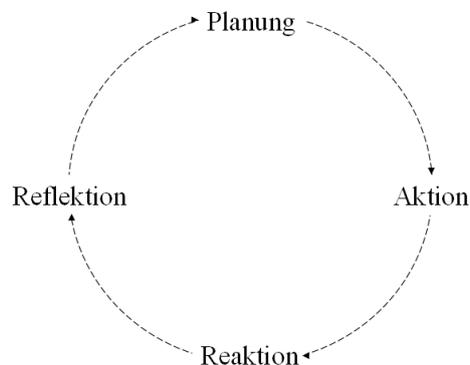


Abbildung 6: Der für das Experiment angepasste Aktion-Reflektion-Kreis.

Die *Planung* wird von den Betreuern vor Beginn des Experiments durchgeführt und mit den Studierenden an einem Termin in der Woche durchgesprochen. Die *Aktion* wird von den Studierenden in der anschließenden Arbeitswoche durchgeführt. Der Termin im Anschluss an diese Woche wird genutzt, um die *Reaktion* und *Reflektion* der Studierenden zu dieser Arbeitswoche zu ermitteln. Wichtig in diesem Zusammenhang ist, dass die Studierenden regelmäßig einmal in der Woche befragt werden, da andernfalls Ideen aber vor allem Emotionen verloren gehen, die für die Auswertung des Experiments notwendig gewesen wären. Weiterhin haben wir die Idee verworfen, diese Fragen mit den anderen Daten, die die Studierenden über das Netz eingeben konnten, zu stellen, da wir auch Diskussion und Austausch zwischen den Studierenden anregen wollten. Der Kreis beginnt an dem Termin der Besprechung der Reaktion und Reflektion von neuem mit der Besprechung der Planung für die nächste Arbeitswoche.

Wir haben diesen Aktion-Reflektion-Kreis als wichtige Ergänzung der sonstigen Daten, die die Studierenden regelmäßig abgeben mussten, verwendet. Dies liegt zum einen daran, dass sich das Experiment, wie oben geschildert, gut dafür eignet, die Probanden gewissermaßen in die Forschertätigkeit einzubeziehen und zum zweiten, um die Motivation und das Verständnis der Studierenden bezüglich diesem Experiment und der zugrunde liegenden Fragestellung zu erhöhen.

Die aus den Interviews gewonnenen Erkenntnisse werden im Abschnitt 5 besprochen. Die Ergebnisse des *Action Research* sind aber die vollständige Beschreibung der Interviewergebnisse, so dass wir die vollständigen Fragen und Antworten im Anhang mit abgedruckt haben.

Szenario

Das Entwicklungsszenario, welches hinter diesen Fragen liegt, ist denkbar einfach: Es gibt zwei Großgruppen, die je drei Innovationen erstellen. Eine Großgruppe dokumentiert dabei künftige Szenarien der Innovation gemäß [BF03], eine Zeitplanung, sowie Arbeitsverteilung und weitere Kommentare in einem Projektprotokoll, während die andere Gruppe nur die künftigen Szenarien beschreibt und diese dann in dem oben beschriebenen Entscheidungsmodell abbildet.

Das Vorgehen bei der Entwicklung der Innovation ist ansonsten völlig gleich: Es wird von den Betreuern eine grobe Featureliste sowie Pressemitteilungen den Studierenden zur Verfügung gestellt, auf deren Basis sie eine Liste an Anforderungen für das von Ihnen zu entwickelnde System erstellen sollen. Auf Basis dieser Anforderungen erstellen sie Testfälle für das System und modellieren es in Statemate. Die Weiterentwicklung des Systems findet in zwei weiteren Iterationen statt.

Um das Szenario möglichst an den Begebenheiten der Praxis auszurichten, werden jeweils drei gleiche Innovationen innerhalb einer Großgruppe erstellt, die wiederum von den Kleingruppen je nur in einer Iteration bearbeitet werden: so kommt es, dass jede Kleingruppe jede der drei Innovationen in genau einer Iteration bearbeitet. Die Notwendigkeit der drei Iterationen ist daher gegeben, um eine Weiterentwicklung der Innovationen zu forcieren, auf Basis derer sich Varianten ausgestalten, die mittelfristig in den Rahmen einer Produktlinie gebettet werden können.

3 Durchführung des Experiments

In diesem Abschnitt beschreiben wir die Durchführung des Experiments im einzelnen. Zunächst gehen wir auf die wesentlichen Einschränkungen ein, die im Rahmen der Untersuchung an

der Universität zu beachten waren. Im Anschluss daran geben wir einen Grobüberblick über den Experimentablauf und gehen dann auf den Ablauf im Detail ein.

Die experimentelle Untersuchung dieses Berichts wurde, wie oben beschrieben, im Rahmen eines Praktikums im Hauptstudium der Informatik-Fakultät an der Universität Ulm im Sommersemester 2005 durchgeführt. Dies impliziert eine Reihe von Faktoren, die im Laufe des Experiments zu beachten waren:

- *Dauer*: Das Experiment war auf die Dauer eines Semesters beschränkt. Damit war ein Zeitrahmen geschaffen, der sinnvoll verwendet und nicht überschritten werden sollte. Dies ist eine wichtige Einschränkung, betrachtet man die externe Validität: Typischerweise nehmen Entwicklungen in der Praxis einen sehr viel längeren Zeitraum ein als drei Monate. Andererseits sind Experimente mit längerer Dauer kaum noch finanzierbar. Betrachtet man weiterhin Experimente im universitären Umfeld, so stellt man fest, dass sie meistens sehr viel kürzer sind: Weit verbreitet sind Kurzexperimente, die nur innerhalb einer Vorlesungsstunde durchgeführt werden. Wir sind der Meinung, dass die Dauer eines Semesters einen guten Kompromiss zwischen finanzierbarer Machbarkeit und dem Nutzen darstellt.
- *Umfang*: Durch die Beschränkung der Dauer folgt auch eine Beschränkung des Umfangs dessen, was in diesem Praktikum vermittelt und was damit in diesem Experiment untersucht werden kann. Der Umfang muss so gewählt werden, dass die Studierenden ihn in dem vorgegebenen Rahmen bewältigen können (andernfalls wirkt sich dies negativ auf die Motivation aus), aber auch, dass man neue und interessante Ergebnisse aus dem Experiment generieren kann. Dies haben wir vor allem dadurch versucht zu gewährleisten, dass wir das Experiment nur an einem Ziel ausgerichtet haben (welche Dokumentationsmethode ist erfolgreicher), so dass wir alle Bemühungen auf die Beantwortung dieser Frage ausrichten konnten. Der edukative Aspekt des Experiments spielt hier natürlich auch eine Rolle, wird aber im Rahmen dieser Veröffentlichung nur am Rande betrachtet.
- *Teilnehmerzahl*: Ein Experiment dieser Art ist abhängig von der Teilnahme der Studierenden. Dies ist ein Faktor, der im Vorherin schwierig vorherzusehen und damit zu planen ist. Die Teilnehmerzahl wirkt sich direkt auf die statistische Signifikanz eines Experiments aus. Wir gehen grundsätzlich von statistisch nicht signifikanten Ergebnissen im Rahmen dieser universitären Experimente aus, da sie, unabhängig von der Anzahl der Studierenden, im Wesentlichen einen wichtigen Mangel haben: Die Probanden sind nicht repräsentativ. Sie sind weder repräsentativ für Entwickler in der Praxis, da sie nicht deren Profil (Erfahrung, Wissen, Probleme, Umfeld, ...) haben, noch sind sie repräsentativ für Studierende der Informatik im Hauptstudium, da sie nicht zufällig ausgewählt wurden: Es werden Studierende aller anderen Universitäten außerhalb von Ulm benachteiligt und von den Studierenden der Informatik in der Universität Ulm erreicht man nur einen kleinen Teil.

Betrachtet man diese Einschränkungen, so ist man bei der Betrachtung der internen Validität des Experiments. Mit der internen Validität wird beschrieben, inwieweit Störvariablen bekannt und kontrolliert sind [Pre01]. Störvariablen sind all die Variablen, die das Ergebnis des Experiments beeinflussen können, die aber nicht als unabhängige Variablen in den Fragen des Experiments benannt sind.

Störvariablen und darauf reagierende Gestaltung des Experimententwurfs sind:

- *Kommunikation*: Die Kommunikation der Studierenden über die Großgruppengrenzen hinweg kann das Bild des Verständnis verzerren. Wenn sich die Studierenden über ihre unterschiedlichen Dokumentationsmethoden austauschen und die andere Dokumentationsmethode gegebenenfalls mit anwenden, so können die Fragen 2 und 4 aus Abschnitt 2.4 nicht mehr beantwortet werden.

Gestaltung Experiment: Diese Störvariable wird im Wesentlichen durch drei Aspekte kontrolliert. Erstens wird den Studierenden die Dokumentationsmethode der jeweils anderen Gruppe nicht beigebracht, damit gibt es einen gewissen Aufwand, sich in die neue Methode selbständig einzuarbeiten. Weiterhin werden die Interviews und dokumentationsbezogenen Diskussionen nur innerhalb der jeweiligen Großgruppen durchgeführt. Zweitens haben wir in mehreren Jahren der Durchführung ähnlicher Experimente festgestellt, dass die Studierenden sich kaum über die Grenzen ihrer Kleingruppen hinweg treffen oder zu fachlichen Themen austauschen. Drittens wurden die Projektprotokolle zu Beginn einer neuen Iteration nicht an die andere Gruppe weiter gegeben, währenddessen die Entscheidungsmodelle weiter gegeben und dabei regelmäßig erweitert wurden. Damit war ab der zweiten Iteration nicht nur die Dokumentationsmethode für die jeweils andere Gruppe unklar, sondern auch die Inhalte waren nicht mehr zu vergleichen, weil sich auch die Innovationen auf unterschiedliche Art und Weise entwickelt haben. Nicht zuletzt hatten die Studierenden auch keinen Anreiz, einen solchen Mehraufwand auf sich zu nehmen: Da das Experiment nicht benotet wurde und weder die eine noch die andere Methode eindeutig besser war, war das Hauptaugenmerk der Studierenden auf die Arbeit an den eigenen Artefakten, vor allem den Statemate-Modellen, gelenkt.

- *Aufteilung der Betreuung*: Das Praktikum wurde von drei Betreuern durchgeführt. Bei der Durchführung der Interviews wurden die beiden Großgruppen getrennt von jeweils unterschiedlichen Betreuern befragt. Wenn sich die Studierenden durch den ihnen zugeordneten Betreuer bevorzugt/benachteiligt gesehen hätten, so könnte das die Motivation beeinflussen und somit zu verzerrten Ergebnissen führen.

Gestaltung Experiment: Um eine einseitige Ausrichtung der Studierenden auf einzelne Betreuer zu vermeiden, haben wir das Experiment so gestaltet, dass zwar in den Interviewrunden zumeist für eine Großgruppe auch der/die bisherige(n) Betreuer zur Verfügung standen, allerdings standen die Betreuer ansonsten immer für alle Studierenden zur Verfügung; eine umfassende Verfügbarkeit der Betreuer ist im Rahmen eines solchen Experiments vonnöten, da die Betreuung der Studierenden neben den Treffen einen hohen Stellenwert einnimmt, da sie mit einer ganzen Reihe ihnen neuer Methoden und Werkzeugen arbeiten müssen, was natürlich einen gewissen Lernaufwand darstellt.

- *Motivation durch unterschiedliche Dokumentationsmethodik*: Die Studierenden werden durch die von ihnen zu bearbeitenden Aufgaben entweder motiviert oder demotiviert. Da wir in diesem Experiment nur den einen Unterschied zwischen den beiden Großgruppen haben, nach welcher Dokumentationsmethodik sie vorzugehen haben, wenn sie Weiterentwicklungen des von Ihnen entwickelten Systems ausmachen, so müssen wir diesen Unterschied genau betrachten, ob er eine Differenz in der Motivation zwischen den beiden Großgruppen begründet.

Gestaltung Experiment: Wir haben das Experiment so gestaltet, dass wir auch diese Störvariable durch mehrere Aspekte kontrollieren können. Zunächst muss man die Innovativität der beiden Ansätze vergleichen und klären, ob aus einer für innovativ befundenen Herangehensweise ein Motivationszuwachs oder -verlust entsteht. Wie oben schon beschrieben, war für die Studierenden weder die eine noch die andere Herangehensweise innovativer: Da für die Studierenden alles neu war, konnten sie auch an dieser Stelle keinen Unterschied ausmachen. Die unserer Meinung innovativere Herangehensweise — die Dokumentation mit Hilfe der Entscheidungsmodelle — wurde von den Studierenden auch nicht als spannender betrachtet, wie wir bei der anfänglichen Verteilung der Studierenden auf die Großgruppen feststellten. Es herrschte vielmehr die Ungewissheit vor, dass mit den Entscheidungsmodellen, die neben den Szenarien zu beschrieben waren, nicht ein Mehraufwand hinzukommt. Auch dieser Punkt ist zu betrachten: Wir wollten, dass die Studierenden in den beiden Großgruppen einen vergleichbaren Aufwand haben. Daher haben wir für die Großgruppe, die die Projektprotokolle schrieben, eine umfangreichere Vorlage mit mehr Aufgaben gefordert, als für die Gruppe, die nur die Szenarien und die Entscheidungsmodelle beschreiben sollten. Damit erreichten wir einen Ausgleich zwischen den beiden Ansätzen bezüglich Innovativität und Aufwand.

- *Aufteilung der Studierenden auf die unterschiedlichen Großgruppen:* Diese Störvariable könnte sich in der Art ausgestalten, dass die Methodik, die von den Studierenden als die herausforderndere betrachtet wird, von schwächeren Studierenden gemieden und von guten Studierenden gesucht wird.

Gestaltung Experiment: Die Studierenden wurden jeweils in Teams bestehend aus zwei Personen auf die beiden Großgruppen verteilt. Dabei haben wir zunächst versucht, die Wünsche der Studierenden zu beachten. Allerdings hatten die meisten Studierenden keine klaren Präferenzen, weil sie zum Zeitpunkt der Einteilung auf die Großgruppen keine Vorstellung hatten, worin sich die beiden Ansätze unterschieden, weil wir dazu keine Angaben gemacht haben. Wir haben sie insoweit informiert, dass sich beide Ansätze mit der Weiterentwicklung der Innovation im Rahmen von Szenarien beschäftigen. Dies war insofern wichtig, damit sich bei der Einteilung nicht die oben beschriebene Aufteilung der Studierenden ergibt. Da wir in diesem Experiment die vergleichsweise große Anzahl von 22 Studierenden zur Verfügung hatten, kann man von einer zufälligen Verteilung der Gruppe ausgehen.

3.1 Grobaufbau des Experiments

Im folgenden Abschnitt geben wir einen Überblick über den Grobaufbau des Experiments. Aus den vorherigen Abschnitten ist die Aufteilung der Studierenden in zwei Großgruppen bekannt, sowie die Bearbeitung von drei unterschiedlichen Innovationen in jeweils einer von drei Iterationen in einer Kleingruppe zu zwei Studierenden. Der vorliegende Abschnitt stellt nun den Gesamtexperimentablauf schematisch dar.

Im Vorfeld des Experiments gab es eine Trainingsphase, in der die Studierenden mit den Werkzeugen und Dokumenten, die im Verlauf des Experiments auf sie zukommen, vertraut gemacht werden sollten. Die grobe Übersicht über den zeitlichen Rahmen des Experiments ist in Abbildung 7 dargestellt. Auf die mit „Training“ bezeichneten Einführungswochen wird im folgenden Abschnitt 3.2 genauer eingegangen.

Außerdem wird in der Abbildung der grobe zeitliche, schematische Ablauf innerhalb einer

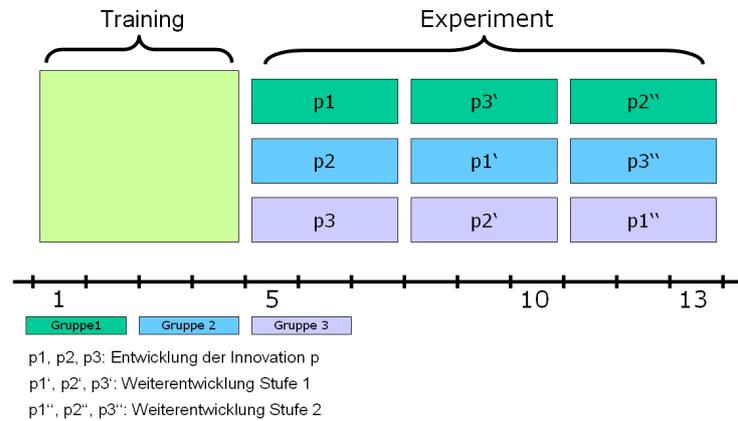


Abbildung 7: Übersicht über das Experiment.

Großgruppe dargestellt. Es werden drei Kleingruppen ausgezeichnet, die in drei Iterationen jeweils eine Innovation (weiter)entwickeln. Ein Projekt bezeichnet hierbei die Entwicklung eines Systems in mehreren Iterationen. Die Projekte sind von p1 bis p3 durchnummeriert und starten in der ersten Iteration, die drei Wochen andauert. In einer zweiten Iteration werden die Systeme getauscht und von einer anderen Gruppe weiterentwickelt (was durch p_i' dargestellt ist). Ebenso geht es in der dritten Iteration weiter. Der Ablauf innerhalb einer Iteration wird im Abschnitt 3.3 detaillierter beschrieben.

3.2 Einführungswochen

In Abbildung 8 wird der Ablauf der Einführungswochen der Trainingsphase gezeigt. In den einzelnen Sitzungen wurden den Studierenden das notwendige Wissen vermittelt, das für das Praktikum notwendig war.

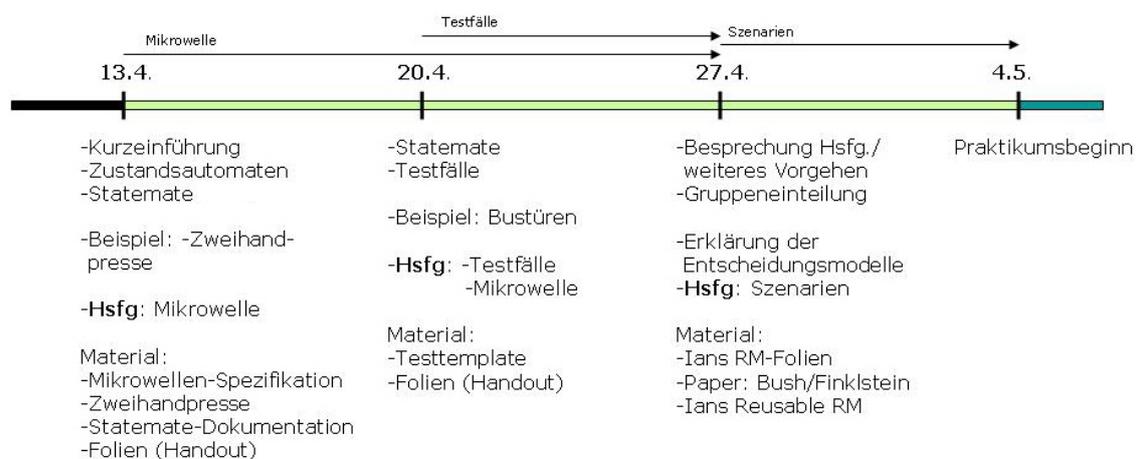


Abbildung 8: Einführungswochen der Studierenden

Beim ersten Treffen wurde eine kurze Einführung in den Verlauf des Experiments gegeben. Um die Zeit, die uns zur Verfügung stand, möglichst gut zu nutzen, wurde außerdem mit der Einführung in das Werkzeug Statemate sowie Zustandsautomaten begonnen.

Statemate sowie zustandsorientierte Modellierung für die Entwicklung von eingebetteten Systemen muss im Vorfeld ausgiebig behandelt werden, da sowohl das Werkzeug eine große Mächtigkeit aufweist, als auch die Studierenden sich mit der Modellierung praktisch nicht auskennen; während die Programmierung typischerweise im Rahmen des Informatik-Studiums ausgiebig behandelt wird, wird die Modellierung zumeist nur theoretisch behandelt. Da Statemate den Zustandsautomaten eine eindeutige Semantik zuordnet, lässt es sich ausführen und ist damit gut geeignet, komplexe Spezifikationen zu modellieren und zu testen. Die Modellierung mit Statemate macht somit auch einen gewissen Charme dieses Praktikums aus, da man sowohl praktische Entwicklungserfahrungen sammeln kann, aber sich nicht mit teilweise trockenen Programmiersprachen beschäftigt. Um die Studierenden mit dem notwendigen Wissen für die Experimentphase auszustatten, mussten sie in der vergleichsweise kurzen Einführungsphase eine Reihe von Hausaufgaben erledigen, um sich mit dem Werkzeug anhand von praktischen Beispielen auseinanderzusetzen. Als erstes Beispiel wurde in der ersten Sitzung eine Zweihandpresse erarbeitet, die nur wenig Funktionalität besitzt, aber an der sehr schön die wesentlichen Bestandteile von Statemate dargestellt werden können. Als Hausaufgabe bekamen die Studierende die Modellierung einer Mikrowelle auf. In Abbildung 8 sind Hausaufgaben durch einen Pfeil gekennzeichnet, der sich am oberen Rand über der Zeitleiste befindet. Die Länge des Pfeils gibt den Zeitraum an, für den die Hausaufgabe geplant war. Unter den jeweiligen Daten der Sitzungen stehen die Punkte, die während der Sitzungen abgehandelt wurden. Dort befinden sich auch die Hausaufgaben, die den Studierenden gestellt wurden und die Materialien, die sie ausgehändigt bekamen oder von der Webseite des Praktikums laden konnten.

Im zweiten Treffen wurde unter anderem die Erstellung von Testfällen besprochen. Die Testfälle sollten dabei in einem Dokument erstellt werden, das jede Gruppe einzeln für die Überprüfung der eigenen Modellierung entwickeln sollte. Die Testfallerstellung und die richtige Aufstellung im Dokument wurde anhand eines Fallbeispiels aus der Automobilindustrie besprochen. Als Gegenstand diente der Schließmechanismus einer Bustür. Das Material, das den Studierenden zu dieser Sitzung gegeben wurde, war eine Testvorlage, in die sie ihre Testfälle eintragen sollten, sowie ein Foliensatz, der das Vorgehen noch einmal erläutert und an einem Beispiel illustriert.

Beim dritten Treffen wurden die Hausaufgaben und die Schwierigkeiten bei der Erstellung besprochen. Danach erfolgte die Einteilung der Studierenden auf die beiden Großgruppen sowie in die Kleingruppen. Aus den elf Kleingruppen, die sich aus den 22 Studierenden bildeten, ordneten sich sechs Kleingruppen der Projektprotokoll-Großgruppe (PG) und fünf Kleingruppen der Entscheidungsmodell-Großgruppe (EG) zu. Innerhalb einer Großgruppe teilten wir die Kleingruppen in zwei Untergruppen auf. Somit hatten wir folgende Gruppeneinteilung:

Im Anschluss an die Gruppeneinteilung wurden die Studierenden der Entscheidungsmodell-Gruppe speziell in der Erstellung der Entscheidungsmodelle geschult.

3.3 Iterationsübersicht

Der Ablauf innerhalb einer Iteration ist in zwei Teile untergliedert (Abb. 9). Der erste Teil besteht aus den Phasen Analyse und Entwurf. In diesem ersten Teil sollten die Anforderungen der jeweiligen Iteration erhoben und dokumentiert und gegebenenfalls die übergebenen Doku-

	Projektprotokoll-Gruppe (1-x-x)	Entscheidungsmodell-Gruppe (2-x-x)
Teilgruppe 1 (x-1-x)	1-1-1	2-1-1
	1-1-2	2-1-2
	1-1-3	2-1-3
Teilgruppe 2 (x-2-x)	1-2-1	2-2-1
	1-2-2	2-2-2
	1-2-3	

Tabelle 1: Gruppeneinteilung der Studierenden.

mente der vorherigen Gruppe untersucht werden. Die Anforderungen sollten auf Basis der von den Betreuern ausgehändigten groben Featureliste inklusive einer detaillierten Beschreibung des technologischen Umfelds für die Realisierung in Statemate (sogenanntes Panel) sowie Pressemittelungen (siehe Anhang) erstellt werden; wir haben diesen wenig kontrollierbaren und beeinflussbaren Mechanismus gewählt, um die Kreativität der Studierenden anzuregen und die Entwicklung sich unterscheidender Systeme zu unterstützen, und somit auch unterschiedliche Vorschläge der Weiterentwicklung der Systeme zu generieren. Die Unterlagen, die wir im Laufe des Praktikums den Studierenden ausgehündigt haben, wurden alle vor Beginn des Praktikums vollständig erstellt, um somit Beeinflussungen durch von Studierenden erstellte Inhalte zu vermeiden.

Da es sich bei den einzelnen Entwicklungsprojekten jeweils um kleinere Aufgaben handelte und diese somit recht einfach umzusetzen waren, wurde für die Modellierungsphase ein Zeitraum von zwei Wochen veranschlagt. Die Studierenden sollten ihre Anforderungen sowie ihren Entwurf, den sie in der ersten Woche einer Iteration fertiggestellt haben, in ein Statemate-Modell einbringen. Dabei schufen die Studierenden in der ersten Iteration die Grundlage für die weiteren Iterationen, indem sie Modelle entwarfen, die den Kern der darauf folgenden Iterationen umfassten; weiterhin waren sie angehalten darauf zu achten, dass die von ihnen selbst identifizierten, denkbaren Weiterentwicklungen der Systeme durch ihr Modell und ihre Anforderungen insoweit unterstützt werden sollten, dass die Weiterentwicklung nicht das bisherige Modell in hohem Maße ändern müsste, um realisiert zu werden. Die Studierenden mussten in den weiteren Iterationen auf den Modellen und Dokumenten anderer Gruppen aufbauen und die Änderungen und Neuerungen in das schon bestehende Modell einbauen. Somit war es ein Gewinn auch für sie selber, wenn sie die Modelle und Dokumente änderungs- und erweiterungsfähig gestalteten. Dies sollten sie vor allem durch eine gute Übersichtlichkeit, Kapselung und Vereinzelung der Modellelemente erreichen.

Stellvertretend für alle drei Iterationen sollen die drei Wochen der zweiten Iteration (siehe Abbildung 10) näher erläutert werden. In der ersten Iterationswoche waren die Studierenden damit beschäftigt, sich in ein ihnen neues System einzuarbeiten, das sie von der Gruppe erhalten haben, die an diesem System in der vorherigen Iteration gearbeitet haben. Beide Großgruppen haben alle Entwicklungsunterlagen der vorherigen Gruppe erhalten (Anforderungen, Statemate-Modelle, Testfälle), allerdings mit einer Ausnahme: Die Projektprotokoll-Gruppe hat das jeweilige Projektprotokoll der Iteration nicht weiter gegeben, während die Entscheidungsmodellgruppe ihre Entscheidungsmodelle (ohne die zugehörigen Szenarien) weiter gegeben hat. Wie oben schon beschrieben, haben wir diesen Unterschied im Ablauf deswegen für nötig befunden, um den Forderungen der externen Validität gerecht zu werden. Momentan

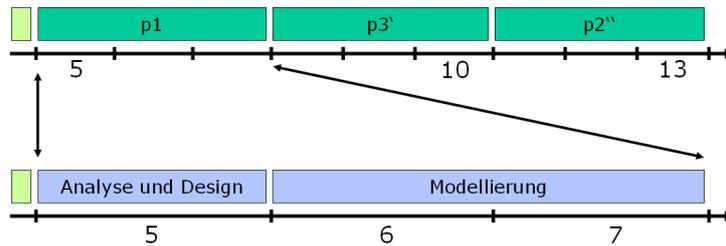


Abbildung 9: Aufteilung einer Iteration.

ist es in der Praxis der Fall, dass Entwicklungsartefakte wie Projektprotokolle nach Projektende nicht mehr auffindbar oder zumindest nicht mehr praktisch nutzbar sind. Eine weitere Verbesserung, die mit Hilfe der dokumentierten Entscheidungsmodelle umgesetzt wird, ist die auch über das Projektende andauernde Verfügbarkeit von verworfenen und weiterhin denkbaren System-Optionen. Um allerdings die Ergebnisse nicht insoweit zu verzerren, dass mehr dokumentiertes Material von einer Gruppe zu der anderen Gruppe gegeben wird, haben wir für beide Großgruppen darauf verzichtet, die Szenarien weiter zu geben. In der zweiten Iteration mussten die Studierenden die übergebenen Materialien und damit die Systeme zunächst ausführlich testen und eine Qualitätsüberprüfung durchführen. Gegebenenfalls musste das System (bei kleineren Mängeln) überarbeitet oder (bei größeren Beanstandungen) an die ursprüngliche Gruppe zur Überarbeitung zurückgegeben werden. Ließ sich dann das System ausführen und unterstützte es die bis dahin geforderten Anforderungen, so musste die Gruppe nun anhand von Presseberichten und groben Anforderungen, die von den Betreuern erstellt wurden, das System in der zweiten Iteration weiterentwickeln. An diesem Ablauf erkennt man, dass die von uns Iterationen genannten Phasen als in sich abgeschlossene Entwicklungsprojekte von den Anforderungen bis hin zu dem simulierbaren System in Statemate in der Praxis anzusehen sind. Natürlich dauert die Entwicklung eines innovativen Systems sehr viel länger, doch mussten wir für die Bedürfnisse dieses Experiments einen recht langen Zeitraum in den Rahmen eines Semesters packen, was zu diesen kurzen Projektzyklen führte. Um sinnvolle Varianten und Ausgestaltungen zu erreichen, mussten mindestens drei Projektdurchläufe beobachtet werden.

In Abbildung 10 ist dargestellt, welche Dokumente die Kleingruppen am Anfang einer Iteration jeweils von wem bekommen haben. Die Kleingruppen der Projektprotokoll-Großgruppe erhielten jeweils von einer anderen Kleingruppe aus der Projektprotokoll-Großgruppe das Anforderungsdokument, das Statemate-Modell und die Testprotokolle. Die Projektprotokolle wurden in Iteration 1 zwar erstellt, werden aber — wie oben erwähnt — nicht an die nächste Gruppe weitergegeben. Die Kleingruppen der Entscheidungsmodell-Großgruppe bekamen dieselben Dokumente wie die Kleingruppen der Projektprotokoll-Großgruppe, erhielten aber zusätzlich die Entscheidungsmodelle. Von den Betreuern wurde dann jeder Gruppe jeweils passend zu der von ihnen zu erstellenden Innovation eine Liste an groben Anforderungen, Pressemitteilungen und Hardware-Anforderungen übergeben, wobei die letzteren für eine abgestimmte Schnittstelle zur Umsetzung in Statemate notwendig waren.

Nach Abschluss der Analyse- und Entwurfsphase am Ende der ersten Woche mussten die Studierenden die von ihnen entwickelten und neu erstellten Dokumente den Betreuern zur Einsicht geben. Diese umfassten für die Projektprotokoll-Gruppe das Anforderungsdokument,

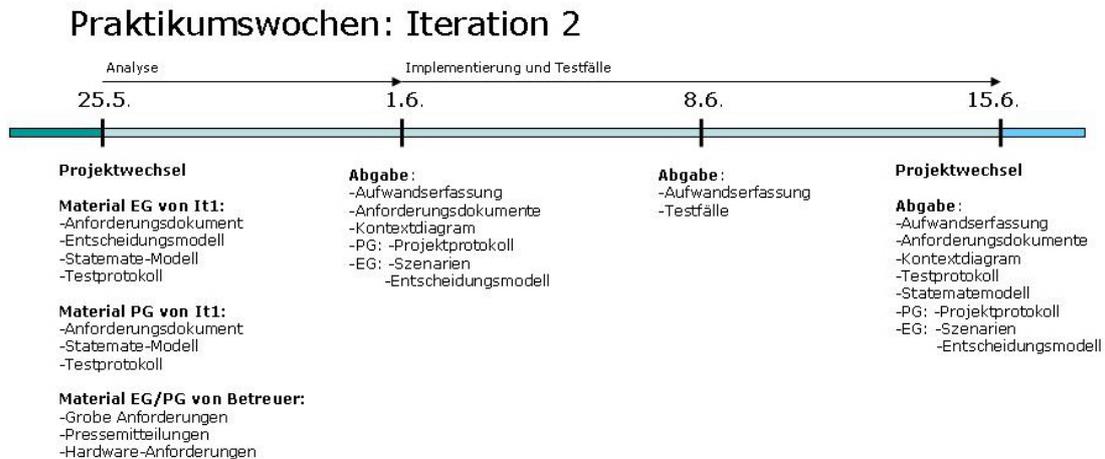


Abbildung 10: Aufgaben innerhalb einer Iteration.

die Kontextdiagramme und das Projektprotokoll. Die Entscheidungsmodell-Großgruppe musste anstatt des Projektprotokolls die Szenarien und das Entscheidungsmodell abgeben. Wie in jeder Woche seit Beginn des Experiments musste eine Aufwandserfassung erstellt und über eine Webanwendung abgegeben werden. Damit war dieser Teil der Iteration abgeschlossen und die Studierende konnten mit der Modellierung des Systems sowie der Erstellung der Testfälle beginnen, wofür sie zwei Wochen Zeit hatten. Am Ende dieser Phase mussten den Betreuern die Testfälle und das Statemate-Modell zur Einsicht vorliegen.

Dieser Experimentaufbau sollte die Anwendbarkeit und Nützlichkeit der beiden Dokumentationsmöglichkeiten (Entscheidungsmodell versus Projektprotokoll) untersuchen. Um eine Antwort auf die Frage zu finden, inwieweit sich die gewählte Dokumentationsform auf die spätere Einbettung der Innovation mit festgestellten Varianten in eine Produktlinien-Systematik auswirkt, sollten die Studierenden ein Merkmaldiagramm [KCH⁺90, HMT01, ES04] nach Abschluss der Iteration 3 entwickeln. Dieses Modell sollte aufzeigen, an welchen Stellen Variabilität auftritt und welche Merkmale sich im Kern aller Produkte der Produktlinie, die auf dieser Innovation beruht, befinden. Die Auswertung der Merkmaldiagramme sollte mithin die Frage beantworten, inwieweit sich die gewählten Entscheidungsmodelle als Werkzeug für das proaktive Domain Engineering eignen. In dem Teil dieses Berichts, der die Ergebnisse beschreibt, werden wir darauf eingehen, warum es nicht möglich war, diese anfangs als einleuchtend erscheinende Methode der Auswertung vorzunehmen und wie wir anstelle dessen vorgegangen sind.

4 Verwendete Dokumente

In diesem Abschnitt werden wir genauer auf die im Verlauf des Experiments verwendeten Dokumente eingehen, mit Ausnahme der Entscheidungsmodelle, weil wir diese schon in Abschnitt 2.2 ausführlich erläutert haben. Nicht alle Dokumente mussten von allen Gruppen erstellt werden. So wird innerhalb der folgenden Absätze unterschieden, ob es sich um ein Dokument handelt, das nur die Entscheidungsmodell-Großgruppe, nur die Projektprotokoll-

Gruppe oder beide Gruppen erstellen mussten oder erhalten haben.

Die Gliederung der folgenden Unterabschnitte ist nach der Abgabe ausgerichtet. An erster Stelle werden die Dokumente behandelt, die beide Gruppen erhielten oder abgeben mussten (Abschnitt 4.1 bis 4.7). Danach folgen die Dokumente, die ausschließlich der Projektprotokoll-Gruppe zuzuordnen sind (Abschnitt 4.8). Am Ende folgen die Dokumente, die nur die Entscheidungsmodell-Großgruppe betrafen (Abschnitt 4.9).

4.1 Aufwandserfassung

Die Aufwandserfassung trägt einen wichtigen Aspekt zur Auswertung der Ergebnisse des Experiments bei. Für dieses Experiment haben wir erstmals eine Aufwandserfassung über das Internet genutzt, die es den Studierenden ermöglichte, ihre Aufwände zeitnah und komfortabel anzugeben und die uns bei der Auswertung der Aufwände unterstützte, da diese schon kategorisiert und auf die jeweilige Gruppe bezogen gleich in einer internen Datenbank vorlagen. Für diese internetbasierte Aufwandserfassung haben wir eine Erweiterung eines bestehenden Programms entwickelt, das in der Abteilung Compilerbau und Programmiermethodik an der Universität Ulm als Lehr- und Lernplattform genutzt wird. Damit verbunden war auch eine komfortable Möglichkeit, die Unterlagen und Materialien für die Studierenden über das Internet abzuholen.

The screenshot shows the 'rubikon Lernplattform' interface. At the top, it says 'Aufwandserfassung: Aufwandserfassung 1. Iteration, 3. Woche' and 'Eingeloggt als matze Logout'. On the left, there are navigation options: 'Veranstaltung', 'Lernziele', and 'Persönlicher Bereich'. The main content area is titled 'Aufwandserfassung' and 'Übersicht über alle bisher eingegebenen Aufwände (in Minuten):'. Below this is a table with the following data:

Zeitpunkt	V	K	F	ÜPP	ÜAD	ÜKD	ÜS	ÜRM	ÜTF	DTF	ESC	ÜSC	S	Aktionen
16.08.2005 12:25	128	30	25	40	35	0	0	0	0	0	0	0	0	
16.08.2005 12:25	0	34	23	34	0	2	0	0	0	0	0	0	0	
16.08.2005 12:26	0	0	0	0	0	0	0	0	23	23	6	55	44	

Below the table, it says 'Geben Sie bitte Ihren neuen Aufwand pro Kategorie ein:'. There are two input sections:

- V: Verständnis**
Aufwand für Verständnis (z.B. Lesen der Anforderungen), Überlegungen. Fields for 'Stunde(n)' and 'Minuten' with input boxes, and a 'Beschreibung/Kommentar' text area.
- K: Kommunikation**
Aufwand für gruppeninterne Kommunikation und (Termin-)Absprachen. Fields for 'Stunde(n)' and 'Minuten' with input boxes, and a 'Beschreibung/Kommentar' text area.

Abbildung 11: Aufwandserfassung der Studierenden.

Abbildung 11 zeigt einen Ausschnitt der Eingabemaske, in der die Studierenden ihre Aufwände angeben konnten. Hier kann man erkennen, dass die Aufwände in Kategorien aufgeteilt wurden. Die Studierenden waren angehalten, ehrlich zu antworten, damit wir ein möglichst genaues Bild der Aufwandsverteilung bekommen konnten. Die Studierenden waren informiert, dass die Erfassung der Aufwände notwendig für die Gestaltung des Experiments und Auswertung des Experiments ist, aber keine Rolle bei der Bewertung der Arbeiten der Studierenden spielt. Weiterhin appellierten wir an den Forschergeist der Studierenden, um zu belastbaren Auswertungen zu kommen; auch dies ist Teil des „Action Research“ Ansatzes, dass die Probanden einen Teil der Verantwortung zur erfolgreichen Durchführung des

Experiments übernehmen.

4.2 Anforderungsdokument

Die Studierenden sollten die von Ihnen auf Basis der groben Beschreibung und Pressemitteilungen erstellten System-Anforderungen in Form einer Excel-Tabelle dokumentieren. Die Betreuer haben den Studierenden eine Vorlage in Excel für die Anforderungsliste zur Verfügung gestellt, die aus vier Spalten besteht:

- Beschreibung des System-Grundmoduls

In dieser Spalte sollen die Anforderungen, Erläuterungen und Entwurfsentscheidungen der Innovation beschrieben werden. Dabei ist das gesamte Dokument gemäß den benötigten Komponenten des Systems sowie den Kontextdiagrammen zu strukturieren. Überschriften sollen fett geschrieben werden. Die Studierenden hatten bezüglich der Anforderungen an ihr System nur die Einschränkung der technischen Machbarkeit zu beachten (es musste mit den vorgegebenen Sensoren und Aktoren sowie dem bestehenden Kommunikationsnetzwerk ausgekommen werden — natürlich hat sich die technische Machbarkeit von Iteration zu Iteration erweitert, um damit die Weiterentwicklung der Technik zu simulieren und die Studierenden regelmäßig zu neuen Ideen anzuregen). Bis auf diese Einschränkung konnten die Studierenden ihre eigenen Ideen umsetzen und wurden diesbezüglich auch von uns unterstützt.

- Form

In dieser Spalte soll näher spezifiziert werden, was in der Hauptspalte (Beschreibung des System-Grundmoduls) in der jeweiligen Zeile dokumentiert wurde: Es werden Anforderungen, Überschriften, Erläuterungen und Entwurfsentscheidungen unterschieden. Für den Fall, dass es sich um eine Entwurfsentscheidung handelt, muss in der Begründung auf das entsprechende Szenario, welches der Entwurfsentscheidung zugrunde liegt, verwiesen werden.

- Begründung

In dieser Spalte gibt es die Möglichkeit, Begründungen anzugeben. Begründungen sollen für die Anforderungen dann angegeben werden, wenn sie sich nicht durch eine grobe Anforderung der Betreuer erschloss, also wenn eine Anforderung aus einer Pressemitteilung abgeleitet wird oder wenn sie sich beispielsweise im Gespräch ergibt. Des weiteren sind Entwurfsentscheidungen (also die Entscheidung für eine spezielle Entwurfsoption) explizit durch das der Entscheidung zugrunde liegende Szenario zu begründen.

- Akzeptanzkriterium

Für den Fall, dass die Anforderung als abstrakter Testfall nicht ausreicht, gibt es hier die Möglichkeit, ein Akzeptanzkriterium anzugeben.

4.3 Grobe System-Anforderungen

Die groben System-Anforderungen, die die Studierenden von den Betreuern als Einstieg in jede Iteration erhalten haben, stellen eine grobe Beschreibung der Funktionalität des Systems dar, wie sie sich einem Nutzer darstellt und die somit vom Management gefordert wird (siehe Anhang 7.1). Sowohl mit dieser Grobbeschreibung als auch einer detaillierten Beschreibung der Hardware und der vorhandenen Sensoren und Aktoren sollten die Studierenden ihre eigenen, ausführlichen Anforderungen für die Innovation erarbeiten. Die Freiheit, die wir den Studierenden dabei belassen haben, sollte sie dazu anregen, sich mit Varianten und Weiterentwicklungen des Systems zu beschäftigen und somit zu stichhaltigen Szenarien zu gelangen.

4.4 Testprotokoll

Neben der Anforderungsdokumentation sollten die Studierenden Testfälle dokumentieren und die Ergebnisse der im Statemate-Modell durchgeführten Testfälle protokollieren. Dabei sollte das Testprotokoll einem bestimmten, von den Betreuern als Vorlage zur Verfügung gestellten, Format genügen:

Zur Identifikation der Gruppe und der Iteration muss zunächst der Gruppenname derjenigen Gruppe, die die Testfälle beschrieben hat sowie derjenige der Gruppe, die das Statemate-Modell mit Hilfe dieser Testfälle getestet hat und die Iteration angegeben werden. Der Durchlauf der Testfälle kann in den folgenden Abstufungen bewertet werden: „alles OK“, „teilweise OK“ (mit Kommentar) und „fehlgeschlagen“. In einem Prüfablauf wird eine kurzgefasste Testübersicht und Testspezifikation beschrieben. Die Testübersicht stellt dabei die „grobe“ Überschrift dar, unter der verschiedene einzelne Tests zusammengefasst werden. Ein Beispiel dafür wäre das Testen des Schließens von Türen. Dabei sind die funktionalen Abläufe für die vier einzelnen Autotüren immer dieselben und können daher unter der Testübersicht „Schließen der Tür“ zusammengefasst werden. In der Testspezifikation muss man nun aber auf die einzelnen Türen des Fahrzeugs eingehen und erhält somit verschiedene Testszenarien innerhalb dieser einen Überschrift.

4.5 Presseberichte

Für die einzelnen Iterationen haben die Betreuer Presseberichte bereitgestellt. Diese hatten den Zweck, die Studierende zu selbständigen und kreativen Denken bei der System-Entwicklung anzuregen. Sie sollten aus den Berichten mögliche System-Anforderungen für die gegenwärtige Entwicklung sowie mögliche Szenarien für die künftige (Weiter)Entwicklung der Innovation generieren. Die Berichte waren alle frei erfunden und in jeder Iteration gab es für jede Innovation auch einen Pressebericht, der unserer Meinung nach keinen Mehrwert für die Entwicklung darstellte. Dies war beabsichtigt, um eine intensivere Auseinandersetzung der Studierenden mit den Presseberichten zu forcieren.

Die Presseberichte finden sich im Anhang 7.1. Für die erste Iteration waren keine Berichte vorgesehen, da zunächst das Grund-System zu entwickeln war, das, da es ja völlig neu entwickelt wurde, noch nicht in Presseberichten behandelt werden konnte.

4.6 Kontextdiagramm

Kontextdiagramme sind aus der strukturierten Analyse bekannt [DeM78]. Wir haben schon in vorherigen Praktika positive Erfahrungen mit der Verwendung von Kontextdiagrammen

gemacht, da sie eine leicht verständliche Schnittstelle zwischen den vorgegebenen Hardware-Anforderungen sowie den von den Studierenden selbst zu spezifizierenden Funktionsanforderungen darstellen; macht man sich klar, wie das System sich in einem Kontextdiagramm darstellt, so ist die Übertragung in die zustandsorientierte Denkweise von State- und State-Transition-Diagrammen einfacher zu bewerkstelligen. Auch in diesem Praktikum haben wir daher Kontextdiagramme nach [HP87] eingeführt, um dadurch die Entwicklungsarbeit der Studierenden in der Analyse- und Entwurfsphase zu unterstützen. Dabei sollte für jede Funktion ein Diagramm erstellt werden. Auf der linken Seite des Kontextdiagramms (siehe Abbildung 12) stehen die Hardware-Einheiten, wie beispielsweise Sensoren. Pfeile zeigen hier Informations- oder Datenflüsse (hier zumeist Signale) an. Der Kreis in der Mitte des Diagramms stellt die Funktion dar, die in dem Kontextdiagramm näher beschrieben werden soll. Auf der rechten Seite werden die Aktoren, Funktionen oder Speicher dargestellt, die von der Funktion angesteuert beziehungsweise verändert werden. Der Kreis in der Mitte stellt also so etwas wie einen Transformator dar, der Signale von HW-Einheiten empfängt, diese entsprechend verarbeitet und dann wieder Signale an HW-Einheiten verschickt.

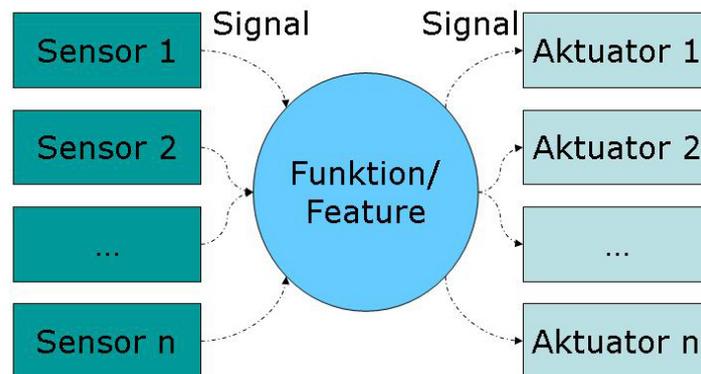


Abbildung 12: Template für das Kontextdiagramm.

4.7 Merkmaldiagramm

Das Merkmaldiagramm der aktuell entwickelten Innovation musste am Ende des Experiments von jeder Gruppe erstellt werden. Die Grundidee eines Merkmaldiagramms ist, Gemeinsamkeiten und Unterschiede der System-Varianten explizit zu machen. Im Anhang 7.3 ist ein Beispiel eines solchen Diagramms dargestellt.

Das Merkmaldiagramm ist ein gerichteter, azyklischer Graph. Ein Merkmaldiagramm besteht aus Konzeptknoten, Untermerkmal, Merkmalknoten, Eltern- und Kinderknoten oder -merkmale. Diese verschiedenen Bezeichnungen beschreiben die unterschiedlichen hierarchischen Instanzen dieses Konzepts. Das Merkmaldiagramm ist vom Aufbau her vergleichbar mit einem Baumdiagramm. Von einem Konzeptknoten oder Elternknoten/-merkmal spricht man, wenn man einen Knoten meint, der nicht als Blatt fungiert. Merkmalknoten, Untermerkmale und Kinderknoten stellen somit Knoten dar, die einen Elternknoten/-merkmal besitzen oder als „Blatt“ des Baums fungieren. Die Merkmalmmodellierung, die heute im Bereich der Produktlinienmodellierung verwendet wird, geht zurück auf die „*Feature-Oriented Domain*“

Analysis“ (*FODA*) [KCH⁺90].

Es gibt verschiedene Möglichkeiten, Knoten in diesem Baum zu platzieren und miteinander zu verbinden. So kann man einen Knoten in das Merkmaldiagramm einbringen, das nicht zu den *notwendigen Merkmalen* gehört. Damit sind beispielsweise *optionale Merkmale* gemeint, die nicht zu der Kernfunktionalität gehören, aber in einigen Produktvarianten dennoch umgesetzt wurden (Option). Diese optionalen Merkmale werden durch einen Kreis oberhalb des Merkmals dargestellt. Neben optionalen Knoten kann man weitere, nicht notwendige Merkmale auszeichnen. Dazu gehören *Alternativen*, die durch eine schwarze ausgefüllte Markierung gekennzeichnet sind, die sich über die Kanten der Merkmale erstreckt, die alternativ zueinander sind. Zu beachten ist dabei, dass eine Option gewählt werden kann aber nicht muss; im Gegensatz dazu muss genau eine der Alternativen gewählt werden. Die *Oder-Merkmale* hingegen gruppieren zwar ebenso wie die Alternativen eine Auswahl von Untermerkmalen eines Knotens, stellen aber die Möglichkeit zur Verfügung, mehr als nur eine Oder-Alternative zu wählen; mindestens eine Oder-Alternative muss jedoch gewählt werden. Weiterhin lassen sich in Merkmaldiagrammen *Abhängigkeiten* zwischen varianten Merkmalen darstellen, mit denen notwendige Einbeziehungen (gerichteter Pfeil) oder Ausschlüsse (Doppelpfeil) beschrieben werden können.

4.8 Projektprotokoll

Die Projektprotokolle wurden von der Projektprotokoll-Gruppe als alternative Dokumentationsform von optionalen Entwurfsentscheidungen verwendet. Die Projektprotokolle umfassten folgende Abschnitte:

- Ziele und Anforderungen des Systems

Hier soll kurz und prägnant festgehalten werden, was in diesem Iterationsschritt zu erarbeiten ist. Dabei sollen die Studierenden die geplanten, umzusetzenden Anforderungen beschreiben.

- Planung des Entwicklungsvorgehens, Aufteilung der Arbeiten sowie Zeitbedarf

Dieser Teil des Dokuments beschäftigt sich mit der Aufteilung der Arbeit zwischen den Studierenden der Kleingruppe. Hierbei soll der Verantwortliche für eine bestimmte Aufgabe festgehalten werden. Ebenfalls soll die Rolle des Studierenden (beispielsweise Modellierer) dokumentiert werden, die in diesem Entwicklungsschritt eingenommen wird. Dazu kommt der geschätzte Aufwand, der für die Bearbeitung dieses Schrittes notwendig ist und der Fertigstellungsgrad, der tatsächlich innerhalb der Iteration erreicht wurde. Dies sollte für die Auswertung als Anhaltspunkt dienen, wie gut die Studierenden ihre eigene und die anfallenden Arbeiten einschätzen können. Weiterhin konnten die Studierenden auf Basis dieser Daten eine Planung und sinnvolle Verteilung der anfallenden Arbeiten vornehmen.

- Beschreibung von mindestens vier Zukunftsszenarien (für die gesamte Iteration)

Um die Studierenden zukunftsorientiert und damit änderungs- und erweiterungsfreundlich arbeiten zu lassen, sollen sie sich mindestens vier Zukunftsszenarien der Innovation

überlegen, die für die Modellierung in Statemate eine Auswirkung haben würden. Dabei sollen sie sich an machbare Szenarien halten und nicht Szenarien aufschreiben, die unrealistisch sind. Bei den Szenarien ist es nicht wichtig, jedesmal grundlegend neue Ideen zu entwickeln, sondern auch kleinere Verbesserungsvorschläge an dem System sind willkommen. Das Szenario beschreibt zum einen die Entwicklung der Umwelt (Technik, Interesse, ...) sowie die interne Entwicklung der Innovation als Folge dieser geänderten Umwelt (in Anlehnung an [BF03]). Weiterhin soll in den Szenarien mit angegeben werden, welche Komponenten des Systems durch eine Änderung betroffen sein würden, also ob beispielsweise neue Sensoren zur Umsetzung des Szenarios benötigt würden.

- Klärung spezifischer Anforderungen, die aus den Pressemeldungen abgeleitet werden

Aus den Pressemitteilungen können Anforderungen abgeleitet werden, die die Studierenden finden und umsetzen sollen. Falls unklar ist, welche Anregung aus den Pressemitteilungen abzuleiten ist oder wenn innerhalb der Kleingruppe unterschiedliche Auffassungen darüber existieren, so können die Studierenden dies mit den Betreuern klären und die Ergebnisse dieser Klärung in den Projektprotokollen festhalten.

- Klärung technischer Fragestellungen

Falls die Studierenden technische Probleme haben, also beispielsweise mit Statemate Probleme haben, sollen sie diese dokumentieren und einen Lösungsvorschlag (gegebenfalls gemeinsam mit den Betreuern) finden.

Ein Beispieldokument aus der dritten Iteration des Experiments können Sie im Anhang unter 7.7 finden.

4.9 Szenarien

Die Projektprotokoll-Gruppen sollen die Szenarien direkt in deren Projektprotokolle einarbeiten. Diese sind somit ein Bestandteil des Hauptprotokolls. Bei der Entscheidungsmodell-Großgruppe sind die Szenarien ausgegliedert, da sie zur Erstellung und Ergänzung des Entscheidungsmodells beitragen sollen.

Der Aufbau des Dokuments, in dem die Szenarien für die Entscheidungsmodell-Großgruppe zu beschreiben sind, ist im Wesentlichen so wie die Szenarien im Projektprotokoll der Projektprotokoll-Gruppe. Dabei sollen ebenfalls realistische Zukunftsszenarien beschrieben werden, die Auswirkungen auf das Statemate-Modell haben. Damit sollen die Studierenden Weiterentwicklungsmöglichkeiten der Innovation identifizieren, deren Realisierung mit dem momentanen Stand der Technik (vorgegeben durch das technische Umfeld, das von den Betreuern bereit gestellt wurde) nicht machbar sind.

5 Ergebnisse

In diesem Abschnitt beschreiben wir nun die Ergebnisse des Experiments. Zunächst geben wir einen Überblick über den unterschiedlichen Verlauf der Aufwände und diskutieren, wie sich die Unterschiede erklären lassen (Frage 1 und 3 des Experiments in Abschnitt 2.4). Im Anschluss daran diskutieren wir anhand der erhobenen Daten das Verständnis der Studierenden in den

einzelnen Iterationsstufen (Frage 2) und gehen dann auf die von den Studierenden selbst geäußerten Anmerkungen und Erfahrungen ein (Frage 4).

Wie in Abschnitt 4.1 bereits erwähnt, verwendeten wir eine Web-Anwendung, um die Aufwände der Studierenden zu erfassen. Damit im Nachhinein zwischen den verschiedenen Aufwänden unterschieden werden konnte, definierten wir Kategorien, denen die Studierenden ihre aufgewendete Zeit zuordnen mussten. Die Kategorien teilten sich auf in Aufwände, die unabhängig von der gewählten Vorgehensweise sind (Verständnis, Kommunikation und Klärung von Fragen, Sonstiges) und solche, die speziell nur in unserem Experimentsetting auftauchen. Dabei wurde für jede Tätigkeit für jeden Dokumenttyp eine eigene Kategorie definiert. Die möglichen Tätigkeiten umfassen jeweils Aufwand, um das entsprechende Dokument zu erstellen, es zu überarbeiten (aufgrund von Fehlern, Anmerkungen der Betreuer etc.) und schließlich ein fremdes Dokument (von einer anderen Gruppe) an die neuen Anforderungen anzupassen und zu ändern. Tabelle 2 gibt einen Überblick über alle Kategorien.

Kürzel	Beschreibung
V	Aufwand für Verständnis (z.B. Lesen der Anforderungen, Überlegungen)
K	Aufwand für Kommunikation und (Termin-)Absprachen innerhalb der Gruppe
F	Aufwand für Nachfragen bei den Betreuern
EPP	Aufwand zur Erstellung des Projektprotokolls
ÜPP	Aufwand für die Überarbeitung des Projektprotokolls aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄPP	Aufwand für die Änderung/Anpassung von Teilen eines fremden Projektprotokolls aufgrund der neuen Anforderungen
EAD	Aufwand, um das Anforderungsdokument auszufüllen
ÜAD	Aufwand für die Überarbeitung des Anforderungsdokuments aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄAD	Aufwand für die Änderung/Anpassung von Teilen eines fremden Anforderungsdokuments aufgrund geänderter Anforderungen
EKD	Aufwand für die Modellierung eines neuen Kontextdiagramms (ausdenken und zeichnen)
ÜKD	Aufwand für die Überarbeitung des Kontextdiagramms aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄKD	Aufwand für die Änderung/Anpassung eines fremden Kontextdiagramms aufgrund der neuen Anforderungen
ES	Aufwand für die Erstellung neuer Szenarien (ausdenken und aufschreiben)
ÜS	Aufwand für die Überarbeitung der Szenarien aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄS	Aufwand für die Änderung/Anpassung von fremden Szenarien aufgrund geänderter Anforderungen
EEM	Aufwand für die Erstellung eines Entscheidungsmodells (ausdenken und zeichnen)
ÜEM	Aufwand für die Überarbeitung eines Entscheidungsmodells aufgrund der Hinweise der Betreuer, Fehler u. ä.

ÄEM	Aufwand für die Änderung/Anpassung von Teilen eines fremden Entscheidungsmodells aufgrund geänderter Anforderungen
ETF	Aufwand für die Erstellung der Testfälle (ausdenken und aufschreiben)
ÜTF	Aufwand für die Überarbeitung der Testfälle aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄTF	Aufwand für die Änderung/Anpassung von Teilen der fremden Testfälle aufgrund der neuen Anforderungen
ESC	Aufwand für die Modellierung der Statecharts (ausdenken und zeichnen)
ÜSC	Aufwand für die Überarbeitung der Statecharts aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄSC	Aufwand für die Änderung/Anpassung von Teilen der fremden Statecharts aufgrund geänderter Anforderungen
DEA	Aufwand für die Einarbeitung in die neuen (fremden) Dokumente
DQS	Aufwand für die Durchführung der Qualitätssicherung der fremden Modelle gemäß vorgegebenen Template (nicht Testfälle testen!)
S	sonstiger Aufwand, der nicht durch die anderen Kategorien abgedeckt wird

Tabelle 2: Überblick über die eingesetzten Kategorien bei der Aufwandserfassung.

Allgemein muss über die Aufwandsauswertung gesagt werden, dass die gemachten Aussagen aufgrund der geringen Gruppengröße und der teilweise starken individuellen Unterschiede zwischen den Studierenden nicht signifikant sein können. Neben diesen Faktoren kommt noch hinzu, dass teilweise eine Studierendengruppe die Aufgabe nicht korrekt verstanden hatte und deswegen einen unplausiblen, erheblich höheren Zeitaufwand hatte. Solche Ausreißer wurden bei der Berechnung der Durchschnittswerte ausgeklammert. Trotz dieser Ungenauigkeiten wird im Folgenden eine Interpretation der Ergebnisse versucht.

Um nun aus den erfassten Aufwänden der Studierenden Rückschlüsse zur Beantwortung der Frage nach der Intuitivität und Effizienz der benutzen Dokumentationsmethoden ziehen zu können, haben wir sie zunächst summiert und pro System pro Phase gemittelt. Im Anhang 7.8 sind diese Daten tabellarisch dargestellt. Für die Auswertung waren zwei Werte besonders interessant: Zum einen der Aufwand für die Erstellung der Dokumentation, zum anderen der Einarbeitungsaufwand in die fremden Dokumente. Unter dem Begriff „Dokumentationsaufwand für Szenarien“ wurden in der Projektprotokollgruppe die Aufwände für das Erstellen bzw. Bearbeiten des Projektprotokolls und der Szenarien (EPP, ÜPP, ÄPP, ES, ÜS, ÄS) zusammengefasst. Für die Entscheidungsmodell-Großgruppe (EG) kam dazu noch der Aufwand für die Erstellung der Entscheidungsmodelle (EEM, ÜEM, ÄEM). Der Aufwand für die Einarbeitung in die Dokumentation der vorherigen Iteration wurde direkt über DEA abgegriffen.

Zunächst wird der prozentuale Mehraufwand der Entscheidungsmodellgruppen betrachtet. Dass die Entscheidungsmodellgruppen einen höheren Aufwand hatten, ist leicht einzusehen, da sie ja die Szenarien nicht nur textuell festzuhalten hatten, sondern diese eben in das grafische Entscheidungsmodell einpflegen mussten. Die Tabellen 3 bis 5 zeigen die gemessenen und gemittelten Aufwände.

Wie zu erwarten, haben die EG einen teilweise erheblich höheren Aufwand bei der Erstellung der Dokumentation. Der Mehraufwand schwankt stark zwischen -16 % und +94 %

Iteration	Aufwand für	PG [min]	EG [min]	Diff.
1	Dokumentation Szenarien	200	383	+92 %
2	Einarbeitung	50	80	+60 %
	Dokumentation Szenarien	143	120	-16 %
3	Einarbeitung	90	120	+33 %
	Dokumentation Szenarien	195	195	0 %
Summe		678	898	+32 %
Gesamtaufwand		4658	3641	-22 %
Qualität		- / -	+	

Tabelle 3: Aufwandsvergleich für das Türschloss-System.

innerhalb eines Systems. Auf alle Fälle erkennt man, dass die Dokumentation mit Entscheidungsmodellen mehr Zeit beansprucht. Warum gerade bei der zweiten Iteration im Türschloss-System ein umgekehrter Effekt auftritt, könnte folgendermaßen erklärt werden: Wie der Tabelle 3 zu entnehmen ist, wurde in der ersten Iteration besonders viel Aufwand für die Erstellung der Szenarien betrieben. Es wäre denkbar, dass durch die gründliche Bearbeitung der ersten Iteration bereits schon viele Szenarien angedacht waren und somit das allgemeine System-Verständnis deutlich höher war. Darauf aufbauend konnten zusätzliche Szenarien in der zweiten Iteration dann leichter eingepflegt werden.

Abgesehen von diesem einen Fall, kann der höhere Aufwand als mögliches Einsparpotential der Entscheidungsmodellgruppen gesehen werden. Wenn, bedingt durch diesen Mehraufwand, genau so viel Zeit bei der Einarbeitung in die fremden Dokumente eingespart werden kann, dann wäre diese Art der Dokumentation nicht nur hilfreich sondern sogar effizienter als das herkömmliche Vorgehen.

Iteration	Aufwand für	PG [min]	EG [min]	Diff.
1	Dokumentation Szenarien	250	290	+16 %
2	Einarbeitung	80	60	-25 %
	Dokumentation Szenarien	245	310	+27 %
3	Einarbeitung	40	30	-25 %
	Dokumentation Szenarien	85	140	+65 %
Summe		700	830	+19 %
Gesamtaufwand		4002	2960	-26 %
Qualität		+ / +	+ / 0	

Tabelle 4: Aufwandsvergleich für das Spurassistentz-System.

Wie aus den Tabellen abzulesen, wurde bei dem Spurassistentz-System und dem Airbag tatsächlich Zeit bei der Einarbeitung in die fremden Dokumente eingespart, wenn auch nicht in der gleichen Höhe wie vorher für die Erstellung aufgewendet werden musste. Warum ausgerechnet das Türschloss wieder aus dem Rahmen fällt, bleibt unklar. Eine mögliche Erklärung hierfür wäre, dass das Türschloss in der ersten Iteration bereits so ausführlich und umfangreich beschrieben wurde, dass sich aufgrund des großen Umfangs ein höherer Einarbeitungsaufwand ergab. Dieser Eindruck verstärkt sich, wenn die absoluten Zeiten betrachtet werden. Dabei

wird deutlich, dass es sich jeweils um 30 Minuten Mehraufwand handelt, der, gemessen am tatsächlichen Erstellungsaufwand, doch vergleichsweise gering ausfällt. Auch ein Blick auf die Entscheidungsmodelle zeigt, dass diese tatsächlich sehr ausführlich erstellt worden sind.

Vergleicht man den Aufwand für die Erstellung und das Verstehen der Dokumentation insgesamt über die drei Iterationen hinweg, erkennt man, dass der Mehraufwand der Entscheidungsmodellgruppen nur noch Werte zwischen 13 % und 32 % einnimmt. Dieser Mehraufwand hält sich in Grenzen und kann vor allem in Hinblick auf zukünftige Entwicklungen als vertretbar angesehen werden.

Sehr interessant ist der Vergleich zwischen dem Gesamtaufwand, d. h. Dokumentationsaufwand zuzüglich dem Erstellungsaufwand für die Anforderungsdokumente, die Kontext- und Statemediagramme und schließlich der Testfälle pro System. Dabei schneiden die Entscheidungsmodellgruppen trotz besserer oder zumindest vergleichbarer Qualität durchwegs besser ab als die Projektprotokollgruppen. Dies lässt sich auf ein tieferes Systemverständnis und einer daraus resultierenden besseren Planung zurückführen. Dadurch wurde eine übersichtlichere Architektur der Systeme erstellt, was sich in einem geringeren Zeitaufwand bei der Fehlersuche und System-Änderung in den nachfolgenden Iterationen niederschlägt.

Iteration	Aufwand für	PG [min]	EG [min]	Diff.
1	Dokumentation Szenarien	150	178	+19 %
2	Einarbeitung	120	60	-50 %
	Dokumentation Szenarien	260	300	+15 %
3	Einarbeitung	60	60	0 %
	Dokumentation Szenarien	85	165	+94 %
Summe		675	763	+13 %
Gesamtaufwand		3231	3608	+12 %
Qualität		+ / -	- / -	

Tabelle 5: Aufwandsvergleich für das Airbag-System.

Schließlich ist es noch interessant, die Veränderung des Szenariendokumentationsaufwands innerhalb eines Systems zu betrachten. Im Airbag- und Spurassistenten-System ist er in der ersten Iteration deutlich höher als in der zweiten; beim Türschloss-System ist dies genau anders herum. Eine mögliche Erklärung greift die oben angefangene Diskussion auf: Das Türschloss ist zwar ein kompliziertes aber für die Studierenden bekanntes System. Daher hatten sie viele Einfälle für neue Szenarien, die in der ersten Iteration bereits dokumentiert wurden. Daher fiel es der Gruppe in der zweiten Iteration deutlich leichter, weitere Szenarien zu finden. Das Spurassistenten-System dagegen ist zwar relativ leicht zu modellieren aber dafür den Studierenden unbekannt. Daher war es nicht einfach, sich neue Szenarien auszudenken und zu begründen. Der Airbag wiederum verhält sich ähnlich zum Spurassistenten-System. Er ist leicht zu modellieren, die eigentliche Funktionsweise ist den Experimententeilnehmern allerdings eher unbekannt. Bei diesem System kommt außerdem hinzu, dass es im Nachhinein betrachtet, aufgrund der zu geringen Funktionsvielfalt eigentlich ungeeignet war, im Rahmen dieses Experiments verwendet zu werden. Die wesentlichen Neuerungen bezogen sich auf neue Hardware und weniger auf interne Logik. Daher finden sich auch in den Entscheidungsmodellen keine Mehrinformationen als in den Projektprotokollen der anderen Großgruppe. Die Qualität der Statecharts schneidet dabei auch schlechter ab als die der Projektprotokollgruppen.

Wiederum lässt sich dies auf das mangelnde System-Verständnis zurückführen, welches durch gute Entscheidungsmodelle unterstützt werden würde. Dies war bei diesem System aber kaum möglich, da die Design Optionen hauptsächlich Hardwareorientiert waren.

Erfahrene Ingenieure hätten wohl gesagt, dass die Anwendung eines solchen Entscheidungsmodells auf ein System wie den Airbag nicht anwendbar ist, weil der Airbag wenig Variabilität aufweist und damit eine Erschließung des künftigen Verhaltens nicht sehr viel Mehrwissen mit sich bringen wird. Die Studierenden allerdings waren von uns beauftragt, dieses Modell auch auf den Airbag anzuwenden und so haben sie sich zu sehr damit beschäftigt, was aus dem Airbag werden kann und haben sich damit den Blick auf das Wesentliche in diesem Fall verstellt. Die Verwendung jeglicher Modelle muss vor deren Einsatz auf Tauglichkeit in dieser speziellen Domäne analysiert werden. Nur dann kann man von einer erfolgreichen Nutzung ausgehen.

Zur Beantwortung der Frage 2 aus Abschnitt 2.4 haben wir das Experiment so gestaltet, dass die Studierenden in der letzten Woche des Praktikums, basierend auf den ihnen vorliegenden Dokumenten bezüglich des von ihnen zuletzt entwickelten Systems in den drei Iterationen, ein Merkmaldiagramm erstellen sollten, aus dem die bisherige und gegebenenfalls die zukünftige Weiterentwicklung des Systems in den unterschiedlichen Varianten hervorgeht. Trotz einer intensiven Schulung der Studierenden und einer ausführlichen Rückmeldung nach Abgabe der Merkmaldiagramme sowie einer daran anschließenden Überarbeitung können die uns vorliegenden Merkmaldiagramme zu einer Auswertung nicht verwendet werden: Aus den Merkmaldiagrammen ist ersichtlich, dass das Prinzip der Funktionsorientierung nicht verstanden wurde und auch die vorhandenen Mittel, wie beispielsweise die Beschreibung der Abhängigkeiten, konnte nicht in einen adäquaten Zusammenhang mit dem Ziel dieses Variabilitätsmodells gebracht werden. Nach Auswertung der Angaben der Studierenden in den Interviews haben wir festgestellt, dass die Studierenden nach sechs Wochen ein gutes Verständnis der Zustandsorientierung entwickelt haben, wie es für Statemate notwendig ist. Innerhalb einer Woche war es für die Studierenden daher nicht möglich, so folgerten wir, das ihnen neue Paradigma der Funktionsorientierung zu erlernen: Die Merkmaldiagramme stellen sich somit auch als eine Mischung von Funktionen und Zuständen dar.

Um dennoch Hinweise aus dem Experiment für die Beantwortung der Frage 2 zu verwenden, betrachten wir im Folgenden die von den Studierenden zu erstellenden Szenarien, um festzustellen, inwieweit sich Verständnis von Statemate und dem zu entwickelnden System manifestiert. Die beiden im Folgenden angegebenen Tabellen lassen sich aus der Tabelle im Anhang 7.9 ableiten.

In Tabelle 6 haben wir dargestellt, welchen prozentualen Anteil die von den Studierenden gefundenen Statemate-relevanten Szenarien von den jeweils pro Iteration geforderten vier Szenarien einnehmen. Jede Kleingruppe beider Großgruppen musste sich zu jeder Iteration 4 neue Szenarien überlegen, die realistisch sind, in einer zukünftigen Weiterentwicklung des Systems einzutreten. Die vorliegende Tabelle 6 beschreibt nun den Anteil der Szenarien, die eine Relevanz in Statemate haben. Szenarien, die nur neue Sensoren oder andere Hardware-Aspekte des Systems betreffen, waren im Rahmen dieses Experiments nicht interessant, weil dies in Statemate so gut wie keine Auswirkung hätte.

Der tendenzielle Verlauf stellt sich für beide Großgruppen gleich dar: In der ersten Iteration finden die Kleingruppen in beiden Großgruppen durchschnittlich zwei Statemate-relevante Szenarien von den geforderten vier Szenarien. Von der ersten Iteration zur zweiten verbessern sich die Gruppen und können mehr Statemate-relevante Szenarien identifizieren. Dies lässt sich durch den Verständnissgewinn bezüglich Statemate erklären, der sich in der zweiten Itera-

	prozentualer Anteil Statemate-relevanter Szenarien der insgesamt vier geforderten Szenarien pro Iteration, Durchschnittswert über alle Kleingruppen der PG	prozentualer Anteil Statemate-relevanter Szenarien der insgesamt vier geforderten Szenarien pro Iteration, Durchschnittswert über alle Kleingruppen der EG
Iteration 1	50%	50%
Iteration 2	58%	54%
Iteration 3	46%	38%

Tabelle 6: Überblick normiertes Verständnis Statemate relevanter Szenarien für beide Großgruppen.

tion einstellt (siehe auch weiter unten die Aussagen der Studierenden selbst zur Entwicklung des Verständnisses). In der dritten Iteration erkennt man eine klare Verschlechterung, sogar unter das Niveau der ersten Iteration. Dies lässt sich dadurch erklären, dass die Studierenden mittlerweile kaum mehr Ideen hatten, wie es mit dem System weiter gehen könnte: Die Kreativität geht den Studierenden aus. Dieser tendenzielle Verlauf ist zwar bei beiden Großgruppen gleich, allerdings ist der Verlauf der Projektprotokoll-Gruppe (mit Ausnahme der ersten Iteration) besser als der der Entscheidungsmodell-Gruppe. Dies könnte einen Hinweis darauf geben, dass die Projektprotokoll-Gruppe ein besseres Verständnis von Statemate sowie eine höhere Kreativität/Motivation bezüglich der Findung neuer Szenarien hat.

Iteration	System	prozentualer Anteil in späterer Iteration umsetzbarer Szenarien als Anforderungen der über beide Großgruppen hinweg meisten gefundenen Statemate-relevanten Szenarien pro Iteration, Durchschnittswert über alle Kleingruppen der PG	prozentualer Anteil in späterer Iteration umsetzbarer Szenarien als Anforderungen der über beide Großgruppen hinweg meisten gefundenen Statemate-relevanten Szenarien pro Iteration, Durchschnittswert über alle Kleingruppen der EG
1	Türschloss	33%	0%
	Spurassistenz	0%	33%
	Airbag	67%	33%
2	Türschloss	60%	60%
	Spurassistenz	0%	75%
	Airbag	17%	67%

Tabelle 7: Überblick normiertes Verständnis kommender Anforderungen durch Szenarien für beide Großgruppen.

Betrachtet man vor dem Hintergrund der Ergebnisse aus Tabelle 6 aber die Einschätzung

der Studierenden bezüglich der Realität, also welche identifizierten Szenarien in einer kommenden Iteration technisch möglich oder gefordert werden, so stellt man fest, dass tatsächlich die Entscheidungsmodell-Gruppe in der zweiten Iteration ein weit besseres Verständnis als die Projektprotokoll-Gruppe entwickelt.

In Tabelle 7 haben wir dargestellt, wie sich das Verständnis bei der Identifikation künftiger Anforderungen durch Statemate relevante Szenarien für die beiden Großgruppen entwickelt. Die vorliegende Tabelle 7 beschreibt das Verhältnis der in einer kommenden Iteration als Anforderung umzusetzenden Szenarien pro Iteration zu der höchsten pro System und Iteration gefunden Anzahl an Statemate-relevanten Szenarien. 100% bedeutet damit, dass die in der Iteration für das System höchste gefundene Anzahl an Statemate-relevanten Szenarien auch in einer der kommenden Iterationen vollständig als Anforderungen realisiert werden musste. 0% bedeutet, dass kein einziges Szenario in der entsprechenden Iteration in einer künftigen Iteration als Anforderung umgesetzt werden musste. Die Werte stellen den Durchschnitt über alle Kleingruppen je System dar. Die Tabelle betrachtet nur die ersten beiden Iterationen, weil nach der dritten Iteration keine weitere folgte, so dass in der dritten Iteration keine Aussage gemacht werden kann, welche Szenarien in einer kommenden Iteration zu Anforderungen werden konnten.

Diese Werte beschreiben die realistische Einschätzung der Studierenden bezüglich der System-Variabilität. Natürlich ist es so, dass die Betreuer an dieser Stelle einen nicht zu vernachlässigenden Einfluss haben, da es letztlich deren Einschätzung der Realität ist, die hier als realistisch angesehen wird. Um an dieser Stelle Störvariablen auszuschließen, haben wir zwei Dinge beachtet: Zunächst wurden alle Inhalte bezüglich Ausgestaltung der Iterationen und der Systeme vor Beginn des Praktikums abgeschlossen, so dass keine Ideen der Studierenden von den Betreuern aufgenommen werden konnten, womit ungerechtfertigte Präferenzierung einer oder mehrerer Gruppen einhergehen würde. Bei der Ausgestaltung der Inhalte haben wir weiterhin darauf geachtet, keine utopischen Ideen umzusetzen, sondern systematisch das System mit seinen Möglichkeiten und Schwächen analysiert und entsprechend Verbesserungen und Erweiterungen erarbeitet. Als weiteren Mechanismus zum Ausschalten einer Störvariablen haben wir den Studierenden die Freiheit gelassen, bei einer technischen Möglichkeit, auch eigene Anforderungen umzusetzen: Da wir zu jeder Iteration den Studierenden nur eine grobe Beschreibung der geforderten Merkmale gaben, waren sie in der Pflicht, eigene Ideen zu generieren, was wir unter anderem durch das Verfassen der Pressemitteilungen unterstützen wollten. Damit waren auch von den Betreuern nicht vorgesehene Anforderungen in späteren Iterationen möglich, die aus beschriebenen Szenarien resultierten.

Die Tabelle 7 beschreibt nun einen interessanten Verlauf: Während in der ersten Iteration die Projektprotokoll-Gruppe bessere Werte als die Entscheidungsmodell-Gruppe aufweist, was sich durch das in Tabelle 6 festgestellte bessere Verständnis von Statemate sowie der höheren Kreativität erklären lässt, verbessert sich die Entscheidungsmodell-Gruppe merklich in der zweiten Iteration, während sich die Projektprotokoll-Gruppe sogar ein wenig verschlechtert. Dies lässt sich durch einen Lerneffekt erklären, der bei der Entscheidungsmodell-Gruppe eintritt, nicht aber bei der Projektprotokoll-Gruppe. Tatsächlich sind die Entscheidungsmodelle also ein geeignetes Instrument, das Verständnis bezüglich der System-Variabilität sowie eine realistische Einschätzung der Weiterentwicklung des Systems zu unterstützen, da das bisherige Wissen leicht verständlich und im Überblick dargestellt ist. Leider lässt sich aufgrund der drei durchgeführten Iterationen der Trend nur in der einen Iteration ersehen und kann nicht in einer weiteren Iteration bestätigt werden.

Wichtig in diesem Zusammenhang war es, dass wir Studierende als Probanden zur Verfü-

gung hatten: Erfahrene Ingenieure hätten wohl nicht eine solche Lernkurve aufweisen können, da sie sicherlich mit der Nutzung von Entscheidungsmodellen vertraut gewesen wären. Insofern war es für dieses Experiment ein Gewinn, Probanden mit wenig Erfahrung in der Software-Entwicklung zu haben, da auf diese Weise schneller die Vorteile und die Anwendbarkeit der untersuchten Ansätze zutage traten.

Ein weiteres wichtiges Ergebnis bei der Auswertung des Experiments betrifft den Umgang der Studierenden mit den vorgeschlagenen Entscheidungsmodellen. Ein Großteil der Studierenden hat die Argumentation auf das System *Fahrzeug* bezogen. Auch die Hinweisschilder werden als systemspezifische Trigger angesehen, die die Weiterentwicklung auf detaillierter Hardware-Ebene vorwegnehmen (siehe Anhang 7.4). Die Studierenden haben insofern recht mit diesem Umgang der Modellierung, dass die Domäne und die Lösung nicht unabhängig voneinander sind. Aber die wesentliche Stärke der Hinweisschilder beispielsweise ist deren Verhalten als Trigger aus der umgebenden Welt; die Studierenden verwenden sie allerdings als interne Kommunikationsschleife im Entwurfsprozess. Dies legt nahe, dass die Studierenden den Umgang von Ideen im weiten Kontext des *System Engineering* nicht gewohnt sind und sich dieser Umgang auch nicht auf die Schnelle erlernen lässt. Eine gedankliche Trennung zwischen der Entwicklung eines Systems als Ganzes sowie dessen Einbettung in eine Umwelt und den internen Details der System-Zustände ist mithin eine wesentliche Voraussetzung für eine erfolgreiche System-Entwicklung.

Ein anderer Teil der Studierenden hat die Entscheidungsmodelle auch auf eine eigene Art und Weise verwendet: Die Vermutungen in der Argumentation im Modell wurden als Szenarien beschrieben. Dabei wurden allerdings hilfreiche Erkenntnisse entwickelt, die sie bei der System-Entwicklung nutzen konnten. Mit anderen Worten haben die Studierenden erkannt, dass es hilfreich ist, Szenarien als Begründung für Teil-Systeme zu nutzen. Die Entscheidungsmodelle finden sich im Anhang im Kapitel 7.4.

Zur Unterstützung der bisher geschilderten Ergebnisse lassen sich nun die Ergebnisse der wöchentlichen Befragungen der Studierenden heranziehen. Im Folgenden gehen wir genauer auf die als von uns für wesentlich erachteten Aussagen der Studierenden ein. Die vollständigen Fragen und Antworten sind im Anhang unter 7.10 enthalten.

Wir ordnen die Aussagen der Studierenden in fünf Kategorien: Reaktion: Motivation, Einschätzung eigener Methodenkompetenz; Reflektion: Verbesserungsvorschläge, Lernkurve und Verständnis des Vorgehens.

Zur Motivation äußern sich die Studierenden über die Großgruppen hinweg recht ähnlich. Vor allem wird die Bedeutung von Statemate herausgehoben und der erste Motivationsschub ergab sich bei der ersten Verwendung von Statemate: PG, 11.05.05: „Bis man mit dem Modell angefangen hat, ging es schleppend, danach motivierter“, EG, 18.05.05: „Statemate macht Spaß“. Eine weitere Motivationssteigerung ergab sich durch die zweite Iteration, die „wesentlich mehr Spaß gemacht hat“ (EG, 15.06.05). Dies lag vor allem daran, dass die Studierenden nun mit dem Vorgehen besser vertraut waren. Auch an dem besseren Verständnis der Weiterentwicklung des Systems (siehe oben) kann man diesen Verständniszuwachs ablesen.

Zur Einschätzung der eigenen Methodenkompetenz: Recht früh werden Anforderungsdokumente und Testfälle als verstanden angesehen, während Kontextdiagramme als unklar erachtet werden. Statemate und die Zustandsorientierung wird sechs Wochen nach Beginn des Experiments als verständlich angesehen: (PG, 15.06.05): „Umsetzung in Zustände werden langsam besser begriffen“. In der dritten Iteration wird angesprochen, was wir oben schon bei der Anzahl Statemate relevanter Szenarien in der dritten Iteration festgestellt haben: die Kreativität geht den Studierenden langsam aus: (PG, 22.06.05): „Szenarien werden zum Teil

lustig“.

Die von den Studierenden geäußerten Verbesserungsvorschläge bezogen sich vor allem auf die Umsetzung ihrer täglichen Arbeitsschritte, es wurden keine Vorschläge derart gemacht, dass das Vorgehen einer bestimmten Methode für die Ansprüche des Experiments geändert werden müssten. So wurde angeregt, dass der Austausch der Dokumente zwischen den Studierenden zu Beginn einer neuen Iteration „nicht über E-Mail stattfinden sollte“ (EG, 25.05.05), um bei einer persönlichen Übergabe gleichzeitig eine Kurzeinführung in das Modell geben zu können und für die Beantwortung von Fragen zur Verfügung zu stehen. Als didaktischen Verbesserungsvorschlag wurden sich mehr Beispiele vor Beginn des Experiments gewünscht (PG, 15.06.05).

Zur Lernkurve haben sich die beiden Großgruppen in der letzten Woche der dritten Iteration sehr ähnlich geäußert (PG/EG, 29.06.05). Durch das Praktikum konnten sie die zustandsorientierte Modellierung in der Praxis anwenden, was die bisherigen theoretischen Kenntnisse vertiefte. Weiterhin betonte die Entscheidungsmodell-Gruppe den Aspekt, dass Teile wiederverwendet werden konnten und dass bei der Eigenentwicklung von Systemen auch darauf zu achten ist, diese so zu gestalten, dass eine andere Gruppe darauf weiter arbeiten kann. Dies mag ein Hinweis für die oben festgestellten geringeren Gesamtaufwände sein.

Zum Verständnis des Vorgehens äußerten sich die Studierenden recht häufig (PG, 11.05.05): „Wozu dient das Projektprotokoll?“ fragte die Projektprotokoll-Gruppe in der ersten Iteration und antwortete zwei Wochen später selbst darauf (PG, 25.05.05): „Zum Einarbeiten ist das Projektprotokoll und das Anforderungsdokument geeignet“; gleichzeitig erkennt die Projektprotokoll-Gruppe, dass das Projektprotokoll nicht ausreicht: „Szenarien mit Auswirkungsänderungen sind im Kopf klar, aber es gibt keine Möglichkeit, dies zu dokumentieren“, was letztlich den Bedarf für so etwas wie die Entscheidungsmodelle (die den Studierenden der Projektprotokoll-Gruppe zu diesem Zeitpunkt nicht bekannt waren) andeutet. Auch die Entscheidungsmodell-Gruppe erkennt die Notwendigkeit der Szenarien: (EG, 25.05.05): „Szenarien und Entscheidungsmodell waren wichtig um sich Gedanken zu machen“. Damit haben die Studierenden den Bedarf an Requirements Engineering Methoden erkannt und selber formuliert. In diesem Zusammenhang ist es noch einmal wichtig zu betonen, dass die beiden Aussagen der Studierenden in zwei Gruppen, getrennt voneinander, zur selben Zeit abgegeben wurden. Weiterhin kannte die Projektprotokoll-Gruppe das Vorgehen der Entscheidungsmodell-Gruppe nicht. Spannend an dieser Erkenntnis der Studierenden ist, dass die Studierenden tatsächlich nur zwei der typischen Motivationen für das Requirements Engineering antrafen:

- Fluktuation der Mitarbeiter (Wechsel der Dokumente und Verantwortung für ein neues Projekt)
- Änderung und Erweiterung der Anforderungen (angestoßen durch die Wünsche der Betreuer)
- KEINE geografische Verteilung
- NICHT sehr große Systeme
- KEINE lange Lebensdauer der Systeme (mit dem Bedarf der Wartbarkeit)

Und dennoch hat sich der Wert des Requirements Engineering den Studierenden erschlossen, speziell der Bedarf einer Unterstützung für künftige Wiederverwendung bei der Entwicklung von Innovationen.

6 Kritische Diskussion des Experiments und Ausblick

Das Experiment gibt Hinweise, dass die frühzeitige Verwendung von Entscheidungsmodellen eine zielorientierte Dokumentation von Wissen, das für das Domain Engineering benötigt wird, darstellt. Da die Entscheidungsmodelle leicht zu erlernen sind und die darin abgebildeten Inhalte im Wesentlichen nicht über die Inhalte hinaus gehen, die typischerweise bei der Entwicklung von Systemen anfallen, stellen sie eine geeignete und nicht aufwändige Form der Dokumentation dar, die für ein kontinuierliches und proaktives Domain Engineering genutzt werden kann. Da die von uns vorgestellten Entscheidungsmodelle mögliche Entwurfsoptionen abbilden und damit nicht den Anspruch haben, den aktuellen Stand der Realität abzubilden, können Sie auch bei der Entwicklung von Innovationen eingesetzt werden, die von bisheriger Forschung bezüglich Wiederverwendung größtenteils ausgeklammert wurden, weil zum Zeitpunkt der Entwicklung von Innovationen noch nicht wiederverwendet werden kann; vielmehr bezogen sich die bisherigen Bestrebungen darauf, die System-Entwicklungsartefakte so in Komponenten zu zerlegen, dass sie sinnvoll wiederzuverwenden sind. Um aber das System in seinen Komponenten so zu gestalten, dass Teile davon sinnvoll wiederverwendet werden können, muss man wissen, wie es sich in der Zukunft ausgestaltet. Dies aber wurde bisher nicht systematisch unterstützt. Mit den Entscheidungsmodellen schlagen wir einen Ansatz vor, der diese Diskrepanz verringert.

Der Ablauf des Experiments verlief erfolgreich. Bis auf die unvorhergesehene Problematik am Ende des Experiments, wo den Studierenden nicht innerhalb einer Woche die Merkmaldiagramme zu vermitteln waren und wir deswegen bei der Betrachtung des Verständnisses der Studierenden bezüglich der System-Variabilität auf die Szenarien selbst zurückgreifen mussten, gestaltete sich der Verlauf des Experiments so, wie wir es geplant haben. Auch die Rückmeldungen der Studierenden bezüglich der verwendeten Methoden und der Arbeitsbelastung waren insoweit positiv, dass trotz anfänglicher Unklarheiten die Arbeiten als sinnvoll und die Aufwände im Rahmen erachtet wurden.

Dennoch weist das Experiment einige Limitationen auf, die wir im Folgenden diskutieren werden. Zunächst ist in Hinblick auf die Auswertung der Ergebnisse die Begrenzung auf drei Iterationen bedauerlich: Der Trend, der sich in der zweiten Iteration bei der Entscheidungsmodell-Gruppe abzeichnet, nach dem sich das Verständnis bezüglich realistischer Abschätzung der künftigen System-Variabilität stark verbessert (siehe Abschnitt 5), lässt sich nicht durch eine weitere Iteration bestätigen, da uns keine mehr zur Verfügung steht. Nimmt man allerdings die Beschränkung auf ein Semester hin, so hätten wir entweder eine kürzere Trainingsphase oder kürzere Iterationszyklen realisieren müssen, um eine weitere Iteration zu ermöglichen. Mit einer kürzeren Trainingsphase wäre eine schlechtere Vorbereitung der Studierenden verbunden gewesen, die in Anbetracht der Aussage der Studierenden, dass sie sich in der Trainingsphase noch mehr Beispiele gewünscht hätten, dann nicht mehr als Basis für das Experiment gereicht hätte. Kürzere Iterationszyklen hätten zur Folge gehabt, dass die Studierenden nur jeweils sehr kleine Aspekte in den Systemen, die sie jeweils betreuen, hätten umsetzen können, worunter die Übertragbarkeit der Ergebnisse gelitten hätte. So lässt sich der identifizierte Trend als Ergänzung und Unterstützung der anderen Hinweise sehen, die in eine ähnliche Richtung deuten.

Weiterhin schlagen wir vor, die Entscheidungsmodelle direkt in dem Werkzeug zu beschreiben, in dem auch die Spezifikationen selbst beschrieben werden. Dies wäre in DOORS [Tel] möglich. Im Fall dieses Experiments allerdings haben die Studierenden die Anforderungen in einer Excel-Tabelle und die Entscheidungsmodelle in PowerPoint erstellt. Da PowerPoint die

Modellierungsarbeit praktisch nicht unterstützt, gab es allerhand Schwierigkeiten, vor allem was die Erweiterbarkeit der Modelle betrifft. Wir gehen davon aus, dass hier das Prinzip, alles in einem Werkzeug zu verwalten, einen Vorteil in der Praxis mit sich bringt. Allerdings konnten wir die tatsächliche Anwendung in diesem Fall nicht im Experiment untersuchen.

In diesem Zusammenhang wäre es sicherlich hilfreich gewesen, bei den Kategorien der Aufwandserfassung zu unterscheiden, wie viel Zeit die Studierenden mit dem reinen Ausdenken und Entwerfen der Dokumente verbracht haben und wie hoch der Aufwand war, diese Skizzen in dem jeweiligen Werkzeug festzuhalten. Damit wäre es möglich gewesen, zum Einen Schwierigkeiten im Umgang des Systems detaillierter zu betrachten und evtl. herauszurechnen, zum Anderen genauere Hinweise auf eine fehlende aber nützliche Werkzeugunterstützung zu erhalten. Dies lässt sich mit den vorhandenen Daten nicht adäquat belegen.

Auch die Auswahl der Systeme war im Nachhinein betrachtet nicht ganz glücklich. Die Funktionalität des Türschloss-Systems konnte sehr gut ohne Änderung der umgebenden Hardware verändert und erweitert werden. Beim Airbag zum Beispiel war dies nur sehr bedingt möglich. Daher hatten die Studierenden auch größere Schwierigkeiten, sich gute Szenarien für dessen Weiterentwicklung auszudenken, was sich unmittelbar wieder auf die Qualität des Experiments auswirkt. Daher empfehlen wir, bei einer Wiederholung des Experiments darauf zu achten, dass sich die verwendeten Systeme bezüglich des Umfangs stärker gleichen und möglichst viel Spielraum bei der Ausgestaltung der Funktionalität ohne Änderung der Umgebung zulassen.

Wir konnten aus dem Experiment vor allem zwei Dinge, die die Didaktik betreffen, lernen: Wie oben schon beschrieben, war es nicht möglich, innerhalb einer Woche den Studierenden die funktionsorientierte Darstellung von Merkmaldiagrammen Nahe zu bringen, wenn wir ein Semester mit Ihnen die Zustandsorientierung angewendet haben. Wir können nicht sagen, ob es tatsächlich an diesen beiden unterschiedlichen Paradigmen lag oder ob nicht viel eher das nahe Ende des Semesters zu einer Verringerung der Motivation führte. Für künftige Experimente würden wir vorschlagen, die Merkmaldiagramme ebenso wie die anderen Techniken anfangs in der Trainingsphase einzuführen und dann zu jeder Iteration ein Merkmaldiagramm erstellen zu lassen. Mit jeder Iteration stieg das Verständnis und die Qualität der zu erstellenden Artefakte und dies würde sicherlich auch für die Merkmaldiagramme zutreffen. Eine weitere didaktische Erfahrung, die wir in diesem Experiment gemacht haben, betrifft die regelmäßigen Interviews der Studierenden. Anfangs war den Studierenden nicht so klar, warum wir die Befragungen durchführen, aber mit der Zeit und vor allem mit gestiegenem Bedürfnis sich mitzuteilen, weil man schlechte aber auch gute Erfahrungen gesammelt hat, nutzten die Studierenden die Ihnen gestellten Fragen, um untereinander Erfahrungen auszutauschen oder auch um Probleme gemeinsam zu klären. Um allerdings die Beteiligung der Studierenden noch zu erhöhen, schlugen wir für künftige Experimente vor, die Fragen der jeweiligen Woche neben der Aufwandserfassung für diese Woche zur Verfügung zu stellen und den Studierenden auch die Möglichkeit zu anonymen Antworten bereit zu stellen. Somit sind sie für die Fragerunde besser vorbereitet und haben gleichzeitig die Möglichkeit, ihre Anmerkungen in dem Moment den Betreuern mitzuteilen, wenn sie in der täglichen Arbeit aufkommt.

Trotz der aufgezeigten Grenzen des Experiments sind die Ergebnisse viel versprechend, da bisher in diesem Kontext keine empirische Forschung durchgeführt wurde. Aus dem Experiment kann man inhaltlich lernen, dass die Verwendung von argumentationsbasierten Entscheidungsmodellen kontinuierliches und proaktives Domain Engineering unterstützt und damit die Basis bereit stellt, Innovationen frühzeitig und mit weniger Aufwand in eine Software-Produktlinie einzubetten.

Literatur

- [ABM00] ATKINSON, C., J. BAYER und D. MUTHIG: *Component-Based Product Line Development: The Kobra Approach*. In: DONHOE, P. (Herausgeber): *Software Product Lines: Experiences and Research Directions*, Seiten 289–309, Norwell, 2000.
- [AM02] ATKINSON, C. und D. MUTHIG: *Enhancing Component Reusability through Product Line Technology*. In: GACEK, C. (Herausgeber): *Proceedings of the 7th International Conference on Software Reuse: Methods, Techniques, and Tools*, Seiten 94–108, Austin, 2002.
- [BCK03] BASS, L., P. CLEMENTS und R. KAZMAN: *Software Architecture in Practice*. The SEI Series in Software Engineering, Addison-Wesley, Boston, 2003.
- [BF03] BUSH, D. und A. FINKELSTEIN: *Requirements Stability Assessment Using Scenarios*. In: *Proceedings of the 11th IEEE International Requirements Engineering Conference*, Seiten 23–32, Monterey Bay, 2003.
- [Bos00] BOSCH, J.: *Design and Use of Software Architectures: Adopting and evolving a product-line approach*. Addison-Wesley, ACM Press Books, Harlow, 2000.
- [Bru01] BRULIN, G.: *The Third Task of Universities or How to Get Universities to Serve their Communities*. In: REASON, P. und H. BRADBURY (Herausgeber): *Handbook of Action Research: Participative Inquiry and Practice*, Seiten 440–446, London, 2001. SAGE Publications.
- [CN02] CLEMENTS, P. und L. NORTHROP: *Software Product Lines: Practices and Patterns*. The SEI Series in Software Engineering. Addison-Wesley, Boston, 2002.
- [DeM78] DEMARCO, T.: *Structured Analysis and System Specification*. Yourdon Press, Prentice-Hall, Englewood Cliffs, NJ, 1978.
- [Dew02] DEWAR, J.A.: *Assumption-Based Planning: A Tool for Reducing Avoidable Surprises*. Cambridge, 2002.
- [ES04] EISENECKER, U.W. und R. SCHILLING: *Merkmalmodellierung — Konzepte, Notationen und Einsatzmöglichkeiten*. In: *Konferenzband der Multikonferenz Wirtschaftsinformatik (MKWI)*, Seiten 157–168, Essen, 2004. MKWI-Teilkonferenz Software-Produktlinien.
- [Fli02] FLICK, U.: *An Introduction to Qualitative Research*. SAGE Publications, London, 2. Auflage, 2002.
- [Har87] HAREL, D.: *Statecharts: A Visual Formalism for Complex Systems*. *Science of Computer Programming*, 8:231–274, 1987.
- [HMT01] HEIN, A., J. MACGREGOR und S. THIEL: *Configuring Software Product Line Features*. In: PULVERMÜLLER, E., A. SPECK, J.O. COPLIEN, M. D’HONDT und W. DEMEUTER (Herausgeber): *Proceedings of the 8th Workshop on Feature Interaction in Composed System (ECOOP) in Association with the 15th European Conference on Object-Oriented Programming*, Seiten 67–69, Budapest, 2001. Universität Karlsruhe, Technical Report No. 2001-14.

- [HN96] HAREL, D. und A. NAAMAD: *The STATEMATE Semantics of Statecharts*. ACM Transactions on Software Engineering and Methodology, 5:4:293–333, 1996.
- [HP87] HATLEY, D.J. und I.A. PIRBHAI: *Strategies for Real-time System Specification*. Dorset House Publishing, 1987.
- [HR01] HERON, J. und P. REASON: *The Practice of Co-operative Inquiry: Research ‘with’ rather than ‘on’ People*. In: REASON, P. und H. BRADBURY (Herausgeber): *Handbook of Action Research: Participative Inquiry and Practice*, Seiten 179–188, London, 2001. SAGE Publications.
- [I-L] *I-Logix Statemate website*. <http://www.ilogix.com>.
- [KCH⁺90] KANG, K.C., S.G. COHEN, J.A. HESS, W.E. NOVAK und A.S. PETERSEN: *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. Technischer Bericht CMU/SEI-90-TR-21, Software Engineering Institute of the Carnegie Mellon University, Pittsburgh, 1990.
- [Lew46] LEWIN, K.: *Action research and minority problems*. Journal of Social Issues, 2:4:34–46, 1946.
- [LG01] LEVIN, M. und D. GREENWOOD: *Pragmatic Action Research and the Struggle to Transform Universities into Learning Communities*. In: REASON, P. und H. BRADBURY (Herausgeber): *Handbook of Action Research: Participative Inquiry and Practice*, Seiten 103–113, London, 2001. SAGE Publications.
- [MW03] MCNIFF, J. und J. WHITEHEAD: *Action Research: Principles and Practice*. RoutledgeFalmer, London, 2. Auflage, 2003.
- [Par01] PARK, P.: *Knowledge and Participatory Research*. In: REASON, P. und H. BRADBURY (Herausgeber): *Handbook of Action Research: Participative Inquiry and Practice*, Seiten 81–90, London, 2001. SAGE Publications.
- [Pre01] PRECHELT, L.: *Kontrollierte Experimente in der Softwaretechnik: Potenzial und Methodik*. Springer-Verlag, Berlin, 2001.
- [PTK04] PUSCHNIG, A. und R. TAVAKOLI KOLAGARI: *Requirements Engineering in the Development of Innovative Automotive Embedded Software Systems*. In: *Proceedings of the 12th IEEE International Requirements Engineering Conference*, Seiten 328–333, Kyoto, 2004.
- [Sch01] SCHEIN, E.H.: *Clinical Inquiry/Research*. In: REASON, P. und H. BRADBURY (Herausgeber): *Handbook of Action Research: Participative Inquiry and Practice*, Seiten 228–237, London, 2001. SAGE Publications.
- [Tel] *Telelogix DOORS website*. <http://www.telelogic.com>.
- [Tor01] TORBERT, W.R.: *The Practice of Action Inquiry*. In: REASON, P. und H. BRADBURY (Herausgeber): *Handbook of Action Research: Participative Inquiry and Practice*, Seiten 250–260, London, 2001. SAGE Publications.
- [Tou58] TOULMIN, S.: *The Uses of Argument*. Cambridge University Press, 1958.

7 Anhang

7.1 Iterationsanforderungen und Presseberichte

7.1.1 Anforderungen und Signale für das Spurassistenten-System (Iteration 1)

Sehr geehrte Damen und Herren,

der Wunsch nach neuen Technologien und Fahrkomfort erfordert neue Innovationen im Automobilbereich. Unsere Forschungsabteilung hat sich Gedanken über ein neuartiges System für die Führung eines PKWs gemacht. Nachdem der Tempomat die Geschwindigkeit eines Fahrzeugs regelt, sollten wir uns nun mit der Lenkung beschäftigen. Angedacht ist ein System, das für die Spureinhaltung des Fahrzeugs (z.B. auf langen Autobahnfahrten) zuständig ist. Dabei soll das System den Fahrer darauf aufmerksam machen, wenn er seine Fahrspur verlässt, ohne Anzeichen eines gewollten Spurwechsels zu zeigen. Somit kommen wir zu den elementaren Bestandteilen des Systems:

1. Das System soll dem Fahrer ein Feedback geben, wenn er seine Spur ohne Blinken verlässt
2. Diese Charakteristik soll bei Wunsch des Fahrers deaktiviert werden können.

Zu Feature 1: Für die Entwickler besteht die Frage, wann ein Spurwechsel gewollt ist und wann nicht. Dabei kann man sich Gedanken über die Lenkerstellung machen, sowie die Einstellungen des Blinkers. Es wäre also von Vorteil, wenn das Auslösen des Alarms nicht willkürlich ist, sondern nur durch einen begründeten Verdacht ausgelöst wird. Außerdem soll dem Fahrer auch unmittelbar Feedback gegeben werden, ob er gerade über die linke oder die rechte Fahrbahnbegrenzung hinaus gefahren ist.

Zu Feature 2: Im Falle einer Unzufriedenheit mit dem Verhalten des Spurassistenten, soll dieser abschaltbar sein. Das soll durch einen Schalter an der Armatur des Fahrers geschehen können. Wenn der Schalter der Armatur auf „aus“ gestellt ist, dann sind alle Signale der Sensoren zu ignorieren und das Fahrzeug gibt keine Warnhinweise an den Fahrer.

Folgende Signale werden von Sensoren geliefert:

- **SASSIS_SENSOR_LEFT_E**
An der Unterseite des Fahrzeugs befinden sich Sensoren für die Detektierung von Spurtrennungen. Falls die Markierungslinie überfahren wird, gibt der Sensor ein Signal auf den CAN-Bus. Der Sensor befindet sich auf der Fahrerseite.
- **SASSIS_SENSOR_RIGHT_E**
An der Unterseite des Fahrzeugs befinden sich Sensoren für die Detektierung von Spurtrennungen. Falls eine Spur gewechselt wird und eine Markierungslinie überfahren wird, gibt der Sensor ein Signal auf den CAN-Bus. Der Sensor befindet sich auf der Beifahrerseite.
- **SASSIS_ACTIVE_C**
Wird von einem Schalter an der Armatur gesteuert und bestimmt, ob der Spurassistent aktiv eingreifen soll oder nicht.

Mit Hilfe der folgenden Signale können die entsprechenden Aktoren angesprochen werden:

- **SASSIS_ALARM_LEFT_C**
Um dem Fahrer zu signalisieren, dass er gerade über die linke Fahrbahnbegrenzung gefahren ist, kann ein Alarm eingeschaltet werden, der sich links vom Fahrer befindet.
- **SASSIS_ALARM_RIGHT_C**
Um dem Fahrer zu signalisieren, dass er gerade über die rechte Fahrbahnbegrenzung gefahren ist, kann ein Alarm eingeschaltet werden, der sich auf der Beifahrerseite befindet.

Folgende Signale sind generell verfügbar:

- **CAR_START_C**
Zeigt an, ob die Zündung an (=TRUE) oder aus (=FALSE) ist.
- **CAR_GAS_D**
Gibt die Stellung des Gaspedals in Prozent an (0% = kein Gas, 100% = Vollgas)
- **CAR_SPEED_D**
aktuelle Geschwindigkeit des Fahrzeugs zwischen 0 und 250 km/h
- **CAR_STEERING_WHEEL_POS_D**
Lenkradstellungswinkel in Grad (-150° (links) bis +150° (rechts))
- **CAR_BLINKER_LEFT_C**
Zeigt an, dass der linke Blinker an ist
- **CAR_BLINKER_RIGHT_C**
Zeigt an, dass der rechte Blinker an ist

7.1.2 Anforderungen und Signale für das Spurassistenten-System (Iteration 2)

Sehr geehrte Damen und Herren,

nach der erfolgreichen Einführung des Spurassistenten haben wir beschlossen, die Funktionalität dieser Innovation zu erweitern. Hierzu haben sich die schlauesten Köpfe unseres Teams zusammengesetzt und ein neues Feature entwickelt, welches in der neuen Version des Spurassistenten realisiert werden soll. Der Ausgangspunkt dafür ist das Lenkrad. Da die Servolenkung des jetzigen Modells bei unterschiedlichen Geschwindigkeiten gleich bleibend ist, sehen wir Gefahren in einem versehentlichen, abrupten Lenkmanöver des Fahrers. Um Unfällen in einer solchen Situation vorzubeugen, wollen wir nun die Einstellung der Lenkungshärte an die Geschwindigkeit koppeln. Damit ist gemeint, dass sich das Lenkrad mit höherer Geschwindigkeit auch schwerer lenken lassen soll. Sinn macht dieses Feature z.B. auf der Autobahn. Dort wird normalerweise eine konstante Geschwindigkeit in eine konstante Richtung gefahren. ruckartige Lenkbewegungen sollen dort ausgeschlossen werden. Wie die vorherigen Features, soll auch dieses Feature mit der Deaktivierung des gesamten Spurassistenten ausgeschaltet werden können. Um diesen Schritt realisieren zu können, haben unsere Spezialisten die Lenkung so umgebaut, dass sie nun von der Software gesteuert werden kann. Zu diesem Zweck haben wir das Signal `CAR_STEERING_WHEEL_POS_D` auf dem CAN-Bus jetzt abgreifbar gemacht. Somit ist es möglich, den Einschlagswinkel des Lenkrads mit diesem Signal zu verändern.

Ihre Aufgabe ist es nun, die alte Steuerung an die neuen Voraussetzungen anzupassen!

Folgende Signale werden von Sensoren geliefert:

- `SASSIS_SENSOR_LEFT_E`
An der Unterseite des Fahrzeugs befinden sich Sensoren für die Detektierung von Spurtrennungen. Falls die Markierungslinie überfahren wird, gibt der Sensor ein Signal auf den CAN-Bus. Der Sensor befindet sich auf der Fahrerseite.
- `SASSIS_SENSOR_RIGHT_E`
An der Unterseite des Fahrzeugs befinden sich Sensoren für die Detektierung von Spurtrennungen. Falls eine Spur gewechselt wird und eine Markierungslinie überfahren wird, gibt der Sensor ein Signal auf den CAN-Bus. Der Sensor befindet sich auf der Beifahrerseite.
- `SASSIS_ACTIVE_C`
Wird von einem Schalter an der Armatur gesteuert und bestimmt, ob der Spurassistent aktiv eingreifen soll oder nicht.

Mit Hilfe der folgenden Signale können die entsprechenden Aktoren angesprochen werden:

- `SASSIS_ALARM_LEFT_C`
Um dem Fahrer zu signalisieren, dass er gerade über die linke Fahrbahnbegrenzung gefahren ist, kann ein Alarm eingeschaltet werden, der sich links vom Fahrer befindet.
- `SASSIS_ALARM_RIGHT_C`
Um dem Fahrer zu signalisieren, dass er gerade über die rechte Fahrbahnbegrenzung gefahren ist, kann ein Alarm eingeschaltet werden, der sich auf der Beifahrerseite befindet.

- **SASSIS_WHEEL_DRAG_D**

Damit kann der Widerstand der Lenkung eingestellt werden (0% = kein Widerstand, 100% = max. Widerstand)

Folgende Signale sind generell verfügbar:

- **CAR_START_C**

Zeigt an, ob die Zündung an (=TRUE) oder aus (=FALSE) ist.

- **CAR_GAS_D**

Gibt die Stellung des Gaspedals in Prozent an (0% = kein Gas, 100% = Vollgas)

- **CAR_SPEED_D**

aktuelle Geschwindigkeit des Fahrzeugs zwischen 0 und 250 km/h

- **CAR_STEERING_WHEEL_POS_D**

Lenkradstellungswinkel in Grad (-150° (links) bis +150° (rechts))

- **CAR_BLINKER_LEFT_C**

Zeigt an, dass der linke Blinker an ist

- **CAR_BLINKER_RIGHT_C**

Zeigt an, dass der rechte Blinker an ist

Nachtrag: Die folgenden Berichte wurden uns von der Stelle für Öffentlichkeitsarbeit angetragen. Bitte lesen sie diese durch und erweitern/verändern sie nach Möglichkeit das Modell soweit, dass die Mängel bzw. die Gefahrenquellen für das bestehende System nicht mehr in Betracht kommen.

Führung mit Schwierigkeiten

Der neue PKW-Spurassistent scheint bei seiner Einführung ein paar gravierende Mängel aufzuweisen.

Düsseldorf, 29.02.2005: Nach der Einführung des neuen PKW-Spurassistenten kommt es nun zu Problemen. Bei der Konstruktion der Sensoren wurde wohl nicht auf die Möglichkeit der Verschmutzung dieser geachtet. Zumindest vernachlässigte man das Thema, was dazu führte, dass ein Fahrer die Kontrolle über sein Fahrzeug verlor und damit einen Unfall provozierte. Hans U. aus Mainz war auf dem Weg nach Hause von einer geschäftlichen Besprechung. Die Besprechung ging bis in den späten Abend und Hans U. war müde von den langen Diskussionen. Als er in einen Sekundenschlaf verfiel, kam er von der Spur ab und geriet gegen die Leitplanken der Autobahn. Der Wagen stellte sich quer und Martha M., die direkt hinter Hans U. herfuhr, konnte nicht mehr ausweichen und fuhr fast ungebremst in den Rexo. Das neue System des Rexos, das den Fahrer unterstützen wollte, hat laut Angaben nicht funktioniert, da die Sensoren verschmutzt waren und darum nicht die Straßenlinien erkennen konnten, die die Fahrbahn abtrennen.

7.1.3 Anforderungen und Signale für das Spurassistenten-System (Iteration 3)

Sehr geehrte Damen und Herren,

die neue Aufgabe, die wir Ihnen stellen, soll die Sicherheit im Straßenverkehr erhöhen. Dabei soll der Fahrer, wenn er einen Notfall erleidet oder in einen Sekundenschlaf gerät, sicher von der Fahrbahn geleitet werden. Unsere Vorstellung des Vorgehens ist, dass der Fahrer nach einer bestimmten Zeit, in der die Warnungen des Fahrzeugs an sind und ignoriert werden, von der Aufgabe entbunden wird, das Fahrzeug selbst zu steuern. In diesem Fall soll das Fahrzeug an den rechten Straßenrand abgestellt werden. Die Warnblinker sollen dabei als Zeichen eines Notfalls aktiviert werden. Vorstellbar ist es auch, dass nicht nach Zeit, sondern nach Aktionen die Lage des Fahrers beurteilt wird. So könnte das mehrmalige Eingreifen in die Lenkung innerhalb bestimmter Intervalle darauf hindeuten, dass der Fahrer nicht mehr in der Lage ist, sein Fahrzeug selbst zu steuern. Ein Beispiel wäre dafür ein Fahrer, der auf seinem Lenkrad eingeschlafen ist und dadurch „Schlangenlinien“ fährt. Ihre Aufgabe ist es nun, dieses Feature in die momentane Steuerung einzubauen.

Folgende Signale werden von Sensoren geliefert:

- **SASSIS_SENSOR_LEFT_E**
An der Unterseite des Fahrzeugs befinden sich Sensoren für die Detektierung von Spurtrennungen. Falls die Markierungslinie überfahren wird, gibt der Sensor ein Signal auf den CAN-Bus. Der Sensor befindet sich auf der Fahrerseite.
- **SASSIS_SENSOR_RIGHT_E**
An der Unterseite des Fahrzeugs befinden sich Sensoren für die Detektierung von Spurtrennungen. Falls eine Spur gewechselt wird und eine Markierungslinie überfahren wird, gibt der Sensor ein Signal auf den CAN-Bus. Der Sensor befindet sich auf der Beifahrerseite.
- **SASSIS_ACTIVE_C**
Wird von einem Schalter an der Armatur gesteuert und bestimmt, ob der Spurassistent aktiv eingreifen soll oder nicht.
- **SASSIS_ERROR_C**
Wenn der Sensor sinnlose Werte liefert (weil er verschmutzt oder einfach kaputt ist), wird diese Bedingung gesetzt.

Mit Hilfe der folgenden Signale können die entsprechenden Aktoren angesprochen werden:

- **SASSIS_ALARM_LEFT_C**
Um dem Fahrer zu signalisieren, dass er gerade über die linke Fahrbahnbegrenzung gefahren ist, kann ein Alarm eingeschaltet werden, der sich links vom Fahrer befindet.
- **SASSIS_ALARM_RIGHT_C**
Um dem Fahrer zu signalisieren, dass er gerade über die rechte Fahrbahnbegrenzung gefahren ist, kann ein Alarm eingeschaltet werden, der sich auf der Beifahrerseite befindet.

- **SASSIS_WHEEL_DRAG_D** Damit kann der Widerstand der Lenkung eingestellt werden (0% = kein Widerstand, 100% = max. Widerstand)
- **SASSIS_WARNING_C**
Mit dieser Bedingung kann ein Warnhinweis in der Bedienkonsole ausgegeben werden. Diese Warnung weist auf eine Fehlfunktion des Spurassistenten-Systems hin.
- **SASSIS_SENSOR_HEATER_C**
Erwärmt Heizspulen um die Sensoren.

Folgende Signale sind generell verfügbar:

- **CAR_START_C**
Zeigt an, ob die Zündung an (=TRUE) oder aus (=FALSE) ist.
- **CAR_GAS_D**
Gibt die Stellung des Gaspedals in Prozent an (0% = kein Gas, 100% = Vollgas)
- **CAR_SPEED_D**
aktuelle Geschwindigkeit des Fahrzeugs zwischen 0 und 250 km/h
- **CAR_STEERING_WHEEL_POS_D**
Lenkradstellungswinkel in Grad (-150° (links) bis +150° (rechts)) JETZT steuerbar !
- **CAR_BLINKER_LEFT_C**
Zeigt an, dass der linke Blinker an ist
- **CAR_BLINKER_RIGHT_C**
Zeigt an, dass der rechte Blinker an ist
- **CAR_TEMPERATURE_SENSOR_D**
Zeigt die momentane Außentemperatur des Fahrzeugs an.
- **CAR_CRASH_E**
Falls ein Unfall auftritt.
- **HW_BLINK_LEFT_C**
schaltet die Blinklichter auf der linken Fahrzeug Seite an (=TRUE) bzw. aus (=FALSE)
- **HW_BLINK_RIGHT_C**
schaltet die Blinklichter auf der rechten Fahrzeug Seite an (=TRUE) bzw. aus (=FALSE)

Nachtrag: Die folgenden Berichte wurden uns von der Stelle für Öffentlichkeitsarbeit angetragen. Bitte lesen sie diese durch und erweitern/verändern sie nach Möglichkeit das Modell soweit, dass die Mängel bzw. die Gefahrenquellen für das bestehende System nicht mehr in Betracht kommen.

Auto, Motor, Sport

Die Zukunftsaussichten des PKW-Spurassistenten auf dem Prüfstand

Wo führt uns die Technik noch hin? Das ist eine Frage, die sich die Automobilindustrie schon lange stellt. Durch die schnell wachsenden Möglichkeiten der Technik, kommt man auf immer ausgefeiltere Ideen, die in den neusten Automobilen zum Einsatz kommen. So ist auch der PKW-Spurassistent wieder eine Neuentwicklung mit zukunftssträchtigen Veranlagungen. So wie er jetzt schon zum Einsatz kommt, ist er in der Lage, den Fahrer aktiv dabei zu unterstützen, sein Wagen sicher durch den Straßenverkehr zu lenken. Nichts desto trotz gibt es immer noch kleine Probleme, mit denen der Hersteller zu kämpfen hat. Wie sieht es aber in Zukunft aus. Was wird aus dem Spurassistenten? Wird er wie viele andere Technologien vor ihm wieder in der Versenkung verschwinden, oder hat er das Zeug zu einem zukunftssträchtigen Teil des Autos zu werden? Die Entwickler des PKW-Spurassistenten legen alles daran, dass diese Technologie weiterhin die Straße sicherer machen kann. So sind die Entwickler dabei, den Spurassistenten aktiv an der Fahrt teilnehmen zu lassen und kombinieren diesen mit anderen Board-Systemen des Autos, wie beispielsweise die Geschwindigkeitsmessung oder das Lenk-System. So soll der Assistent in der Lage sein aktiv in die Lenkung einzugreifen und den Fahrer zu schützen, sollte dieser in eine Notlage geraten sein, in der er nicht mehr im Stande ist sein Fahrzeug zu lenken (Herzinfarkt oder ähnliches).

Streit mit der Technik

Meinungskonflikt zwischen der Technik eines Autos und seinem Besitzer

Armin B., der auf dem Weg nach Hause über die Bundesstraße B10 Richtung Senden fuhr, kam in eine gefährliche Situation, in die ihn die eigentlich unterstützende Hilfstechneik seines Autos brachte. Der erst vor kurzem erstandene Neuwagen der Marke WWW besitzt die Technologie des PKW-Spurassistenten. Als Armin B. nun gerade einen Laster überholen wollte, aktivierte sich der Spurassistent und verhinderte den Spurwechsel. Daraufhin musste Armin B. eine Vollbremsung hinlegen, die um Haaresbreite zu einer Kollision mit dem Laster führte. Der PKW-Spurassistent wurde auf Wunsch von Armin B. wieder aus dem Wagen entfernt, was die Herstellerfirma des Wagens auf Kulanz veranlasste.

Winterzeit mit Folgen

Der PKW-Spurassistent von Rexo erweist sich wieder als fehlerhaft.

Die Fehlersträhne von Rexo reißt nicht ab. Nach der Panne mit den verschmutzten Sensoren kommt ein neues Problem auf den Autohersteller zu. Diesmal macht ihm die Winterzeit zu schaffen. Ein Modell der Baureihe MR2 hat während einer Fahrt ständig ein Signal der Spurassistenten-Sensoren empfangen. Der Fahrer des Wagens hat auf eine Vermutung hin den Wagen durch eine Waschstraße gefahren und nach der Reinigung war der Defekt behoben. Es sieht stark danach aus, als hätte Rexo wieder einmal ein Eigentor geschossen, als sie nach der letzten Panne mit dem PKW-Spurassistenten behaupteten, dass das System nun sicher vor solchen Zwischenfällen sei. Rexo hat sich zu diesem Vorfall noch nicht geäußert oder eine Stellungnahme dazu gemacht. Eins ist aber sicher: es werden bald einige Autos zurück in die Werkstätten gefahren, um die Sensoren nachzurüsten.

7.1.4 Anforderungen und Signale für das Airbag-System (Iteration 1)

Sehr geehrte Damen und Herren,

nach einschlägiger Beratung in unserem Beratungsteam haben wir beschlossen, folgende Funktionen in der Innovation Airbag zu realisieren:

1. Der Airbag soll bei einem Unfall ausgelöst werden.
2. Das Feature Airbag ist auf Wunsch des Fahrers abschaltbar.

Nähere Erläuterung:

Da der Wissensstand über Unfälle angewachsen ist und der Wunsch nach einem Abfang-System immer größer wird, haben wir uns entschlossen eine Innovation entwickeln zu lassen, die eben diese realisieren soll. Nach reichlichem Diskutieren sind wir zu dem Entschluss gekommen, die Innovation Airbag entwickeln zu lassen. Der Airbag ist eine Konstruktion eines mit Luft gefüllten „Kissens“, das in der Lage ist, sich in einem Bruchteil einer Sekunde aufzublasen. Durch das kontrollierte entweichen lassen von Luft bei dem Auftreffen eines Kopfes auf das Luftkissen, kann die Verletzungsgefahr drastisch gesenkt werden.

Zu Feature 1:

Im Falle eines Unfalls sollen die Insassen des Fahrzeugs geschützt werden. Dies soll mit Hilfe des Airbags unterstützt werden. Wenn der Zündschlüssel des Fahrzeugs gedreht worden ist und die Zündung aktiv ist, soll der Airbag ebenfalls aktiv sein. Sollte in diesem Zustand ein gegnerisches Fahrzeug mit dem eigenen Fahrzeug kollidieren, sollen sich die Airbags automatisch aktivieren.

Zu Feature 2:

Auf Wunsch des Fahrers kann die Funktion des Airbags abgeschaltet werden. Das soll dem Fahrer die Möglichkeit lassen, bei Bedenken und mangelndem Vertrauen in diese Technologie, diese abzuschalten. Im Falle eines Unfalls, wird bei einem deaktivierten Airbag keinerlei Funktion ausgeführt.

Folgende Signale werden von Sensoren geliefert:

- **AIRBAG_DRIVER_SENSOR_E**
Bei einem harten, frontalen Schlag (Auftreffen eines anderen Fahrzeugs) löst dieser Sensor aus. Der Sensor ist in der Stoßstange auf der Fahrerseite des Fahrzeugs positioniert.
- **AIRBAG_CODRIVER_SENSOR_E**
Bei einem harten frontalen Schlag (Auftreffen eines anderen Fahrzeugs) löst dieser Sensor aus. Der Sensor ist in der Stoßstange auf der Beifahrerseite des Fahrzeugs positioniert.
- **AIRBAG_DRIVER_SIDE_SENSOR_E**
Für das seitliche Auftreffen des Fahrzeugs wurde auch ein Sensor in der Seitenverkleidung des Wagens montiert. Bei einem seitlichen Unfall löst der Sensor einen Impuls aus.

- **AIRBAG_CODRIVER_SIDE_SENSOR_E**
Für das seitliche Auftreffen des Fahrzeugs wurde auch ein Sensor in der Seitenverkleidung des Wagens montiert. Bei einem seitlichen Unfall löst der Sensor einen Impuls aus.
- **AIRBAG_ACTIVE_C**
Das Feature Airbag soll auch abschaltbar sein. Diese Bedingung repräsentiert die Schalterstellung in der Mittelkonsole. Standardmäßig ist diese Bedingung wahr.

Mit Hilfe der folgenden Signale können die entsprechenden Aktoren angesprochen werden:

- **AIRBAG_DRIVER_E**
Durch dieses Signal wird der Airbag auf der Fahrerseite angewiesen auszulösen.
- **AIRBAG_CODRIVER_E**
Durch dieses Signal wird der Airbag auf der Beifahrerseite angewiesen auszulösen.
- **AIRBAG_DRIVER_SIDE_E**
Durch dieses Signal wird der Seitenairbag auf der Fahrerseite angewiesen auszulösen.
- **AIRBAG_CODRIVER_SIDE_E**
Durch dieses Signal wird der Seitenairbag auf der Beifahrerseite angewiesen auszulösen.

Da der Auslöseprozess chemisch funktioniert, ist ein weiteres Auslösen erst nach der Reparatur des Airbag-Systems möglich!

Folgende Signale sind generell verfügbar:

- **CAR_START_C**
Zeigt an, ob die Zündung an (=TRUE) oder aus (=FALSE) ist.
- **CAR_GAS_D**
Gibt die Stellung des Gaspedals in Prozent an (0% = kein Gas, 100% = Vollgas)
- **CAR_SPEED_D**
aktuelle Geschwindigkeit des Fahrzeugs zwischen 0 und 250 km/h
- **CAR_STEERING_WHEEL_POS_D**
Lenkradstellungswinkel in Grad (-150° (links) bis +150° (rechts))
- **CAR_BLINKER_LEFT_C**
Zeigt an, dass der linke Blinker an ist
- **CAR_BLINKER_RIGHT_C**
Zeigt an, dass der rechte Blinker an ist

7.1.5 Anforderungen und Signale für das Airbag-System (Iteration 2)

Sehr geehrte Damen und Herren,

die Effizienz des Airbags wurde in verschiedenen Crashesituationen getestet. Dabei kamen die Wissenschaftler zu dem Schluss, dass er zwar hilft, Verletzung in Unfallsituationen zu lindern. Aber durch das schlagartige Aufblasen mit voller Entfaltungskraft des Airbags kam es gerade bei kleinen Unfällen zu zusätzlichen Verletzungen der Insassen. Dies könnte vermieden werden, wenn der Airbagdruck geschwindigkeitsabhängig geregelt werden würde. Dazu wurde in der Chemieabteilung bereits eine neue Patrone entwickelt, die es ermöglicht, den Druck stufenlos von 0 bis 100% der maximalen Leistung zu kontrollieren. Dafür ist es notwendig, vor dem Auslösen über den CAN-Bus das Signal AIRBAG_PRESSURE_D mit dem korrekten Prozentsatz zu senden. Die Messungen haben ergeben, dass ab einer Geschwindigkeit von 100 km/h der maximale Druck notwendig ist und dass aufgrund der quadratisch eingehenden negativen Beschleunigung, der Druck ebenfalls nicht linear sondern quadratisch steigt.

Passen Sie das vorhandene Steuergerät an diese neuen Anforderungen an!

Zur Erinnerung:

Folgende Signale werden von Sensoren geliefert:

- AIRBAG_DRIVER_SENSOR_E
Bei einem harten, frontalen Schlag (Auftreffen eines anderen Fahrzeugs) löst dieser Sensor aus. Der Sensor ist in der Stoßstange auf der Fahrerseite des Fahrzeugs positioniert.
- AIRBAG_CODRIVER_SENSOR_E
Bei einem harten frontalen Schlag (Auftreffen eines anderen Fahrzeugs) löst dieser Sensor aus. Der Sensor ist in der Stoßstange auf der Beifahrerseite des Fahrzeugs positioniert.
- AIRBAG_DRIVER_SIDE_SENSOR_E
Für das seitliche Auftreffen des Fahrzeugs wurde auch ein Sensor in der Seitenverkleidung des Wagens montiert. Bei einem seitlichen Unfall löst der Sensor einen Impuls aus.
- AIRBAG_CODRIVER_SIDE_SENSOR_E
Für das seitliche Auftreffen des Fahrzeugs wurde auch ein Sensor in der Seitenverkleidung des Wagens montiert. Bei einem seitlichen Unfall löst der Sensor einen Impuls aus.
- AIRBAG_ACTIVE_C
Das Feature Airbag soll auch abschaltbar sein. Diese Bedingung repräsentiert die Schalterstellung in der Mittelkonsole. Standardmäßig ist diese Bedingung wahr.

Mit Hilfe der folgenden Signale können die entsprechenden Aktoren angesprochen werden:

- AIRBAG_DRIVER_E
Durch dieses Signal wird der Airbag auf der Fahrerseite angewiesen auszulösen.

- **AIRBAG_CODRIVER_E**
Durch dieses Signal wird der Airbag auf der Beifahrerseite angewiesen auszulösen.
- **AIRBAG_DRIVER_SIDE_E**
Durch dieses Signal wird der Seitenairbag auf der Fahrerseite angewiesen auszulösen.
- **AIRBAG_CODRIVER_SIDE_E**
Durch dieses Signal wird der Seitenairbag auf der Beifahrerseite angewiesen auszulösen.
- **AIRBAG_PRESSURE_D**
Dieses Signal gibt den Druck für den Airbag vor (0% = kein Druck, 100% = max. Druck)

Da der Auslöseprozess chemisch funktioniert, ist ein weiteres Auslösen erst nach der Reparatur des Airbag-Systems möglich!

Folgende Signale sind generell verfügbar:

- **CAR_START_C**
Zeigt an, ob die Zündung an (=TRUE) oder aus (=FALSE) ist.
- **CAR_GAS_D**
Gibt die Stellung des Gaspedals in Prozent an (0% = kein Gas, 100% = Vollgas)
- **CAR_SPEED_D**
aktuelle Geschwindigkeit des Fahrzeugs zwischen 0 und 250 km/h
- **CAR_STEERING_WHEEL_POS_D**
Lenkradstellungswinkel in Grad (-150° (links) bis +150° (rechts))
- **CAR_BLINKER_LEFT_C**
Zeigt an, dass der linke Blinker an ist
- **CAR_BLINKER_RIGHT_C**
Zeigt an, dass der rechte Blinker an ist

Nachtrag: Die folgenden Berichte wurden uns von der Stelle für Öffentlichkeitsarbeit angetragen. Bitte lesen sie diese durch und erweitern/verändern sie nach Möglichkeit das Modell soweit, dass die Mängel bzw. die Gefahrenquellen für das bestehende System nicht mehr in Betracht kommen.

Airbagvandalen

In Berlin machten sich Jungendlich mit Baseballschlägern an Autos zu schaffen, um die Airbags der Karossen auszulösen.

Berlin Mitte, Di. 23.11.2004: Autos mit Airbags in Berlin Mitte waren in den letzten Tagen nicht sicher. Jugendliche, die das für einen Spaß hielten, zogen mit Baseballschlägern ausgestattet durch die Gegend und lösten mit einem gezielten Schlag auf die Stoßstange des Fahrzeugs die Airbags aus. Der Schaden, der dabei entstand geht in die Hunderttausende. Die Polizei ist auf der Suche nach den Tätern und einige Hinweise verdichten sich schon, wo die Täter zu suchen sind. Für sachdienliche Hinweise ist die zuständige Polizei sehr dankbar

und auf die Ergreifung der Schuldigen ist eine Belohnung von 10.000 Euro angesetzt. Die Tat ereignete sich vom Montagabend auf den darauf folgenden Dienstag im Raum Berlin Mitte. Betroffen waren hauptsächlich Fahrzeuge neueren Datums, die mit der Sicherheitstechnologie Airbag ausgestattet waren. Laut Angaben, war es eine Gruppe von vier bis fünf Jugendliche im Alter von 14-17 Jahren.

Projekt: Intelligentes Luftkissen

Vorhaben: ein an die Geschwindigkeit gekoppelter Airbag

Beschreibung: Der bisherige Stand der Airbagtechnologie lässt es momentan nicht zu, dass ein Airbag in verschiedenen Härtegraden aufblasbar ist. Nach einer Auslösung des Airbags, geht dieser in einer festgelegten Zeit auf und entfaltet sich komplett. Es gibt dabei nur zwei Zustände die ein Airbag hat: er ist zusammengefaltet oder er ist ausgelöst und voll aufgeblasen. Nun soll es anders werden. Nach einer eingehenden Beratung wollen wir nun, dass das Airbag-Kissen dynamisch nach der Geschwindigkeit aufgeblasen wird. Dabei soll eine Härteregulierung von 0-100% möglich sein. Nach ausreichenden Tests wurde festgestellt, dass der Airbag erst ab einer Geschwindigkeit von 90 km/h voll aufgepumpt sein muss, damit er nicht zusätzliche Verletzungen auslösen kann.

7.1.6 Anforderungen und Signale für das Airbag-System (Iteration 3)

Sehr geehrte Damen und Herren,

nach weiteren Untersuchungen an dem bisherigen System, haben wir festgestellt, dass es immer noch eine Weiterentwicklung benötigt. Unsere Spezialisten haben uns darauf aufmerksam gemacht, dass der Airbag in Bezug auf das Auslösen eine Verbesserung erfahren könnte. Dabei wollen wir diesmal Wert darauf legen, den Airbag „intelligent“ zu machen. Dabei denken wir nicht an das Erstellen einer KI, die Augen und Ohren für den Fahrer ersetzen soll, aber etwas in dieser Art schwebte uns schon vor. Ihre Aufgabe soll es diesmal sein, den Airbag an einen Annäherungssensor anzubinden, der ständig den Wert des Abstands zum vorderen Fahrzeug auf den CAN-BUS legt.

Genauere Beschreibung der Idee:

Wenn die Zündung des Fahrzeugs aktiv und der Airbag angeschaltet ist, soll das neue System aktiv werden. Dabei befindet sich ein neuer Sensor an der vorderen Stoßstange des Fahrzeugs. Dieser Sensor reagiert auf die Entfernung großer Objekte (z.B. Autos oder Bäume) die sich vor dem Fahrzeug befinden. Sollte sich nun die Entfernung zu diesem Gegenstand zu schnell verringern, ist wohl davon auszugehen, dass ein Zusammenstoß nicht mehr zu vermeiden ist und daher sollen in dieser Phase evtl. Vorbereitungen für den Zusammenprall geschehen. Neben einigen anderen Aktionen (Fenster schließen,...) sollte vor allem der Airbag nicht erst nach dem Zusammenstoß, sondern schon wenige Millisekunden vorher aktiviert werden, damit er beim tatsächlichen Aufprall schon voll aufgeblasen ist. Um diese Funktion zu realisieren, müssen Sie also den Abstandssensor wiederholt abfragen und dabei die Änderung pro Zeit berechnen. Bei einer zu schnellen Annäherung soll der Airbag dann beim Abstand 0.0 ausgelöst werden.

Zur Erinnerung:

Folgende Signale werden von Sensoren geliefert:

- **AIRBAG_DRIVER_SENSOR_E**
Bei einem harten, frontalen Schlag (Auftreffen eines anderen Fahrzeugs) löst dieser Sensor aus. Der Sensor ist in der Stoßstange auf der Fahrerseite des Fahrzeugs positioniert.
- **AIRBAG_CODRIVER_SENSOR_E**
Bei einem harten frontalen Schlag (Auftreffen eines anderen Fahrzeugs) löst dieser Sensor aus. Der Sensor ist in der Stoßstange auf der Beifahrerseite des Fahrzeugs positioniert.
- **AIRBAG_DRIVER_SIDE_SENSOR_E**
Für das seitliche Auftreffen des Fahrzeugs wurde auch ein Sensor in der Seitenverkleidung des Wagens montiert. Bei einem seitlichen Unfall löst der Sensor einen Impuls aus.
- **AIRBAG_CODRIVER_SIDE_SENSOR_E**
Für das seitliche Auftreffen des Fahrzeugs wurde auch ein Sensor in der Seitenverkleidung

dung des Wagens montiert. Bei einem seitlichen Unfall löst der Sensor einen Impuls aus.

- **AIRBAG_ACTIVE_C**
Das Feature Airbag soll auch abschaltbar sein. Diese Bedingung repräsentiert die Schalterstellung in der Mittelkonsole. Standardmäßig ist diese Bedingung wahr.
- **AIRBAG_CODRIVER_SEAT_SENSOR_C**
Befindet sich ein Objekt auf dem Beifahrersitz wird die Condition „true“
- **AIRBAG_CHILD_SEAT_SENSOR_C**
Meldet, wenn ein Kindersitz auf der Beifahrerseite installiert worden ist.
- **AIRBAG_TAILGATE_D**
Abstand zum vorderen Fahrzeug/Objekt

Mit Hilfe der folgenden Signale können die entsprechenden Aktoren angesprochen werden:

- **AIRBAG_DRIVER_E**
Durch dieses Signal wird der Airbag auf der Fahrerseite angewiesen auszulösen.
- **AIRBAG_CODRIVER_E**
Durch dieses Signal wird der Airbag auf der Beifahrerseite angewiesen auszulösen.
- **AIRBAG_DRIVER_SIDE_E**
Durch dieses Signal wird der Seitenairbag auf der Fahrerseite angewiesen auszulösen.
- **AIRBAG_CODRIVER_SIDE_E**
Durch dieses Signal wird der Seitenairbag auf der Beifahrerseite angewiesen auszulösen.
- **AIRBAG_PRESSURE_D**
Dieses Signal gibt den Druck für den Airbag vor (0% = kein Druck, 100% = max. Druck)
- **AIRBAG_SENSOR_HEATER_C**
Aktiviert Heizspulen für die Airbag-Sensoren.

Da der Auslöseprozess chemisch funktioniert, ist ein weiteres Auslösen erst nach der Reparatur des Airbag-Systems möglich!

Folgende Signale sind generell verfügbar:

- **CAR_START_C**
Zeigt an, ob die Zündung an (=TRUE) oder aus (=FALSE) ist.
- **CAR_GAS_D**
Gibt die Stellung des Gaspedals in Prozent an (0% = kein Gas, 100% = Vollgas)
- **CAR_SPEED_D**
aktuelle Geschwindigkeit des Fahrzeugs zwischen 0 und 250 km/h
- **CAR_STEERING_WHEEL_POS_D**
Lenkradstellungswinkel in Grad (-150° (links) bis +150° (rechts)) JETZT steuerbar !

- **CAR_BLINKER_LEFT_C**
Zeigt an, dass der linke Blinker an ist
- **CAR_BLINKER_RIGHT_C**
Zeigt an, dass der rechte Blinker an ist
- **CAR_TEMPERATURE_SENSOR_D**
Zeigt die momentane Außentemperatur des Fahrzeugs an.
- **CAR_CRASH_E**
Falls ein Unfall auftritt.
- **HW_BLINK_LEFT_C**
schaltet die Blinklichter auf der linken Fahrzeug Seite an (=TRUE) bzw. aus (=FALSE)
- **HW_BLINK_RIGHT_C**
schaltet die Blinklichter auf der rechten Fahrzeug Seite an (=TRUE) bzw. aus (=FALSE)

Nachtrag: Die folgenden Berichte wurden uns von der Stelle für Öffentlichkeitsarbeit ange-
tragen. Bitte lesen Sie diese durch und erweitern/verändern Sie nach Möglichkeit das Modell
soweit, dass die Mängel bzw. die Gefahrenquellen für das bestehende System nicht mehr in
Betracht kommen.

Todesgefahr Airbag

*Durch das Fehlauslösens eines Airbags kam es zu einem tragischen Unfall, bei dem
ein Kleinkind fast ums Leben kam.*

Stuttgart, Di. 13.10.2004: Technologie, die zum Schutz der Menschen entwickelt wurde
kann auch dazu beitragen, dass Menschen zu Schaden oder gar ums Leben kommen. So ge-
schah es vor einigen Tagen in Stuttgart im Parkhaus in der Weingartenstraße. Die junge
Mutter Miranda D. saß in ihrem Wagen und wollte gerade das Parkhaus verlassen, als ihr
Handy anfang zu klingeln. Miranda nahm den Anruf entgegen und im selben Augenblick, als
sie den Anruf annahm, schnellte der Airbag aus der Vorrichtung an den Armaturen des Autos
und presste den Säugling fest gegen den Beifahrersitz. Nach einiger Zeit löste sich der Druck
des Airbags wieder und die junge Mutter konnte ihr Kind befreien. Sie rief auf der Stelle
den Notarzt an und der Säugling wurde ins Krankenhaus gebracht. Außer ein paar kleinere
Prellungen und Quetschungen, hat das kleine Kind einen gebrochen Arm als Folge des aus-
gelösten Airbags zu ertragen.

Anruf mit Folgen

*Ein Anruf eines Freundes wurde Alexander R. auf der Autobahn Richtung Karlsruhe
zum Verhängnis.*

Ulm, Mi. 06.02.2005: Auf der Autobahn Richtung Karlsruhe ereignete sich um ca 16:20
Uhr ein tragischer Unfall, bei dem glücklicherweise niemand zu Schaden gekommen ist. Alex-
ander R. war gerade auf dem Weg zu seiner Geliebten, als er überraschend von seiner Frau
angerufen wurde. In seinem brandneuen Portus war eine Telefonhalterung in den Armaturen
des Wagens angebracht. Als die Verbindung des Telefons aufgebaut worden ist, hat es wohl
eine Überlagerung des Airbagsensors gegeben und löste diesen aus. Alexander R. wurde durch

den Airbag überrascht und verlor die Kontrolle über sein Fahrzeug. Er kam von der Straße ab und überschlug sich einige Male. Portus äußerte sich bis zu diesem Zeitpunkt noch nicht zu diesem Vorfall. Man geht allerdings davon aus, dass die Schuld an diesem Unglück bei den Herstellern der Sensoren liegt und auch aus dieser Richtung eine Rechtfertigung zu erwarten ist. Der Schaden der an dem Fahrzeug und der Gesundheit von Alexander R. zu beklagen ist, ist noch nicht geklärt, da der Hersteller der Sensoren und Portus sich noch nicht einig konnten.

Sensordefekt auf Grund von Vereisung

Wegen vereister Sensoren kam es auf der Autobahn von Karlsruhe nach Stuttgart zu einem Unfall, an dem zehn Autos beteiligt waren.

Vaihingen, So. 03.01.2005: Durch den Kälteeinbruch am 24.12. und den erhöhten Niederschlag kam es vielerorts zu Unfällen. Einer der größten ereignete sich in der Nähe von Vaihingen, wenige Kilometer vor Stuttgart. Ein junger Mann, Matthias S., verließ sich zu sehr auf die Sicherheitseinrichtungen seines neuen Goliath. Er fuhr einen überholenden LKW zu dicht auf und als dieser aufgrund eines Stauendes stark bremsen musste, schoss Matthias M. mit seinem Fahrzeug mit voller Wucht in das Ende des LKW. Normalerweise entspricht dies einem typischen Crashtest, wie er mit allen neuen Autos hundertfach getestet wird. Leider lösten in diesem Fall, wohl aufgrund der extrem kalten Witterung, die entsprechenden Sensoren nicht aus und so wurde der Fahrer schwer verletzt in eine Unfallspezialklinik geflogen, wo er sich bis jetzt in einem sehr kritischen Zustand befindet. Die Automobilhersteller weisen in diesem Zusammenhang noch mal darauf hin, dass passive Sicherheitskomponenten jederzeit ausfallen können, man sich daher nicht darauf verlassen sollte und seinen Fahrstil entsprechend anpassen muss.

Hoher Sachschaden bei kleinem Auffahrunfall

Beim einem leichten Auffahrunfall entstand aufgrund fehlerhafter Technik ein Schaden in Höhe von 15.000,-.

Augsburg, Do. 05.05.2005: Herr Thomas K. hatte es wieder einmal zu eilig. Bei einer Ampel, die gerade auf grün geschaltet war, fuhr er zu schnell - zumindest schneller als sein Vordermann - los und berührte dabei leicht dessen Stoßstange. Durch diese leichte Erschütterung wurden beide Airbags ausgelöst. Der Beifahrerairbag, der sinnlos aufgeblasen wurde, da sich Thomas K. alleine unterwegs befand, schlug die Handyhalterung weg und zerstörte die Windschutzscheibe. Die Aufwände für die Reparatur der Windschutzscheibe, ein neues Mobiltelefon und die Instandsetzung der Airbags und Gurtstraffer beläuft sich auf ca. EUR 15.000,-.

7.1.7 Anforderungen und Signale für das Verriegelungs-System (Iteration 1)

Sehr geehrte Damen und Herren,

immer wieder kommt es vor, dass Mitfahrer vergessen, die Türen beim Verlassen des Autos zu verriegeln. Dieser Umstand macht es entweder für Einbrecher sehr leicht oder aber der Fahrer sieht sich gezwungen, nach jedem Abschließen der Fahrertür, alle Türen noch mal zu kontrollieren. Für die Bequemlichkeit des Fahrers soll eine Innovation entwickelt werden, die es ermöglicht, sämtliche Türen eines Fahrzeugs auf einmal zu entriegeln und zu verriegeln. Die wesentlichen Kernpunkte, die das System beherrschen sollte, sind im Folgenden aufgelistet:

1. Das Entriegeln und Verriegeln aller Türen (inkl. Kofferraum) mit einer Schlüsselumdrehung soll möglich sein.
2. Der Kofferraum soll zusätzlich aber auch unabhängig von der Zentralverriegelung gesteuert werden können.
3. Komfortfunktionen, wie z.B. eine Vermeidung des „Sich-Ausschließens“ sollen berücksichtigt werden.

Zu Feature 1:

Mit einer Schlüsselumdrehung (Fahrer- oder Beifahrerseite) sollen sich alle Riegel öffnen bzw. schließen. Dadurch kann das Abschließen einzelner Türen vermieden werden und der Komfort wird erhöht.

Zu Feature 2:

Falls sich (z.B. bei einer größeren Reise) Wertgegenstände im Kofferraum befinden, soll es zusätzlich möglich sein, diesen unabhängig von der Zentralverriegelung absperren zu können. Soll hingegen der Kofferraum z.B. von anderen Personen aus- oder eingeladen werden, wäre es sinnvoll den Kofferraum unabhängig von der Zentralverriegelung aufsperrn zu können.

Zu Feature 3:

Da durch das gleichzeitige Verschließen aller Türen die Gefahr des „Sich-selbst-Ausschließens“ evtl. erhöht wäre, ist es notwendig, gerade für diesen Fall, besondere Vorkehrungen zu treffen. Darüber hinaus sind auch noch weitere Funktionen möglich, die dem Komfort und der Sicherheit der Fahrgäste nutzen.

Folgende Signale werden von Sensoren geliefert:

- **DOOR_DRIVER_LOCK_E**
Sobald der Schlüssel im Türschloss der Fahrertür um mindestens 30° nach rechts gedreht wird, wird dieses Signal ausgesendet.
- **DOOR_DRIVER_UNLOCK_E**
Sobald der Schlüssel im Türschloss der Fahrertür um mindestens 30° nach links gedreht wird, wird dieses Signal ausgesendet.
- **DOOR_CODRIVER_LOCK_E**
Sobald der Schlüssel im Türschloss der Beifahrertür um mindestens 30° nach links gedreht wird, wird dieses Signal ausgesendet.

- **DOOR_CODRIVER_UNLOCK_E**
Sobald der Schlüssel im Türschloss der Beifahrertür um mindestens 30° nach rechts gedreht wird, wird dieses Signal ausgesendet.
- **DOOR_BOOT_LOCK_C**
Sobald der Schlüssel im Türschloss der des Kofferraumdeckels um 90° nach rechts gedreht wird, wird diese Bedingung wahr.
- **DOOR_BOOT_UNLOCK_C**
Sobald der Schlüssel im Türschloss der des Kofferraumdeckels genau senkrecht steht, wird diese Bedingung wahr.
- **DOOR_BOOT_ZV_C**
Sobald der Schlüssel im Türschloss der des Kofferraumdeckels um 90° nach links gedreht wird, wird diese Bedingung wahr.
- **DOOR_DRIVER_OPENED_C**
Signal eines Schalters, der überprüft, ob die Fahrertür offen ist.
- **DOOR_BEHIND_DRIVER_OPENED_C**
Signal eines Schalters, der überprüft, ob die Tür links hinten offen ist.
- **DOOR_CODRIVER_OPENED_C**
Signal eines Schalters, der überprüft, ob die Beifahrertür offen ist.
- **DOOR_BEHIND_CODRIVER_OPENED_C**
Signal eines Schalters, der überprüft, ob die Tür rechts hinten offen ist.
- **DOOR_BOOT_OPENED_C**
Signal eines Schalters, der überprüft, ob der Kofferraumdeckel offen ist.

Mit Hilfe der folgenden Signale können die entsprechenden Aktoren angesprochen werden:

- **DOOR_DRIVER_LOCKED_C**
Mit diesem Signal wird die Verriegelung der Fahrertür angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_BEHIND_DRIVER_LOCKED_C**
Mit diesem Signal wird die Verriegelung der Tür hinten links angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_CODRIVER_LOCKED_C**
Mit diesem Signal wird die Verriegelung der Beifahrertür angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_BEHIND_CODRIVER_LOCKED_C**
Mit diesem Signal wird die Verriegelung der Tür hinten rechts angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_BOOT_LOCKED_C**
Mit diesem Signal wird die Verriegelung des Kofferraumdeckels angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.

Folgende Signale sind generell verfügbar:

- **CAR_START_C**
Zeigt an, ob die Zündung an (=TRUE) oder aus (=FALSE) ist.
- **CAR_GAS_D**
Gibt die Stellung des Gaspedals in Prozent an (0% = kein Gas, 100% = Vollgas)
- **CAR_SPEED_D**
aktuelle Geschwindigkeit des Fahrzeugs zwischen 0 und 250 km/h
- **CAR_STEERING_WHEEL_POS_D**
Lenkradstellungswinkel in Grad (-150° (links) bis +150° (rechts))
- **CAR_BLINKER_LEFT_C**
Zeigt an, dass der linke Blinker an ist
- **CAR_BLINKER_RIGHT_C**
Zeigt an, dass der rechte Blinker an ist

7.1.8 Anforderungen und Signale für das Verriegelungs-System (Iteration 2)

Sehr geehrte Damen und Herren,

nach der erfolgreichen Einführung der Zentralverriegelung soll nun eine Erweiterung dieses Systems stattfinden. Dabei denken wir an eine Fernbedienung, mit deren Hilfe man die Türe des Fahrzeugs ansteuern kann. Auf diese grandiose Idee kamen wir, als wir uns Gedanken über den Bedienungskomfort der Schließanlage machten. Der Ausgangspunkt für die Überlegung war die Zeitersparnis und der „Fluss des Einsteigens“. Mit dem Fluss des Einsteigens ist der Workflow von Zeitpunkt an betrachtet, in dem man sich dem Fahrzeug nähert bis zu dem Zeitpunkt, in dem man im Auto sitzt und bereit ist diese zu starten. Die Idee war nun, das übliche Aufschließen mit dem Schlüssel schon auf dem Weg zu Auto machen zu können und somit ein unnötiges Zielen mit dem Schlüssel und das anschließende Drehen in die richtige Richtung zu umgehen. So kamen wir auf eine kleine Bedienungseinrichtung, die in den Schlüssel integriert sein kann und die es ermöglicht die Türen des Autos von einer bestimmten Entfernung aus möglich zu machen. Vorgesehen sind drei Knöpfe mit deren Hilfe man die Türen ansteuert:

- „Kofferraum“ entriegelt den Kofferraum. Dient ausschließlich der Ansteuerung des Kofferraums. Durch ein einmaliges Drücken öffnet oder schließt sich dieser, abhängig von der Ausgangssituation.
- „Türen schließen“ schließt alle Türen (Kofferraum ausgenommen). Ein einmaliges Drücken schließt alle Türen.
- „Türen öffnen“ öffnet die Türen. Es existieren verschiedene Modi, die durch ein drücken innerhalb der Zeitspanne einer Sekunde der Reihe nach durchgeschaltet werden. Wenn innerhalb der Sekunde kein weiterer Druck auf die Taste „Türe öffnen“ erfolgt, öffnet die Schließanlage in dem gewählten Modus.
 - Modi 1: nur die Fahrertür öffnet sich.
 - Modi 2: die Fahrer- und die Beifahrertür öffnen sich.
 - Modi 3: Alle Türen (Kofferraum ausgenommen) öffnen sich.

Ihre Aufgabe ist es nun, dieses neue Feature in das bestehende System einzubauen! Dabei können sie auf die neue Hardware zurückgreifen, die über den CAN-Bus des Systems folgende Signale einer Empfänger-Funkeinheit liefert:

- DOOR_RADIO_BOOT_E
- DOOR_RADIO_CLOSE_E
- DOOR_RADIO_OPEN_E

Folgende Signale werden von Sensoren geliefert:

- DOOR_DRIVER_LOCK_E
Sobald der Schlüssel im Türschloss der Fahrertür um mindestens 30° nach rechts gedreht wird, wird dieses Signal ausgesendet.

- **DOOR_DRIVER_UNLOCK_E**
Sobald der Schlüssel im Türschloss der Fahrertür um mindestens 30° nach links gedreht wird, wird dieses Signal ausgesendet.
- **DOOR_CODRIVER_LOCK_E**
Sobald der Schlüssel im Türschloss der Beifahrertür um mindestens 30° nach links gedreht wird, wird dieses Signal ausgesendet.
- **DOOR_CODRIVER_UNLOCK_E**
Sobald der Schlüssel im Türschloss der Beifahrertür um mindestens 30° nach rechts gedreht wird, wird dieses Signal ausgesendet.
- **DOOR_BOOT_LOCK_C**
Sobald der Schlüssel im Türschloss der des Kofferraumdeckels um 90° nach rechts gedreht wird, wird diese Bedingung wahr.
- **DOOR_BOOT_UNLOCK_C**
Sobald der Schlüssel im Türschloss der des Kofferraumdeckels genau senkrecht steht, wird diese Bedingung wahr.
- **DOOR_BOOT_ZV_C**
Sobald der Schlüssel im Türschloss der des Kofferraumdeckels um 90° nach links gedreht wird, wird diese Bedingung wahr.
- **DOOR_DRIVER_OPENED_C**
Signal eines Schalters, der überprüft, ob die Fahrertür offen ist.
- **DOOR_BEHIND_DRIVER_OPENED_C**
Signal eines Schalters, der überprüft, ob die Tür links hinten offen ist.
- **DOOR_CODRIVER_OPENED_C**
Signal eines Schalters, der überprüft, ob die Beifahrertür offen ist.
- **DOOR_BEHIND_CODRIVER_OPENED_C**
Signal eines Schalters, der überprüft, ob die Tür rechts hinten offen ist.
- **DOOR_BOOT_OPENED_C**
Signal eines Schalters, der überprüft, ob der Kofferraumdeckel offen ist.
- **DOOR_RAID_PROTECTION_C**
Signal eines Schalters, der einen Sicherheitsmechanismus einschalten soll.
- **DOOR_RADIO_BOOT_E**
Signal eines Schalters, der sich auf der Fernbedienung des Autoschlüssels befindet und das Öffnen des Kofferraums auslösen soll
- **DOOR_RADIO_OPEN_E**
Signal eines Schalters, der sich auf der Fernbedienung des Autoschlüssels befindet und das öffnen der Türen steuern soll.
- **DOOR_RADIO_CLOSE_E**
Signal eines Schalters, der sich auf der Fernbedienung des Autoschlüssels befindet und das schließen der Türen steuern soll.

Mit Hilfe der folgenden Signale können die entsprechenden Aktoren angesprochen werden:

- **DOOR_DRIVER_LOCKED_C**
Mit diesem Signal wird die Verriegelung der Fahrertür angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_BEHIND_DRIVER_LOCKED_C**
Mit diesem Signal wird die Verriegelung der Tür hinten links angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_CODRIVER_LOCKED_C**
Mit diesem Signal wird die Verriegelung der Beifahrertür angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_BEHIND_CODRIVER_LOCKED_C**
Mit diesem Signal wird die Verriegelung der Tür hinten rechts angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_BOOT_LOCKED_C**
Mit diesem Signal wird die Verriegelung des Kofferraumdeckels angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.

Folgende Signale sind generell verfügbar:

- **CAR_START_C**
Zeigt an, ob die Zündung an (=TRUE) oder aus (=FALSE) ist.
- **CAR_GAS_D**
Gibt die Stellung des Gaspedals in Prozent an (0% = kein Gas, 100% = Vollgas)
- **CAR_SPEED_D**
aktuelle Geschwindigkeit des Fahrzeugs zwischen 0 und 250 km/h
- **CAR_STEERING_WHEEL_POS_D**
Lenkradstellungswinkel in Grad (-150° (links) bis +150° (rechts))
- **CAR_BLINKER_LEFT_C**
Zeigt an, dass der linke Blinker an ist
- **CAR_BLINKER_RIGHT_C**
Zeigt an, dass der rechte Blinker an ist

Nachtrag: Die folgenden Berichte wurden uns von der Stelle für Öffentlichkeitsarbeit angetragen. Bitte lesen sie diese durch und erweitern/verändern sie nach Möglichkeit das Modell soweit, dass die Mängel bzw. die Gefahrenquellen für das bestehende System nicht mehr in Betracht kommen.

Funksignal auf Abwegen

Hubert K., der von einem Kinobesuch nach Hause fahren wollte, stieß auf ein kleines Problem mit seinem Wagen.

Hubert K., der nach dem Besuch einer Vorstellung im Dietrich-Kino kam, wollte wie gewohnt in seinen Wagen einsteigen und nach Hause fahren. Er schlenderte über den schlecht beleuchteten Parkplatz und suchte nach seinem Auto. Als er es fand, schloss er mit Hilfe des Schlüssels seinen Wagen auf und stieg ein. Er wunderte sich über die verschobenen Sitze und die falsch eingestellten Spiegel in seinem Wagen und korrigierte die Stellung dieser. Nach kurzer Zeit wurde ihm aber bewusst, dass es sich bei dem Wagen nicht um seinen handelte, da es von der Ausstattung her nicht sein Wagen sein konnte. Das fiel ihm als erstes an dem Autoradio auf, das er ersetzt hatte. Dieser Wagen verfügte an der Stelle, an der eigentlich ein modernes Sony-Radio stecken sollte, noch immer über ein vom Werk geliefertes Modell eines Autoradios. Hubert K. meldete diesen Vorfall der Polizei, in der Absicht, dass der eigentliche Wagenbesitzer sich nicht bei dieser über einen Einbruch in seinen Wagen beschweren würde. Der Wagen von Hubert K. stand keine 20 Meter von dem Wagen entfernt, in dem er gesessen hat und in dem schlechten Licht, der auf dem Parkplatz herrschte, war das baugleiche Modell kaum zu unterscheiden, trotz des leicht unterschiedlichen Farbtons.

Auto, Motor und Sport

Die Schließautomatik der Zukunft.

Mecking führte mit der neuen Schließanlage für ihr Oberklassemodell neue Maßstäbe ein. Die entwickelte Schließautomatik ist für die bequemeren Menschen unter den Autofahrern gedacht. Die Idee der Schließautomatik ist dabei, dass man per Knopfdruck eines kleinen Senders einem Empfangsgerät im inneren des Wagens einen Öffnen-Befehl gibt. Dabei trägt der Fahrer zusätzlich zu seinem Schlüssel einen kleinen handlichen Sender am Schlüsselbund, der nicht größer als eine Zigaretenschachtel ist. In diesem kleinen Päckchen befindet sich die notwendige Technik um der Türe einen eindeutigen Sicherheitscode zu übermitteln. Das Gegenstück im Inneren des Fahrzeugs nimmt das Signal auf und öffnet dann die gewünschten Türen.

Die Ampelräuber

In neuster Zeit häufen sich die gewaltsamen Übergriffe auf Autofahrer, die auf ihre Grünphase an der Ampel warten.

So lustig der Titel auch klingen mag, so schrecklich sind die Verbrechen, deren Art schon einen Namen bekommen hat. Der Name der für diese Art von Raub festgelegt wurde, nennt sich „Carchecking“. Die dreisten Diebe werden immer unverschämter. So warten sie an Kreuzungen auf die Rotphase der Ampeln und rauben dann Fahrzeuge aus, deren Türen nicht verschlossen sind. Meist sind Frauen die Opfer dieser schändlichen Überfälle. Die Räuber nutzen dabei das Sicherheitsgefühl aus, dass ein Auto ausstrahlt. So glaubt sich jeder in Sicherheit, wenn er hinter seinem Lenkrad sitzt. Nur vergessen die meisten, dass die Tür neben ihnen während der Fahrt meist nicht verschlossen ist und somit einen Angriffspunkt auf den Fahrer und damit den Wertsachen des Fahrers darstellt. Die Verbrecher haben es meist auf kleinere Wertgegenstände wie beispielsweise Geldbörsen, Schmuck oder Taschen abgesehen. Doch hin und wieder werden die Fahrer genötigt auszusteigen und das Fahrzeug dem Verbrecher zu überlassen. Die Polizei versucht gerade vergeblich Gegenmaßnahmen einzuleiten. Doch leider sind die Räuber sehr variabel, was die Orte des Verbrechens angeht, denn in einer Stadt, selbst in einer kleinen, gibt es viele Kreuzungen.

7.1.9 Anforderungen und Signale für das Verriegelungs-System (Iteration 3)

Sehr geehrte Damen und Herren,

um einen höheren Grad an Sicherheit und Komfort in unser neuestes Fahrzeug zu bringen, haben wir uns entschlossen, eine Änderung am bisherigen Carchecking-Sicherungs-System vorzunehmen. Diese Veränderung ist für die Bedienungsfreundlichkeit gedacht. Der Ausgangspunkt ist eine aktive Charchecking-Sicherung. Die momentane Situation lässt es nicht zu, die Türen von außen zu öffnen, sollte ein Dieb sich z.B. an einer Ampel an das Fahrzeug heranmachen. Nun soll es aber möglich sein, die Türen von innen her zu öffnen. Zu diesem Zweck wurden neue Türöffner in das Innere des Autos eingebaut. Die Funktionalität des alten Türschließmechanismus soll soweit erhalten bleiben. Folgende Modifikationen sind zu realisieren: Wenn die Carchecking-Sicherung aktiv ist und das Fahrzeug sich im Stand befindet, soll es möglich sein, die Türen von innen zu öffnen. Dabei wird der Hebel im Inneren der Fahrzeugkabine betätigt. Beim ersten Drücken des Hebels öffnet sich der Schließmechanismus der Türe. Beim zweiten Drücken des Hebels schwingt die Türe auf. Wird die Türe wieder mechanisch geschlossen, soll der Schließmechanismus offen bleiben. Erst, wenn das Fahrzeug sich wieder in Bewegung setzt, soll der Schließmechanismus der Türe diese wieder verriegeln.

Die Idee, die hinter dieser Weiterentwicklung steckt, ist die Tatsache, dass es möglich sein soll z.B. eine Person von einem Ort abzuholen, ohne die Sicherheitsfunktionen deaktivieren zu müssen (Person vom Bahnhof abholen). Ein weiteres Beispiel ist das Einwerfen eines Briefs in einen Briefkasten bei laufendem Motor.

Folgende Signale werden von Sensoren geliefert:

- **DOOR_DRIVER_LOCK_E**
Sobald der Schlüssel im Türschloss der Fahrertür um mindestens 30° nach rechts gedreht wird, wird dieses Signal ausgesendet.
- **DOOR_DRIVER_UNLOCK_E**
Sobald der Schlüssel im Türschloss der Fahrertür um mindestens 30° nach links gedreht wird, wird dieses Signal ausgesendet.
- **DOOR_CODRIVER_LOCK_E**
Sobald der Schlüssel im Türschloss der Beifahrertür um mindestens 30° nach links gedreht wird, wird dieses Signal ausgesendet.
- **DOOR_CODRIVER_UNLOCK_E**
Sobald der Schlüssel im Türschloss der Beifahrertür um mindestens 30° nach rechts gedreht wird, wird dieses Signal ausgesendet.
- **DOOR_BOOT_LOCK_C**
Sobald der Schlüssel im Türschloss der des Kofferraumdeckels um 90° nach rechts gedreht wird, wird diese Bedingung wahr.
- **DOOR_BOOT_UNLOCK_C**
Sobald der Schlüssel im Türschloss der des Kofferraumdeckels genau senkrecht steht, wird diese Bedingung wahr.

- **DOOR_BOOT_ZV_C**
Sobald der Schlüssel im Türschloss der des Kofferraumdeckels um 90° nach links gedreht wird, wird diese Bedingung wahr.
- **DOOR_DRIVER_OPENED_C**
Signal eines Schalters, der überprüft, ob die Fahrertür offen ist.
- **DOOR_BEHIND_DRIVER_OPENED_C**
Signal eines Schalters, der überprüft, ob die Tür links hinten offen ist.
- **DOOR_CODRIVER_OPENED_C**
Signal eines Schalters, der überprüft, ob die Beifahrertür offen ist.
- **DOOR_BEHIND_CODRIVER_OPENED_C**
Signal eines Schalters, der überprüft, ob die Tür rechts hinten offen ist.
- **DOOR_BOOT_OPENED_C**
Signal eines Schalters, der überprüft, ob der Kofferraumdeckel offen ist.
- **DOOR_RAID_PROTECTION_C**
Signal eines Schalters, der einen Sicherheitsmechanismus einschalten soll.
- **DOOR_RADIO_BOOT_E**
Signal eines Schalters, der sich auf der Fernbedienung des Autoschlüssels befindet und das Öffnen des Kofferraums auslösen soll
- **DOOR_RADIO_OPEN_E**
Signal eines Schalters, der sich auf der Fernbedienung des Autoschlüssels befindet und das öffnen der Türen steuern soll.
- **DOOR_RADIO_CLOSE_E**
Signal eines Schalters, der sich auf der Fernbedienung des Autoschlüssels befindet und das schließen der Türen steuern soll.
- **HW_DOOR_DRIVER_GRIFF_INNEN_E**
Signal zum Entriegeln und Öffnen der Fahrertüre mit dem Hebel im Innenraum des Autos.
- **HW_DOOR_CODRIVER_GRIFF_INNEN_E**
Signal zum Entriegeln und Öffnen der Beifahrertüre mit dem Hebel im Innenraum des Autos.
- **HW_DOOR_BEHIND_DRIVER_GRIFF_INNEN_E**
Signal zum Entriegeln und Öffnen der Türe hinter der Fahrertüre mit dem Hebel im Innenraum des Autos.
- **HW_DOOR_BEHIND_CODRIVER_GRIFF_INNEN_E**
Signal zum Entriegeln und Öffnen der Türe hinter der Beifahrertüre mit dem Hebel im Innenraum des Autos.

Mit Hilfe der folgenden Signale können die entsprechenden Aktoren angesprochen werden:

- **DOOR_DRIVER_LOCKED_C**
Mit diesem Signal wird die Verriegelung der Fahrertür angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_BEHIND_DRIVER_LOCKED_C**
Mit diesem Signal wird die Verriegelung der Tür hinten links angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_CODRIVER_LOCKED_C**
Mit diesem Signal wird die Verriegelung der Beifahrertür angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_BEHIND_CODRIVER_LOCKED_C**
Mit diesem Signal wird die Verriegelung der Tür hinten rechts angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_BOOT_LOCKED_C**
Mit diesem Signal wird die Verriegelung des Kofferraumdeckels angesteuert. Sobald diese Condition TRUE wird, wird sie verriegelt.
- **DOOR_LOCK_HEATER_C**
Aktiviert Heizspulen in den Türen (um die Türschlösser)

Folgende Signale sind generell verfügbar:

- **CAR_START_C**
Zeigt an, ob die Zündung an (=TRUE) oder aus (=FALSE) ist.
- **CAR_GAS_D**
Gibt die Stellung des Gaspedals in Prozent an (0% = kein Gas, 100% = Vollgas)
- **CAR_SPEED_D**
aktuelle Geschwindigkeit des Fahrzeugs zwischen 0 und 250 km/h
- **CAR_STEERING_WHEEL_POS_D**
Lenkradstellungswinkel in Grad (-150° (links) bis +150° (rechts)) JETZT steuerbar !
- **CAR_BLINKER_LEFT_C**
Zeigt an, dass der linke Blinker an ist
- **CAR_BLINKER_RIGHT_C**
Zeigt an, dass der rechte Blinker an ist
- **CAR_TEMPERATURE_SENSOR_D**
Zeigt die momentane Außentemperatur des Fahrzeugs an.
- **CAR_CRASH_E**
Falls ein Unfall auftritt.
- **HW_BLINK_LEFT_C**
schaltet die Blinklichter auf der linken Fahrzeug Seite an (=TRUE) bzw. aus (=FALSE)

- HW_BLINK_RIGHT_C

schaltet die Blinklichter auf der rechten Fahrzeug Seite an (=TRUE) bzw. aus (=FALSE)

Nachtrag: Die folgenden Berichte wurden uns von der Stelle für Öffentlichkeitsarbeit ange-
tragen. Bitte lesen sie diese durch und erweitern/verändern sie nach Möglichkeit das Modell
soweit, dass die Mängel bzw. die Gefahrenquellen für das bestehende System nicht mehr in
Betracht kommen.

Panne der Polizei

Eine Polizeistreife stand vor verschlossenen Türen

In einer eisigen Nacht im Laufe der letzten Woche kam es zu einem lustigen Zwischenfall.
Eine Streife der örtlichen Polizei versuchte nach der Klärung eines Unfalls auf der Kreuzung
von Frauenstraße und Olgastraße wieder in ihren Wagen zu gelangen. Leider hatte der Eisre-
gen und die frostigen Temperaturen von unter -5 Grad ihre Arbeit schon erledigt und ließen
die Türen des Fahrzeugs zufrieren. So waren die Polizisten gezwungen, Unterstützung an-
zufordern, um wieder in das Innere ihres Wagens gelangen zu können. Hier sieht man, dass
die Natur nicht mal vor dem Arm des Gesetzes halt macht. Nach Auskunft der Polizeistelle,
werden die Streifenwagen in der nächsten Zeit nachgerüstet. Bei dieser Nachrüstaktion sollen
die Schlösser der Fahrzeuge mit Heizspulen versehen werden, die in der Lage sind, ab einer
bestimmten Außentemperatur die Schlösser des Fahrzeugs zu erwärmen und so einen Zwi-
schenfall wie den letzten zu vermeiden.

ZV als Hindernis

*Bei einem Unfall auf der Autobahn kam es bei der Rettung der Verletzten aufgrund
der modernen Technik zu Schwierigkeiten.*

Auf der Autobahn von Stuttgart nach München auf der Höhe von Odelzhausen kam es
zu einem Unfall bei dem 5 Fahrzeuge ineinander rasten. Dabei kam ein Fahrzeug von der
Autobahn ab und blieb in einem angrenzenden Acker stehen. Die Elektrik des Fahrzeugs be-
kam einen Schaden und es war nicht möglich, die Türen des Fahrzeugs zu öffnen, obwohl dies
keinen allzu großen Schaden erlitten hatte. Die Feuerwehr war gezwungen, das Auto komplett
aufzuschneiden, um die Insassen zu bergen, da der Türöffnungsmechanismus nicht mehr mit
Strom versorgt wurde und die ZV alle Türen geschlossen hielt.

7.2 Anforderungstabelle

Beschreibung Airbag-System	Form	Begründung (für Entwurfsentscheidung Verweis auf Szenario)	Akzept. krit.
1 Airbag-System	Überschrift		
Das Airbag-System soll vom Fahrer aus mittels einem Knopf aktivierbar bzw. deaktivierbar sein. Das System soll den Fahrer und die anderen Insassen vor Verletzungen bei einem Autounfall schützen. Das System wird aktiv, sobald die Zündung des Fahrzeugs eingeschaltet ist.			
1.1 Airbag aktiviert	Überschrift		
Wenn die Zündung eingeschaltet ist und der Kontrollknopf aktiviert wurde, dann ist das Airbag-System aktiv.	Anforderung	Das Airbag-System soll nur dann auslösen, wenn die Insassen im Fahrzeug sind. Dies ist der Fall, wenn die Zündung eingeschaltet ist. In dem Fall soll der Airbag nur auf Wunsch des Fahrers im Falle eines Unfalls auslösen, also wenn der Kontrollknopf aktiviert ist.	
1.1.1 Airbag auslösen	Überschrift		
<ul style="list-style-type: none"> - Vordere Airbags werden in jedem Fall ausgelöst, außer der Aufprall erfolgt nur von einer der Seiten. - Bei einem seitlichen Frontalaufprall werden die entsprechende Seitenairbags und beide Frontairbags ausgelöst. - Bei einem seitlichen Aufprall werden beide Seitenairbags ausgelöst. 	Anforderung	<ul style="list-style-type: none"> - Somit ist der Schutz beider Insassen bei einem Frontalzusammenstoß gewährleistet. Bei einem seitlichen Aufprall erfolgt von vorne keine Krafteinwirkung, somit werden nur die seitlichen Airbags ausgelöst. - Hier erfolgt Krafteinwirkung von vorne und von der Seite, daher werden alle Airbags ausgelöst. 	

1.2 Airbag deaktiviert	Überschrift		
Selbst bei eingeschalteter Zündung wird das Airbag-System nicht ausgelöst, unabhängig aus welcher Richtung ein Unfall erfolgt.	Anforderung	Aktiviertes Airbag-System könnte unter Umständen Gefährdung für Insassen darstellen, deshalb ist es auf Wunsch deaktivierbar (z.B. Kindersitz)	
1.3 Airbag wurde ausgelöst	Überschrift		
Nur durch Reparatur kann das Airbag-System wieder in einen arbeitsbereiten Zustand überführt werden. Davor ist ein erneutes Auslösen des Systems nicht mehr möglich.	Erläuterung		
2 Airbagdruck	Überschrift		
Der Airbagdruck wird der Geschwindigkeit angepasst, wobei der Druck von 0 -100 % bedeutet: 0 % Airbag wird nicht ausgelöst und 100 % Airbagkissen wird maximal aufgeblasen. Der Airbag wird erst ab einer bestimmten Geschwindigkeit ausgelöst.	Erläuterung		
2.1 vorderer Airbagdruck	Überschrift		
Bei einem Crash von vorne werden die vorderen Airbags ausgelöst, wobei der Druck von der Geschwindigkeit des Autos abhängt.	Anforderung	Damit es nicht zu zusätzlichen Verletzungen der Insassen im Auto kommt.	
2.2 seitlicher Airbagdruck	Überschrift		
Bei einem seitlichen Crash werden die seitlichen Airbags ausgelöst, wobei der Druck von der Geschwindigkeit des Autos abhängt.	Anforderung	Damit es nicht zu zusätzlichen Verletzungen der Insassen im Auto kommt.	
3. Knieairbag	Überschrift		
Anbringen eines Knieairbags an der Fahrerseite und Beifahrerseite. Der Knieairbag löst sich gleichzeitig mit den vorderen Airbags aus.	Anforderung	Die Insassen, die sich auf der Fahrer- und Beifahrerseite befinden, werden zusätzlich gesichert, da der gesamte Beckenbereich zurückgehalten wird (siehe Szenario IT1).	

4. Airbags für Hintersitze	Überschrift		
Es gibt Seiten- und Vorderairbags für die Hintersitze, die sich gleichzeitig mit den vorderen auslösen. Bei einem Seitenaufprall lösen die Seitenairbags vorne und hinten aus und bei einem frontalen Zusammenstoß lösen die Frontairbags vorne und hinten aus.	Anforderung	Um die Insassen auf den Rücksitzen bei einem Unfall auch zu schützen (siehe Szenario IT6).	
5. Annäherungs-System	Überschrift		
Ist die Zündung und der Airbag eingeschaltet, so soll der Airbag an einen Annäherungssensor angebunden werden. Dieser Sensor reagiert auf die Entfernung großer Objekte, die sich vor dem Fahrzeug befinden.	Erläuterung		
5.1 Annäherungssensor	Überschrift		
6. Deaktivierung des Beifahrerairbags	Überschrift		
Wenn sich kein Objekt oder ein Kindersitz auf der Beifahrerseite befindet, so soll der Beifahrerairbag und der Seitenairbag auf der Beifahrerseite nicht ausgelöst werden.	Erläuterung		
6.1 Kein Objekt auf Beifahrerseite	Überschrift		
Die Zündung und das Airbag-System ist an. Es befindet sich kein Objekt auf der Beifahrerseite. Der Beifahrerairbag und der Seitenairbag auf der Beifahrerseite werden nicht ausgelöst.	Anforderung	Vermeidung von zusätzlichen Schäden am Auto.	
6.2 Kindersitz auf Beifahrerseite	Überschrift		
Die Zündung und das Airbag-System ist an. Es befindet sich ein Kindersitz auf der Beifahrerseite. Der Beifahrerairbag und der Seitenairbag auf der Beifahrerseite werden nicht ausgelöst.	Anforderung	Vermeidung von Verletzungen von Kindern im Kindersitz	

7. Temperatur-System	Überschrift		
Bei niedrigen Temperaturen kann es dazu kommen, dass die Airbag-Sensoren einfrieren und nicht mehr funktionsfähig sind.	Erläuterung		
7.1 Airbag-Sensoren beheizen	Überschrift		
Die Zündung und das Airbag-System ist an. Bei niedrigen Temperaturen werden die Airbag-Sensoren durch eine Heizspule geheizt.	Anforderung	Der Ausfall des Airbag-Systems durch niedrige Temperaturen wird behoben.	

Tabelle 8: Eine von den Studierenden erstellte Anforderungstabelle zum Airbag-System.

7.3 Merkmaldiagramm

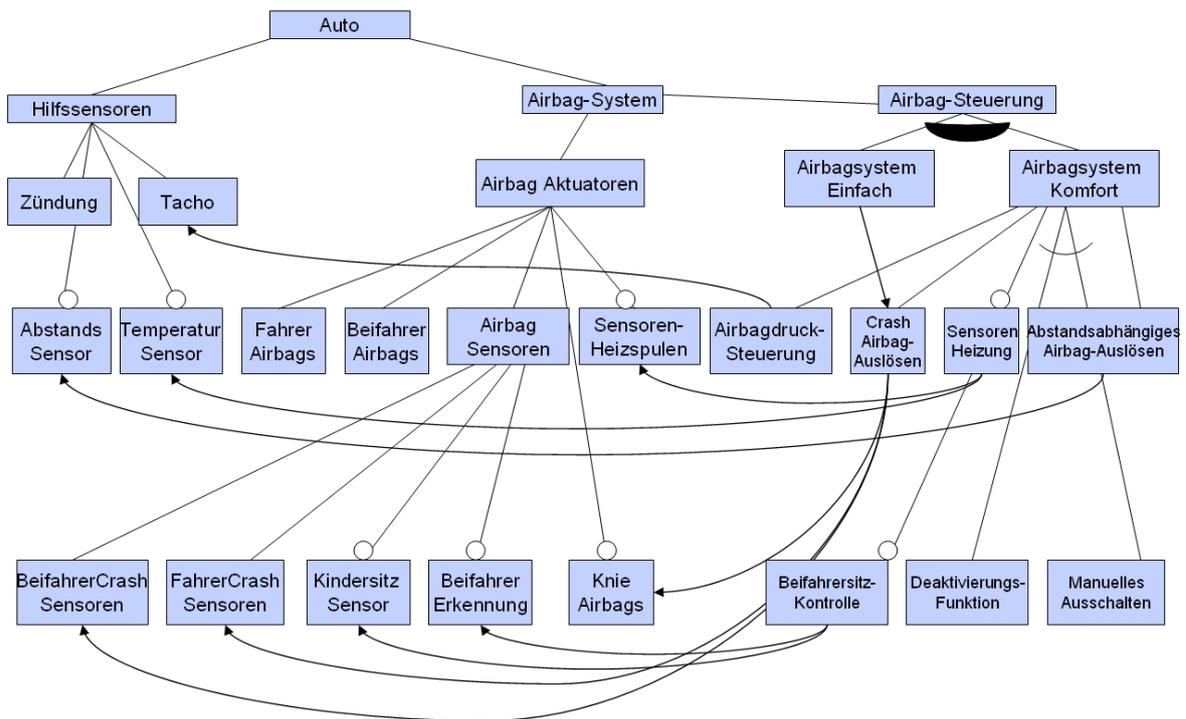


Abbildung 13: Ein von den Studierenden erstelltes Merkmaldiagramm — fehlerhaft hierarchisiert und fehlerhafte Verwendung der Abhängigkeiten ist erkennbar; es ist nicht funktionsorientiert sondern vor allem Hardware-orientiert.

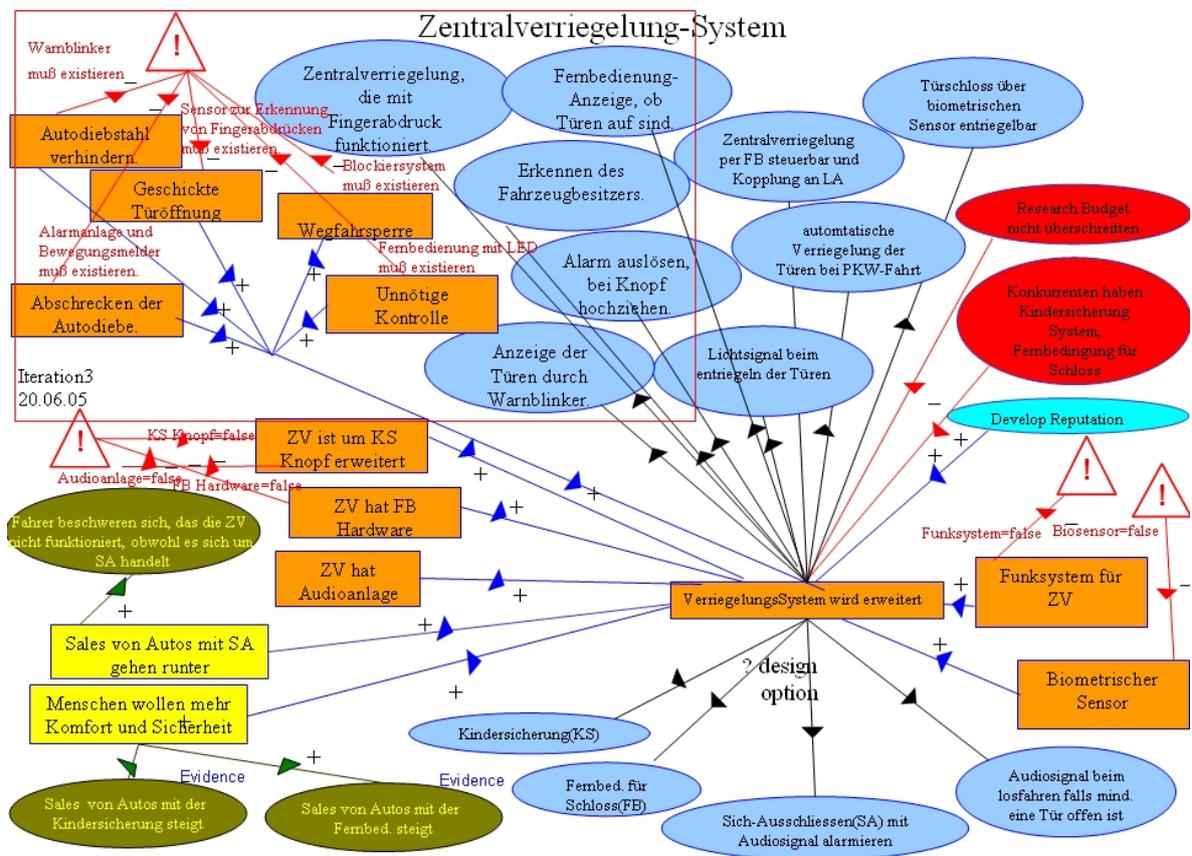


Abbildung 15: Das Entscheidungsmodell der Zentralverriegelungsgruppe — man erkennt die Verwendung von Szenarien als Optionen.

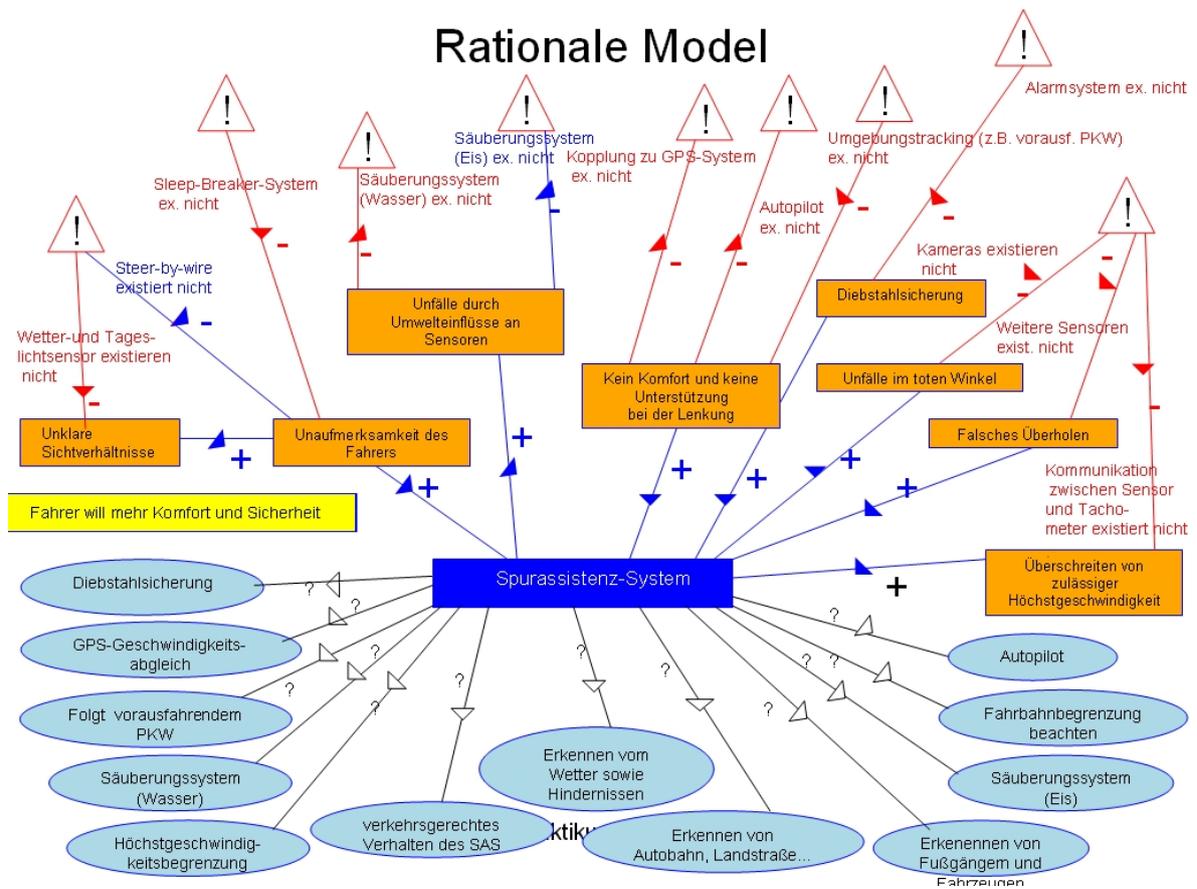


Abbildung 16: Das Entscheidungsmodell der Spurassistentengruppe.

7.5 Szenarien

ID	Szenarion	Betroffene Komponente
IT3-1: Kindersitz- erkennungs- Sensor	Umwelt: Durch das Aufblasen des Airbags werden Kinder, die sich auf dem Beifahrersitz befinden, verletzt. Um das zu vermeiden, wurde ein neuer Kindersitzerkennungs-Sensor entwickelt. System: Beim Erkennen eines Kindersitzes auf der Beifahrerseite wird auf keinen Fall der Airbag ausgelöst.	Kindersitz- erkennungs- Sensor
IT3-2: Heizen der Airbag- Sensoren	Umwelt: Durch vereiste Airbag-Sensoren im Winter funktioniert das Airbag-System fehlerhaft. Es wurde ein Temperatur-Sensor entwickelt, der die Außentemperatur des Fahrzeuges liefert. System: Falls der Temperatur-Sensor einen negativen Wert liefert, werden die Heizspulen für die Airbag-Sensoren aktiviert.	Temperatur- Sensor, Heizspulen
IT3-3: Knie Air- bag	Umwelt: Für den Schutz vor menschlichen Verletzungen bei Unfällen reichen die frontal und seitlich eingebauten Airbags nicht aus. Durch den Knie Airbag wird der Beckenbereich der Insassen umgehend und effektiv zurückgehalten und damit wird die Rückenhalterung stabilisiert. System: Bei einem Aufprall wird der Knie Airbag gleichzeitig mit dem Fahrer- bzw. Beifahrer-Airbag ausgelöst.	Unteres Seg- ment der Instrumen- tentafel
IT3-4: Anti- Sliding-Bag	Umwelt: Um bei einem Unfall das Abtauchen der Insassen unter dem Gurt zu verhindern, wurde der Anti-Sliding-Bag entwickelt. Er hält den Insassen in einer aufrechten Position, so dass er beim Aufprall den gesamten Frontairbag belastet, nicht nur den unteren Teil. System: Der Anti-Sliding-Bag befindet sich unter der Sitzfläche. Bei einem Frontalzusammenstoß befüllt der Gasgenerator eine metallene Tasche, die den vorderen Teil des Sitzpolsters anhebt und so eine stoppende Barriere vor der Hüfte des Insassen bildet. Das verwendete Stahlblech ist so dünn, dass es sich deformiert und die Form des Beckens annimmt. Gleichzeitig wird durch ein Druckventil ein gleich bleibender Innendruck während der gesamten Unfalldauer beibehalten.	Sitze

Tabelle 9: Die Szenarien der Entscheidungsmodellgruppen

7.6 Statemate-Modell am Beispiel vom Spurassistenten-System

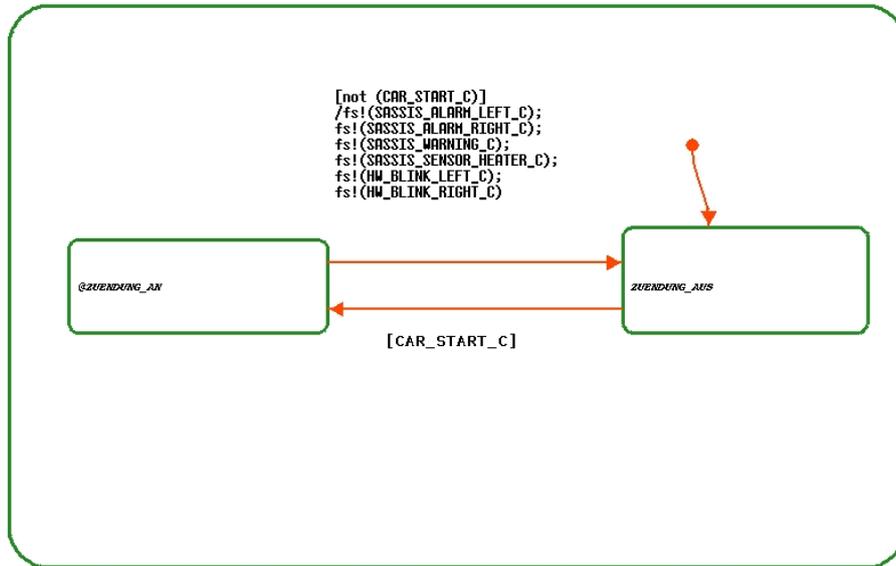


Abbildung 17: Spurassistenten-System Statechart 1

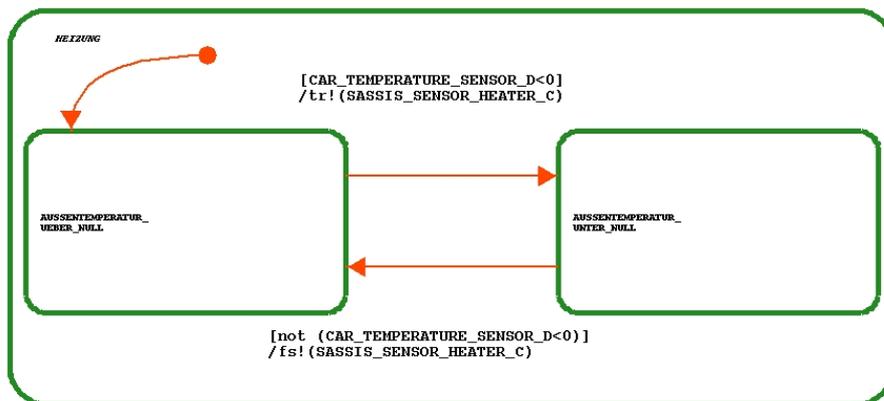


Abbildung 18: Spurassistenten-System Statechart 2

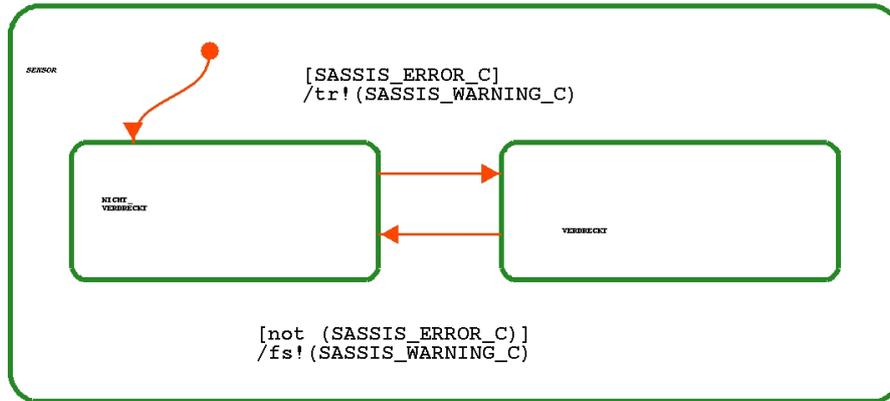


Abbildung 19: Spurassistentz-System Statechart 3

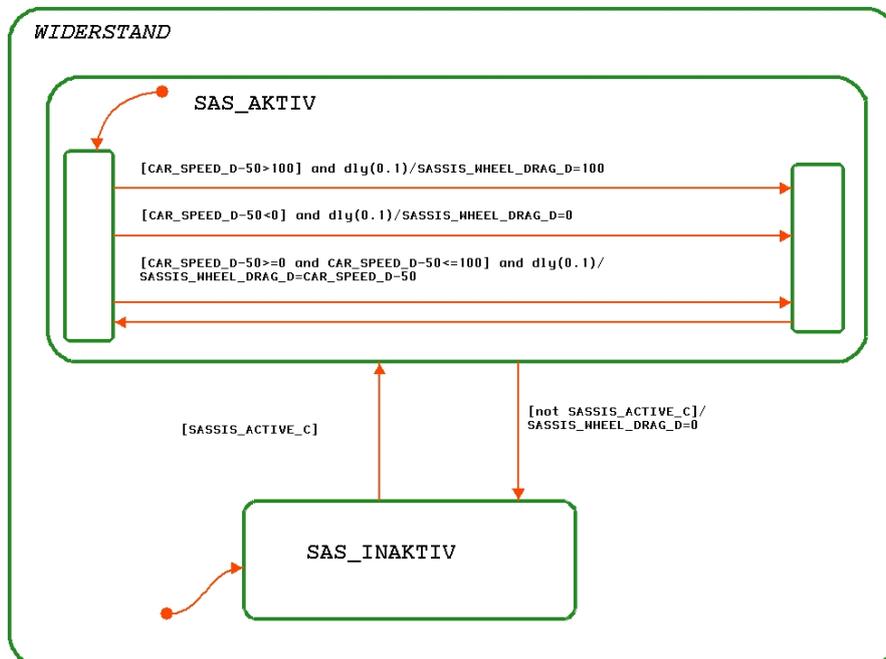


Abbildung 20: Spurassistentz-System Statechart 4

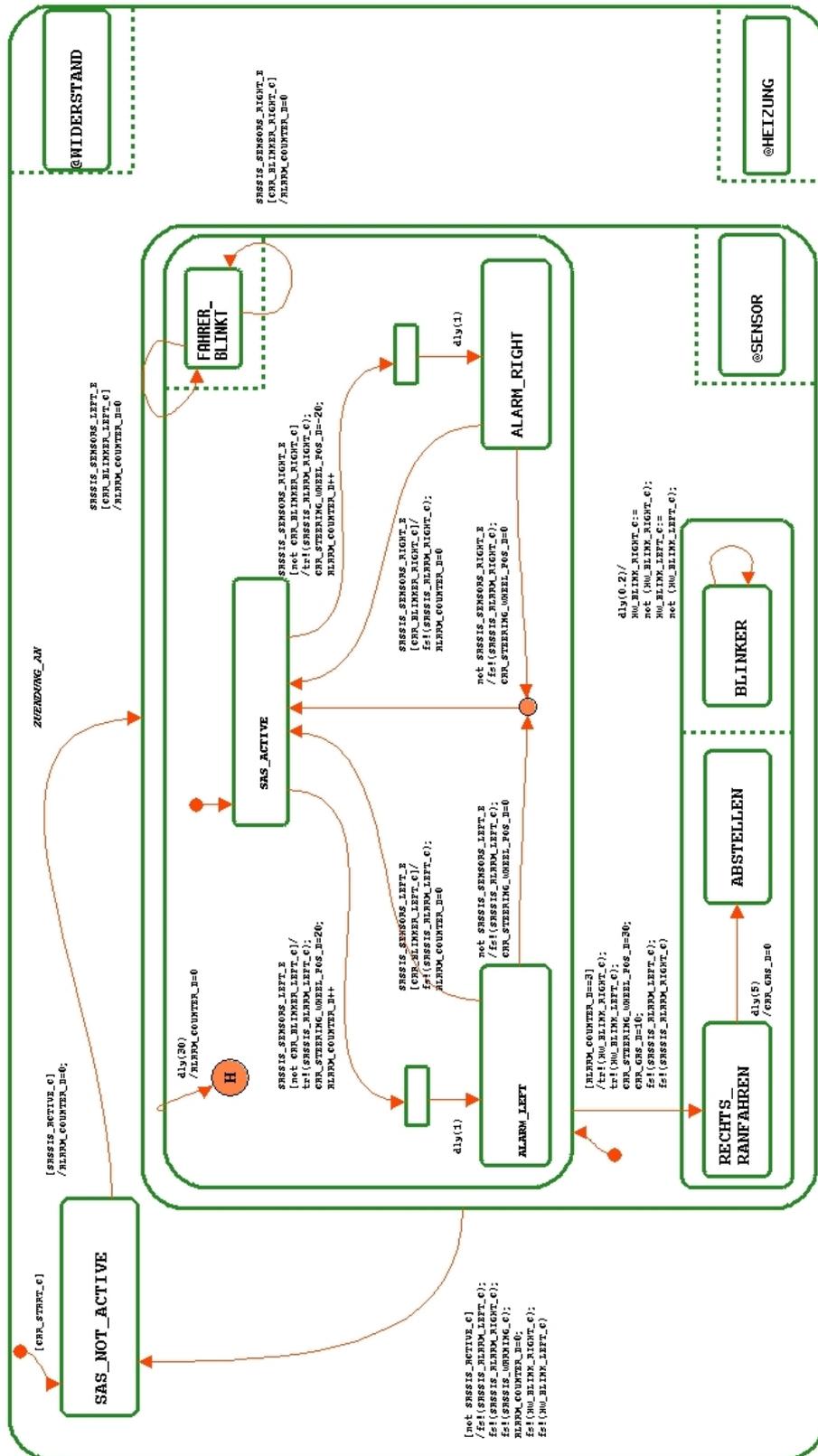


Abbildung 21: Spurrassistenz-System Statechart 5

7.7 Projektprotokoll

Projektprotokoll für Gruppe 1-2-1 Iteration Nummer 3

Verantwortlicher: Studierender A/Studierender B

Im vorliegenden Projektprotokoll sollen die einmal wöchentlich stattfindenden Projektteammeetings protokolliert werden. Dabei sollen vor allem folgende Aspekte besprochen werden:

- Ziele und Anforderungen des Systems
- Planung des Entwicklungsvorgehens, Aufteilung der Arbeiten sowie Zeitbedarf
- Beschreibung von mindestens vier Zukunftsszenarien (für die gesamte Iteration)
- Klärung spezifischer Anforderungen, die aus den Pressemeldungen abgeleitet werden
- Klärung technischer Fragestellungen

Ziele System und Planung Vorgehen

Kurzbeschreibung der Ziele des Systems:

- Der Airbag soll an einen Annäherungssensor angebunden werden, so dass ein frühzeitiges Auslösen des Airbags ermöglicht wird.
- Deaktivierung des Beifahrerairbags, um ein unnötiges Auslösen des Airbags zu vermeiden, falls sich kein Objekt auf der Beifahrerseite befindet. Oder es befindet sich ein Kindersitz auf der Beifahrerseite; durch Nicht-Auslösen des Airbags verhindert man Verletzungen von Kindern.
- Ausfall der Sensoren bei niedrigen Temperaturen wird vermieden.

Kurzbeschreibung der wesentlichen Arbeiten und der benötigten Prozessschritte:

- Aufgabenerfassung (Verständnis der Anforderungen für das System)
- Kontextdiagramm-Erstellung
- Anforderungstemplate-Erstellung
- Projektprotokoll-Erstellung
- State-Mate Modellierung
- Protokollierung der Testfälle
- Qualitätssicherung

Kurzdarstellung der geplanten Zeitschiene (fest!):

- 20.06-23.06.2005: Anforderungsdokumente, Kontextdiagramm, Projektprotokoll, Testfälle, Qualitätssicherung

- 27.06-30.06.2005: Statematemodell, Testfälle

Aufgaben, Aufwand und Fertigstellungsgrad:

Name Verantwortlicher	Rolle	Aufgabe	Geschätzter Aufwand	Fertigstellungsgrad
A/B	Diagramm-Erstellung	Erstellung der Kontextdiagramme	1 Stunde	100% fertig gestellt nach 30 Minuten
A/B	Protokollierung der Anforderungen	Erstellung des Anforderungsdokumentes	1 Stunde	100% fertig gestellt nach 45 Minuten
A/B	Projektprotokoll	Projektprotokoll	1 Stunde	100% fertig gestellt nach 45 Minuten
A/B	Qualitätssicherer	Protokollierung der Qualitätssicherheit	1 Stunde	100% fertig gestellt nach 1 Stunde
A/B	Testerstellung	Erstellung der Testfälle	2 Stunde	0 % fertig gestellt
A/B	Statematemodellierung	Entwurf des Statematemodells	4 Stunde	0% fertig gestellt
A/B	Tester	Testen des Modells	2 Stunde	0% fertig gestellt

Szenarien

Im Folgenden sollen Zukunftsszenarien für die Umwelt und das System selber entworfen werden. Dabei ist zu beachten, dass für die Iteration von drei Wochen mindestens vier Szenarien gefunden werden sollen. Die Szenarien dienen dazu, die Stabilität der Anforderungen abzuschätzen und spezielle Entwurfsentscheidungen begründet treffen zu können. Diese Entwurfsentscheidungen werden auch als solche in der Anforderungsspezifikation erfasst, Begründungen müssen explizit auf die hier beschriebenen Szenarien verweisen.

ID	Szenario	Betroffene Komponente
IT2-1	Airbag für die hinteren Sitze, angebracht an der vorderen Sitzreihe und auch an den Seiten	Sensor
IT2-2	Eingebaute Repair-Funktion	Airbag
IT2-3	Vor der Auslösung des Airbags, soll eine Straffung des Gurtes erfolgen. Damit es nicht zu einem starken Zusammenstoß mit dem Airbag kommt.	Sicherheitsgurt
IT2-4	Nach einem Crash (Auto ist im Stillstand) bei dem die Airbags nicht ausgelöst wurden, soll kein Airbag ausgelöst werden! Vermeidung nachträglicher Verletzungen und Erleichterung der Rettungsarbeiten	Sensor

Klärung spezifischer Anforderungen aus den Pressemitteilungen

Spezifische, aus den Pressemitteilungen abgeleitete Anforderungen werden in der folgenden Tabelle kurz beschrieben, wobei auf die entsprechende Pressemitteilung verwiesen wird.

Pressemitteilung	Deaktivierung des Airbags, falls ein Kindersitz am Beifahrersitz montiert ist.	Priorität
Todesgefahr Airbag, Stuttgart, Di. 13.10.2004	Deaktivierung des Airbags, falls ein Kindersitz am Beifahrersitz montiert ist.	Hoch
Anruf mit Folgen, Ulm, Mi 06.02.2005	Durch zusätzlich angebrachte technische Geräte, sollen die Airbagsensoren nicht beschädigt werden oder besser isoliert werden.	Hoch
Sensordefekt aufgrund von Vereisung, Vaihingen, So. 03.01.2005	Bei niedrigen Temperaturen kann es dazu kommen, dass die Airbag-Sensoren einfrieren und nicht mehr funktionsfähig sind. Die Airbag-Sensoren werden durch eine Heizspule geheizt.	Hoch
Hoher Sachschaden bei kleinem Auffahrunfall, Augsburg, Do. 05.05.2005	Deaktivierung des Airbags, falls sich kein Objekt am Beifahrersitz befindet.	mittel

7.8 Aufwände der Gruppen je Iteration

Aufwände für die Iteration 1:

Gruppe	V	K	F	EPP	ÜPP	ÄAD	EKD	ÜKD	ES	ÜS	ÄEM	ÜEM	E	T	F	ÜT	F	ESC	ÜSC	S
1-1-1	120	0	135	0	210	0	300	0	210	0	30	0	0	0	180	120	360	90	60	
1-1-2	90	30	30	40	0	30	0	10	0	20	0	0	0	30	0	30	180	0	0	
1-1-3	150	100	80	60	0	90	210	90	0	120	0	0	0	180	60	50	480	60	0	
1-2-1	60	30	40	90	10	90	10	210	0	60	0	0	0	180	20	60	180	30	0	
1-2-2	70	600	40	120	60	120	0	30	20	60	0	0	0	60	20	85	240	30	120	
1-2-3	100	20	30	240	0	360	10	60	0	60	20	0	0	240	0	30	480	60	10	
2-1-1	315	130	25	0	0	80	80	30	10	30	40	100	220	30	40	120	390	60	190	
2-1-2	240	120	30	0	0	150	0	120	60	80	0	120	0	120	0	0	0	0	0	
2-1-3	45	145	20	0	0	160	130	80	30	70	20	140	60	0	80	40	160	160	0	
2-2-1	300	180	105	0	0	180	0	120	0	0	120	120	135	120	0	0	480	120	300	
2-2-2	165	160	70	0	0	30	5	30	15	60	15	60	20	50	0	600	240	0	310	

= Türschloss
 = Airbag
 = Spurassistenten-System

Aufwände für die Iteration 2:

Gruppe	V	K	F	EPP	ÜPP	ÄAD	EKD	ÜKD	AKD	ES	ÜS	ÄEM	ÜEM	E	T	F	ÜT	F	ESC	ÜSC	DEA	S	DQS			
1-1-1	120	180	40	150	60	90	120	60	0	150	50	0	60	0	0	0	90	0	15	720	0	30	120	30	60	
1-1-2	40	30	0	30	5	0	60	0	0	30	0	0	0	0	0	0	0	0	0	0	0	40	0	60		
1-1-3	60	0	30	30	0	0	135	90	90	20	20	0	180	0	0	0	210	0	30	90	270	60	180	240	240	
1-2-1	140	70	80	90	0	20	90	0	20	30	0	10	20	0	0	0	120	0	30	60	240	30	40	0	60	
1-2-2	0	150	270	220	30	0	120	60	100	90	0	0	0	0	0	0	3940	0	40	420	240	540	0	60	300	
1-2-3	90	140	10	180	0	30	180	80	40	60	140	10	20	80	0	0	180	0	0	60	360	0	60	50	60	
2-1-1	80	50	70	0	0	0	100	90	0	60	30	0	70	60	0	160	90	0	90	60	0	120	60	390	60	
2-1-2	290	541	200	0	0	0	75	30	0	30	15	0	60	0	0	122	70	0	60	360	120	60	90	30	120	
2-1-3	30	80	50	0	0	0	120	60	0	80	10	40	90	40	120	0	50	180	20	60	60	120	0	60	480	
2-2-1	135	120	50	0	0	0	60	20	0	40	15	0	60	0	180	0	0	15	0	15	0	180	0	60	0	60
2-2-2	65	55	13	0	0	0	0	10	0	0	15	0	0	30	0	0	60	20	0	90	0	0	70	30	90	

Aufwände für die Iteration 3:

Gruppe	V	K	F	EPP	ÜPP	ÄAD	EKD	ÜKD	AKD	ES	ÜS	ÄEM	ÜEM	E	T	F	ÜT	F	ESC	ÜSC	DEA	S	DQS		
1-1-1	30	180	0	0	0	60	90	0	0	180	0	0	20	0	0	0	240	0	0	360	0	60	60	120	
1-1-2	60	150	45	40	20	0	40	20	0	30	30	0	30	0	0	0	150	0	60	610	0	40	30	60	
1-1-3	60	0	60	60	0	180	0	0	120	0	0	0	120	0	0	0	240	0	30	150	630	60	120	300	180
1-2-1	120	40	0	60	10	0	30	20	0	40	0	0	20	0	0	0	0	0	0	300	0	60	0	60	
1-2-2	0	120	160	55	5	20	5	30	30	5	0	15	0	0	0	0	60	5	0	50	500	0	0	330	
1-2-3	40	150	220	120	30	0	100	0	0	150	20	0	60	0	0	0	240	40	0	120	480	240	40	120	100
2-1-1	60	20	0	0	0	0	80	0	0	80	0	0	40	0	0	10	90	0	140	120	0	210	0	100	30
2-1-2	180	250	80	0	0	25	0	20	0	60	0	80	0	80	0	120	0	0	1080	60	0	30	0	0	0
2-1-3	20	170	56	0	0	90	0	0	210	0	0	90	0	105	0	80	0	10	50	360	0	120	0	180	
2-2-1	300	60	50	0	0	90	0	0	30	40	0	60	0	90	0	80	0	0	60	300	0	60	60	0	0
2-2-2	0	125	0	0	0	20	0	0	30	0	0	0	0	0	0	60	0	0	90	210	60	20	30	0	0

Kürzel	Beschreibung
V	Aufwand für Verständnis (z.B. Lesen der Anforderungen, Überlegungen)
K	Aufwand für Kommunikation und (Termin-)Absprachen innerhalb der Gruppe
F	Aufwand für Nachfragen bei den Betreuern
EPP	Aufwand zur Erstellung des Projektprotokolls
ÜPP	Aufwand für die Überarbeitung des Projektprotokolls aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄPP	Aufwand für die Änderung/Anpassung von Teilen eines fremden Projektprotokolls aufgrund der neuen Anforderungen
EAD	Aufwand, um das Anforderungsdokument auszufüllen
ÜAD	Aufwand für die Überarbeitung des Anforderungsdokuments aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄAD	Aufwand für die Änderung/Anpassung von Teilen eines fremden Anforderungsdokuments aufgrund geänderter Anforderungen
EKD	Aufwand für die Modellierung eines neuen Kontextdiagramms (ausdenken und zeichnen)
ÜKD	Aufwand für die Überarbeitung des Kontextdiagramms aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄKD	Aufwand für die Änderung/Anpassung eines fremden Kontextdiagramms aufgrund der neuen Anforderungen
ES	Aufwand für die Erstellung neuer Szenarien (ausdenken und aufschreiben)
ÜS	Aufwand für die Überarbeitung der Szenarien aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄS	Aufwand für die Änderung/Anpassung von fremden Szenarien aufgrund geänderter Anforderungen
EEM	Aufwand für die Erstellung eines Entscheidungsmodells (ausdenken und zeichnen)
ÜEM	Aufwand für die Überarbeitung eines Entscheidungsmodells aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄEM	Aufwand für die Änderung/Anpassung von Teilen eines fremden Entscheidungsmodells aufgrund geänderter Anforderungen
ETF	Aufwand für die Erstellung der Testfälle (ausdenken und aufschreiben)
ÜTF	Aufwand für die Überarbeitung der Testfälle aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄTF	Aufwand für die Änderung/Anpassung von Teilen der fremden Testfälle aufgrund der neuen Anforderungen
ESC	Aufwand für die Modellierung der Statecharts (ausdenken und zeichnen)
ÜSC	Aufwand für die Überarbeitung der Statecharts aufgrund der Hinweise der Betreuer, Fehler u. ä.
ÄSC	Aufwand für die Änderung/Anpassung von Teilen der fremden Statecharts aufgrund geänderter Anforderungen
DEA	Aufwand für die Einarbeitung in die neuen (fremden) Dokumente
DQS	Aufwand für die Durchführung der Qualitätssicherung der fremden Modelle gemäß vorgegebenen Template (nicht Testfälle testen!)
S	sonstiger Aufwand, der nicht durch die anderen Kategorien abgedeckt wird

Tabelle 10: Überblick über die eingesetzten Kategorien bei der Aufwandserfassung.

7.9 Auswertung der Szenarien der Gruppen je Iteration

Gruppe	Iteration	System	Anzahl echter Szenarien (nicht durch bisherige Anforderungen gefordert)	davon Statemate- relevant	davon unreali- stische Szena- rien	davon in einer der folgen- den Iterationen umzusetzen- de Anforderung
1-1-1	1	Türschloss	2	2	0	1
	2	Spurassistentz-System	-	-	-	-
	3	Airbag	-	-	-	-
1-1-2	1	Airbag	4	4	0	3
	2	Türschloss	3	3	0	2
	3	Spurassistentz-System	3	3	0	-
1-1-3	1	Spurassistentz-System	3	1	0	0
	2	Airbag	6	3	0	0
	3	Türschloss	3	1	0	-
1-2-1	1	Türschloss	3	1	0	0
	2	Spurassistentz-System	3	2	0	0
	3	Airbag	3	2	0	-
1-2-2	1	Airbag	3	2	1	1
	2	Türschloss	4	2	0	1
	3	Spurassistentz-System	2	2	0	-
1-2-3	1	Spurassistentz-System	3	2	1	0
	2	Airbag	3	2	1	1
	3	Türschloss	1	1	0	-
2-1-1	1	Türschloss	1	0	0	0
	2	Spurassistentz-System	2	2	0	2
	3	Airbag	3	2	0	-
2-1-2	1	Airbag	3	1	2	1
	2	Türschloss	3	2	0	2
	3	Spurassistentz-System	2	1	1	-
2-1-3	1	Spurassistentz-System	4	3	0	1
	2	Airbag	4	3	0	2
	3	Türschloss	4	2	0	-
2-2-1	1	Türschloss	2	2	0	0
	2	Spurassistentz-System	1	1	0	1
	3	Airbag	1	0	0	-
2-2-2	1	Airbag	4	3	1	1
	2	Türschloss	4	2	0	1
	3	Spurassistentz-System	2	2	0	-

7.10 Fragebögen zur qualitativen Auswertung

7.10.1 1. Iteration, 1. Woche

Fragen für die qualitative Auswertung vom 11.5.05

Antworten der PG

Antworten der EG

Verständnisfragen

1. zum Vorgehen

- Wochenplanung unklar \Rightarrow wird noch verbessert
- Szenarien bzw. Szenarienprotokoll unklar

2. zu den Produkten

3. zu den Templates

- Beispiel in den Vorlagen war unklar; bessere Erklärung und Beschreibung ist notwendig
- Erklärung und Folien reichen nicht aus, wenn man es selber machen soll

Reaktion:

Emotionale Fragen

1. Was hat euch geärgert?

- Aufgabenverteilung ist nicht klar; Was sollen die „Rollen“ darstellen?
- Was nutzen die 3 Stunden am Mittwoch? Die Zeit kann man auch anders nutzen
- Beispiele zu abstrakt, sie müssten detaillierter ausgearbeitet werden!
- Wir wissen nicht genau, was wir tun sollen
- Verwirrung bei der Wochenplanung
- Konsistentes Beispiel während der Einführungsphase (Roter Faden) notwendig gewesen

(a) Kontextdiagramme

- Folien sind nicht aussagekräftig \Rightarrow keiner weiß, wie man sie richtig ausfüllt
- Verwirrend. Der Nutzen der Kontextdiagramme ist nicht klar geworden

(b) Anforderungsdokumente

(c) Aufgabenstellung

- Die Aufgabenstellung ist nicht klar. Was soll z.B. die Zeitschiene im Projektprotokoll bedeuten?
- Auflistung der Aufgaben war hilfreich

(d) Zeitaufwand

- keine Beschwerden
- Es ist nur möglich, den Gesamtaufwand anzugeben — feingranularere Aussagen sind schwer zu machen

(e) Projektprotokolle

- Zu was dient das Projektprotokoll? Nutzen ist unklar.

(f) Vorlagen

2. Was hat euch Spaß gemacht?

- keine Aussage
- Bei Mikrowelle machte es Spaß, weil sie vorgegeben war
- Hier ist der Schritt, die Anforderungen in Statemate umzusetzen

3. Was hat gut funktioniert, was nicht?

- Problem mit Powerpoint — Powerpoint eignet sich nicht zur Darstellung der Entscheidungsmodelle, vor allem, wenn man viel daran ändern möchte
- Kontextdiagramme verstehen wir jetzt besser und können sie damit besser anwenden.
- Wir müssen zu viel lesen.

4. Erfolgserlebnisse?

5. Motivationssteigerung/-senkung?

- keine Aussage
- Bis man mit dem Modell angefangen hat, ging es schleppend; danach motivierter

Reflektion:

1. Wo gibt es eine Diskrepanz zwischen den eigenen Vorstellungen/Erwartungen und den gemachten Erfahrungen?

- Anforderungsdokument: Kein einfaches Runterschreiben, sondern Skizze notwendig; Probleme mit dem Anforderungsüberblick \Rightarrow grobes Modell für den Ablauf als Orientierung notwendig
- Zwischen Aufgabenstellung und Durchführung gibt es immer Diskrepanz! \rightarrow Modellieren anfangs schwieriger als vorgestellt

2. Was sind eurer Meinung/eures Verständnisses nach die Ziele des Experiments?

- Nutzen im späteren Arbeitsleben: Weiterarbeiten an unbekanntem Themen mit Spezifikation/Modell als Grundlage
- Vergleichen, welche der beiden Vorgehensweisen besser sind, um aus Vergangenheit zu lernen
- Lohnt sich der Aufwand in die Erstellung und das Verständnis der Entscheidungsmodelle

3. Gibt es weitere Ziele des Experiments, als die von uns vorgestellten, aufgrund der gewählten Methodik, Beispiele, ...?

- keine Aussage
- Neue Idee, Ziele, Features für Innovationen
- Neue Methoden zur kreativen Handhabung von Anforderungen

4. Ergibt sich euch der Nutzen der geforderten Aufgaben? Dienen die Aufgaben den oben vorgestellten Zielen?

- keine Aussage
- Entwicklungsprozess vollständig mitmachen
- Dokumentation der Prozesse interessant; Überblick, was alles zu tun ist

5. Welche Verbesserungsvorschläge bzgl. Methodik, Templates, Ablauf allgemein?

- keine Aussage
- Klärung von Fragen, die nach dem ersten Termin auftreten

Allgemein:

- Wochenplanungsfolien sollen aktualisiert werden und dann ins Netz gestellt werden

7.10.2 1. Iteration, 2. Woche

Fragen für die qualitative Auswertung vom 18.5.05

Antworten der PG

Antworten der EG

Verständnisfragen

1. zum Vorgehen

- Reihenfolge: Testfälle Implementierung (Statemate); wieso erst Testfälle? Unklar ist, dass Testfälle die Funktionalität spezifizieren, die in Anforderungen beschrieben werden

2. zu den Testfällen

Reaktion:

Emotionale Fragen

1. Was hat euch geärgert?

(a) Statemate allgemein

- Grafik-Bug: Panel kann man nicht richtig gut testen, da es einen grafischen Bug gibt
- In der Hardware sind noch Bugs drin: Crash der Airbags (Reparieren tut nicht so wie es soll)
- Statemate-Semantik ist verwirrend; schlechte Fehlermeldung
- Statemate-Modelle anschauen und Feedback geben

(b) Zeitaufwand

- Aufwandserfassung nicht voll funktionsfähig
- Zeitkategorien unklar
- Kommunikation; was lohnt sich aufgenommen zu werden? Welche Aufwände fallen unter sonstiges?

(c) StateMate Hardwarepanel

- Panel hilfreich für Testfallermittlung

(d) Vorlagen

2. Was hat euch Spaß gemacht?

- Woche war besser als die vorherige
- StateMate geht schneller voran als vorherige Aufgaben
- StateMate

3. Was hat gut funktioniert, was nicht?

- Besseres Verständnis insgesamt

4. Erfolgserlebnisse?

- Geht schneller voran; nicht so schleppend wie Analyse

5. Motivationssteigerung/-senkung?

- Praktische Modellierung macht mehr Spaß
- Powerpoint ist frustrierend

Reflektion:

1. Seid ihr mit der Vorgehensweise einverstanden oder fehlen euch noch wichtige Schritte, welche sind überflüssig?

- Vorgehensweise ist ok
- Keiner hat einen Verbesserungsvorschlag
- Was man bei Anforderungen machen musste, war nicht notwendig; andere waren nicht dieser Meinung

2. Entspricht euer Zeitplan noch der Realität?

- Zeitplan ist noch gut eingehalten
- Gute Einschätzung der Aufwände
- Weit drunter
- Der Zeitplan wird meist nicht verstanden (Sinn)

3. Warum ist der Umfang des Projekts angemessen oder nicht?

- Viel zu wenig Aufwand / keine Aussage
- Unterschiedliche Meinungen: zu wenig/zu viel; resultiert aus unterschiedlichen Projekten
- Statemate kommt zu kurz, weil viel Aufwand in die Erstellung der Anforderungen fließt; verteilt sich auf andere Arbeiten

4. Welche Verbesserungsvorschläge bzgl. Methodik, Templates, Ablauf allgemein?

7.10.3 1. Iteration, 3. Woche

Fragen für die qualitative Auswertung vom 25.5.05

Antworten der PG

Antworten der EG

Allgemeines zur ersten Iteration

1. Gut/Erfolgsenerlebnis

- zum Einarbeiten der Beschreibbarkeit war das Projektprotokoll und das Anforderungsdokument hilfreich

- Motivation ist vorhanden

2. Schlecht

- furchtbare Aufwandserfassung
- viel Schreibarbeit
 - Viel Schreibarbeit; aber jetzt besser

- Motivation ist vorhanden

Verständnisfragen zum Entwicklungsprozess

- Szenarien mit Auswirkungsänderungen wären im Kopf klar. Aber nicht dokumentiert

Vorschläge zur Verbesserung des Entwicklungsprozesses

- Je eine Person hat die Hälfte der Testfälle erstellt. Vier-Augen-Prinzip als Ergänzung
- Grundsätzlich das Vier-Augen-Prinzip weil man sich dann besser ergänzen kann
- Wenn man etwas getrennt macht, dann werden Fehler später entdeckt ⇒ Mehraufwand
- Szenarien, Entscheidungsmodelle waren wichtig → gut um sich Gedanken zu machen
- Austausch zwischen Gruppen nicht mit E-Mail zu empfehlen, da so wichtige Hinweise verloren gehen (im Gegensatz zu einer persönlichen Übergabe)

Vorschläge zum Ablauf (Verbesserungen im Experiment)

Vorschläge zum Ablauf der nächsten Iteration

7.10.4 2. Iteration, 1. Woche

Fragen für die qualitative Auswertung vom 1.6.05

Antworten der PG

Antworten der EG

Verständnisfragen

1. zu den Anforderungen

- unklar, wie die Pressemitteilungen zu verstehen sind

2. zu der Qualitätssicherung

- Template hilfreich

Reaktion:

Emotionale Fragen

1. Was hat euch geärgert?

(a) Kontextdiagramme

(b) Anforderungsdokumente

- Abgleich der „alten“ Dokumente: sollen diese verbessert werden?
- Entwurfsentscheidungen sollen die Szenarien u.U. enthalten (Verweise)

(c) Aufgabenstellung

(d) Zeitaufwand

(e) Projektprotokolle

(f) Tausch

2. Was hat euch Spaß gemacht?

3. Was hat gut funktioniert, was nicht?

4. Erfolgserlebnisse?

- Die Erstellung von Testfällen: Fragestellungen, die zum Teil beantwortet wurden

5. Motivationssteigerung/-senkung?

Reflektion:

1. Was hat euch die Einarbeitung in die Dokumente der anderen Gruppe für die eigene Arbeit gebracht?

- Beispiel einer anderen Gruppe betrachten
- Detailtiefe hat sich stark unterschieden
- Herangehensweise war stark unterschiedlich
- Vor allem Kennenlernen des anderen Systems

2. Einschätzung des eigenen Lernerfolgs/Lernkurve?

- Kontextdiagramm war im Nachhinein besser verständlich
- Statemate/Anforderungen war verständlich

3. Nutzen des Qualitätssicherungscheckdokuments?

- Fehlfunktionen fehlen
- Zu allgemeines Dokument (ja/nein-Antworten)
- Bessere Beschreibung bei Fehlern und Auffälligkeiten
- Überarbeitungszeit notwendig (gefundene Fehler beheben); sonst ist das Dokument nutzlos
- Kontrolle war hier hilfreich
- Anderer Blickwinkel
- Hätte zur Sicherung der eigenen Dokumente nur bedingt geholfen → da Feinheiten nur beachtet werden

4. Welche Verbesserungsvorschläge bzgl. Methodik, Templates, Ablauf allgemein?

- Bei der Übergabe ein Treffen zur Einarbeitung
- Beim Tausch ein gemeinsames Treffen nach dem Qualitätscheck

7.10.5 2. Iteration, 2. Woche

Fragen für die qualitative Auswertung vom 8.6.05

Antworten der PG

Antworten der EG

Verständnisfragen

1. zum Vorgehen

2. zu den Testfällen
 - Testfälle bei einer Gruppe zu ausführlich; sie haben schon 35 Seiten und da ist noch kein Ende in Sicht

Reaktion:

Emotionale Fragen

1. Was hat euch geärgert?
 - Im Wesentlichen ist alles soweit in Ordnung.
 - kleine Meinungsverschiedenheiten über die Gruppengrenzen hinweg; die Türschlossgruppe (35 Seiten) hat das Projekt vollkommen überarbeiten müssen und sagte das auch; die andere Gruppe hat sich angegriffen gefühlt

- (a) Testfalltemplate
 - Schwammige System-Beschreibungen führen zu Verwirrungen und Unstimmigkeiten

- (b) Testfallerstellung
 - Testfallerstellung zum Teil schwierig; welche Fälle sind zu vernachlässigen, welche nicht

- (c) Statemate allgemein

(d) Zeitaufwand

(e) Statestate Hardwarepanel

- Hat soweit gut getan (bis auf den Grafikbug)

(f) Vorlagen

2. Was hat euch Spaß gemacht?

- Statestatemodellierung war soweit problemlos

3. Was hat gut funktioniert, was nicht?

4. Erfolgserlebnisse?

5. Motivationssteigerung/-senkung?

Reflektion:

1. Entspricht euer Zeitplan noch der Realität?

- Geplante Zeit und die eigentliche weichen ab: zu viel Bearbeitung der alten Aufgabe. Dadurch zusätzlicher, nicht eingeplanter Aufwand.

2. Warum ist der Umfang des Projekts angemessen oder nicht?

- Soweit ist die Stimmung gleich: der Aufwand hält sich im Rahmen und ist sehr fair
- Nicht viel

3. Welche Verbesserungsvorschläge bzgl. Methodik, Templates, Ablauf allgemein?

7.10.6 2. Iteration, 3. Woche

Fragen für die qualitative Auswertung vom 15.6.05

Antworten der PG

Antworten der EG

Verständnisfragen

1. zum Vorgehen
2. zu Statemate

Reaktion:

Emotionale Fragen

1. Was hat euch geärgert? (Statemate, Hardwarepanel, Zeitaufwand, Testen)

- zu wenig zu implementieren
- Statemate: richtiges Panel mit richtigem Projekt. Weil Panel vorhanden war

2. Was hat euch Spaß gemacht?

- solange es klappt macht es auch Spaß
- Modellierung zu überlegen (Alternativen klären)

3. Motivationssteigerung/-senkung?

- war meist in Ordnung
- Umsetzung schnell aber nicht einfach
- Übernahme des Modells geht
- 2. Iteration hat wesentlich mehr Spaß gemacht
- Weniger Frustration: man wusste was zu tun ist
- Weiterarbeiten macht mehr Spaß

Reflektion:

1. Ist die Zustandsmodellierung mittlerweile klarer?

- Aspekte von StateMate sind jetzt klarer
- Umsetzung in Zustände wird langsam besser begriffen
- Unterschiedliche Lösungsmengen: Probleme mit der Parallelität, weil schwer zu Synchronisieren → besser: in sich abgeschlossene Module in StateMate.
- Modellkomponenten erstellen und vom Controller trennen
- StateMatemodell war einfach zu erweitern, kompakt (einmal nur ein Zustand erweitern, einmal nur Pfeile)
- Am Anfang war zu klären, welche Pfeile welche Funktionalität umfassen → dann Unterzustände ergänzen

2. Einschätzung der eigenen Lernkurve?

(StateMate, Anforderungen, Spezifikationen, Testfälle, Entscheidungsmodell, Szenarien, Projektplan, Zeitaufwand)

- Mikrowelle war höheres Lernergebnis, aber dafür auch mehr Aufwand
- Zustandsautomaten haben sich jetzt besser in Gedächtnis abgesetzt
- am wichtigsten: StateMate und Zusammenspiel mit dem Panel, sowie bekannte Signale

(a) Was habt ihr bisher gelernt?

(b) Was ist euch immer noch unklar?

- in ein vorhandenes Entscheidungsmodell neue Dinge einbringen ist schwierig

7.10.7 3. Iteration, 1. Woche

Fragen für die qualitative Auswertung vom 22.6.05

Antworten der PG

Antworten der EG

Reaktion:

1. Was ist euch aufgefallen?

- Ihnen fallen keine mehr Szenarien ein
- Szenarien wurden zum Teil lustig
- Panel und Spezifikation kamen (zu) spät

Reflektion:

1. Arbeitsaufteilung (wie habt ihr die Aufgaben aufgeteilt)?

- Aufteilung lohnt sich nicht, da der Umfang der Projekte zu klein ist
- Andere hatten sich zuerst aufgeteilt, dann die Dokumente ausgetauscht und schließlich wieder gemischt
- gemeinsam ist man effektiver \Rightarrow daher von Anfang an gemeinsam gearbeitet
- eine andere Gruppe hat sich die Aufgaben immer aufgeteilt
- gerade die Entwicklung von Entscheidungsmodellen war gemeinsam besser

2. Arbeitsaufteilung im Verlauf (Historie; hat sich die Aufteilung geändert)?

- 1. Iteration anders, danach gleich

3. Einschätzung der Lernkurve der anderen

- eine längere Einführungsphase wäre besser
- Die Statematamodelle wurden besser
- Sowohl Szenarien als auch Anforderungen und EM-Modelle nicht verbessert (eher individuelle Unterschiede zw. den Gruppen)
- einige Dateiformate für Erweiterungen ungünstig (z.B. PDF)

4. Einarbeitung in die anderen Aufgaben (drei unterschiedliche Ansätze)?

- Türschloss ist komplizierter als die anderen beiden Systeme

5. Kommunikation zwischen den Tauschgruppen (Support)

- gute Unterstützung
- Nachfrage war nicht nötig
- Per E-Mail, allerdings war Nachfrage nicht nötig

6. Einschätzung des kommenden Aufwands? (mehr oder weniger als die letzten Aufgaben?)

- Türschloss mehr
- Türschloss recht aufwändig
- Rest weniger

7. Nur EM-Gruppen: Sind die Hinweisschilder sinnvoll wählbar unabhängig von den technischen Design Optionen?

7.10.8 3. Iteration, 2. Woche

Fragen für die qualitative Auswertung vom 29.6.05

Antworten der PG

Antworten der EG

Verständnisfragen

1. zum Vorgehen
2. zu den Testfällen

Reaktion:

Emotionale Fragen

1. Was hat euch geärgert?

- (a) Testfallerstellung
- (b) Statemate allgemein

- Positiv: Wiederverwendung von Anteilen aus der selbsterstellten 2. Iteration: in der dritten Iteration ein Controller, der drei Zustände abfragt

- (c) Zeitaufwand

- (d) Statemate Hardwarepanel

- Anforderung der Heizspulen war verwirrend
- Probleme mit dem Panel bzgl. des Zündung-Buttons

2. Was hat euch Spaß gemacht?

3. Was hat gut funktioniert, was nicht?

- Import hat bei einigen Probleme gemacht

4. Erfolgserlebnisse?

5. Motivationssteigerung/-senkung?

- Es geht dem Ende zu \Rightarrow also Steigerung

Reflektion:

1. Was habt ihr alles gelernt im Praktikum? (Werkzeug, Prozess: Anforderungen, Entwurf, Implementierung, Modellierung)

- Umgang mit StateMate
- Zustandsorientiertes Denken
- Probleme bei der Abnahme Projekte anderer
- Probleme bei der Anforderungsumsetzung
- Zustandsorientierte Entwicklung bisher theoretisch; jetzt Bezug zur Praxis
- Hineindenken in andere Systeme: Dabei auf Konsistenz achten; auch bei Eigen-System-Entwicklung darauf achten, was für Dokumente und Informationen die anderen brauchen
- Wie werden Anforderungen analysiert und festgehalten

2. Ist die Planung eines Software-Projekts klar geworden?

- Zum Teil; schwierig bei diesem kleinen Umfang

3. Wie ist die Eigeneinschätzung bzgl. StateMatekönnen?

- mittel: Glauben damit umgehen zu können, aber nur in ihrem „beschränkten“ Bereich
- Im Praktikum Vorgehen in größeren Projekten kennen gelernt; ansonsten eher kleinere Projekte bekannt; Notwendigkeit der Qualitätssicherung und Dokumentation

4. Nur EM-Gruppen: Wird das Weiterarbeiten mit den Diagrammen/Dokumenten komplizierter?

7.10.9 3. Iteration, 3. Woche

Fragen für die qualitative Auswertung vom 6.7.05

Antworten der PG

Antworten der EG

Allgemeine Fragen über das Vorgehen über das Praktikum hinweg

1. Mit welchen Schwierigkeiten hattet ihr bei den verschiedenen Iterationen zu tun?
 - Import hat bei einigen Probleme gemacht
2. Wie sieht es mit der Motivation über die Iterationen hinweg aus?
3. Gab es ein Projekt, das besonders interessant war?
 - SAS: war verspielt und war daher interessant
 - Türe: war eine Herausforderung, komplexere Steuerung, Einwirken vieler Teil-Systeme
4. Gab es ein Projekt, das euch nicht gefallen hat? Wenn ja, warum?
 - Türschloss: zu komplex; Aufwand war viel höher
 - Airbag: war einfach; geringer Umfang, klare Vorgaben
5. Gibt es offene Frage bezüglich des Sinns der Vorgehensweise (Wieso dieser Ablauf für uns interessant war)? Oder habt ihr einfach nur getan was man euch geheißen hat?
 - Wie kommen die Betreuer nun zu Ergebnissen? (EG-PG-Auswertung → wie vergleichen?)
6. Wo lagen die Einschränkungen bei der Entwicklung der Innovationen (Templates ungenau, Panel defekt, Werkzeug zu schwer, Vorgehensweise nicht geeignet, ...)?
7. Welche Probleme tauchten bei der Weiterentwicklung der Innovationen auf?

Liste der bisher erschienenen Ulmer Informatik-Berichte
Einige davon sind per FTP von `ftp.informatik.uni-ulm.de` erhältlich
Die mit * markierten Berichte sind vergriffen

List of technical reports published by the University of Ulm
Some of them are available by FTP from `ftp.informatik.uni-ulm.de`
Reports marked with * are out of print

- 91-01 *Ker-I Ko, P. Orponen, U. Schöning, O. Watanabe*
Instance Complexity
- 91-02* *K. Gladitz, H. Fassbender, H. Vogler*
Compiler-Based Implementation of Syntax-Directed Functional Programming
- 91-03* *Alfons Geser*
Relative Termination
- 91-04* *J. Köbler, U. Schöning, J. Toran*
Graph Isomorphism is low for PP
- 91-05 *Johannes Köbler, Thomas Thierauf*
Complexity Restricted Advice Functions
- 91-06* *Uwe Schöning*
Recent Highlights in Structural Complexity Theory
- 91-07* *F. Green, J. Köbler, J. Toran*
The Power of Middle Bit
- 91-08* *V.Arvind, Y. Han, L. Hamachandra, J. Köbler, A. Lozano, M. Mundhenk, A. Ogiwara,*
U. Schöning, R. Silvestri, T. Thierauf
Reductions for Sets of Low Information Content
- 92-01* *Vikraman Arvind, Johannes Köbler, Martin Mundhenk*
On Bounded Truth-Table and Conjunctive Reductions to Sparse and Tally Sets
- 92-02* *Thomas Noll, Heiko Vogler*
Top-down Parsing with Simultaneous Evaluation of Noncircular Attribute Grammars
- 92-03 *Fakultät für Informatik*
17. Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen
- 92-04* *V. Arvind, J. Köbler, M. Mundhenk*
Lowness and the Complexity of Sparse and Tally Descriptions
- 92-05* *Johannes Köbler*
Locating P/poly Optimally in the Extended Low Hierarchy
- 92-06* *Armin Kühnemann, Heiko Vogler*
Synthesized and inherited functions -a new computational model for syntax-directed semantics
- 92-07* *Heinz Fassbender, Heiko Vogler*
A Universal Unification Algorithm Based on Unification-Driven Leftmost Outermost Narrowing

- 92-08* *Uwe Schöning*
On Random Reductions from Sparse Sets to Tally Sets
- 92-09* *Hermann von Hasseln, Laura Martignon*
Consistency in Stochastic Network
- 92-10 *Michael Schmitt*
A Slightly Improved Upper Bound on the Size of Weights Sufficient to Represent Any Linearly Separable Boolean Function
- 92-11 *Johannes Köbler, Seinosuke Toda*
On the Power of Generalized MOD-Classes
- 92-12 *V. Arvind, J. Köbler, M. Mundhenk*
Reliable Reductions, High Sets and Low Sets
- 92-13 *Alfons Geser*
On a monotonic semantic path ordering
- 92-14* *Joost Engelfriet, Heiko Vogler*
The Translation Power of Top-Down Tree-To-Graph Transducers
- 93-01 *Alfred Lupper, Konrad Froitzheim*
AppleTalk Link Access Protocol basierend auf dem Abstract Personal Communications Manager
- 93-02 *M.H. Scholl, C. Laasch, C. Rich, H.-J. Schek, M. Tresch*
The COCOON Object Model
- 93-03 *Thomas Thierauf, Seinosuke Toda, Osamu Watanabe*
On Sets Bounded Truth-Table Reducible to P-selective Sets
- 93-04 *Jin-Yi Cai, Frederic Green, Thomas Thierauf*
On the Correlation of Symmetric Functions
- 93-05 *K.Kuhn, M.Reichert, M. Nathe, T. Beuter, C. Heinlein, P. Dadam*
A Conceptual Approach to an Open Hospital Information System
- 93-06 *Klaus Gaßner*
Rechnerunterstützung für die konzeptuelle Modellierung
- 93-07 *Ullrich Keßler, Peter Dadam*
Towards Customizable, Flexible Storage Structures for Complex Objects
- 94-01 *Michael Schmitt*
On the Complexity of Consistency Problems for Neurons with Binary Weights
- 94-02 *Armin Kühnemann, Heiko Vogler*
A Pumping Lemma for Output Languages of Attributed Tree Transducers
- 94-03 *Harry Buhrman, Jim Kadin, Thomas Thierauf*
On Functions Computable with Nonadaptive Queries to NP
- 94-04 *Heinz Faßbender, Heiko Vogler, Andrea Wedel*
Implementation of a Deterministic Partial E-Unification Algorithm for Macro Tree Transducers

- 94-05 *V. Arvind, J. Köbler, R. Schuler*
On Helping and Interactive Proof Systems
- 94-06 *Christian Kalus, Peter Dadam*
Incorporating record subtyping into a relational data model
- 94-07 *Markus Tresch, Marc H. Scholl*
A Classification of Multi-Database Languages
- 94-08 *Friedrich von Henke, Harald Rueß*
Arbeitstreffen Typtheorie: Zusammenfassung der Beiträge
- 94-09 *F.W. von Henke, A. Dold, H. Rueß, D. Schwier, M. Strecker*
Construction and Deduction Methods for the Formal Development of Software
- 94-10 *Axel Dold*
Formalisierung schematischer Algorithmen
- 94-11 *Johannes Köbler, Osamu Watanabe*
New Collapse Consequences of NP Having Small Circuits
- 94-12 *Rainer Schuler*
On Average Polynomial Time
- 94-13 *Rainer Schuler, Osamu Watanabe*
Towards Average-Case Complexity Analysis of NP Optimization Problems
- 94-14 *Wolfram Schulte, Ton Vullings*
Linking Reactive Software to the X-Window System
- 94-15 *Alfred Lupper*
Namensverwaltung und Adressierung in Distributed Shared Memory-Systemen
- 94-16 *Robert Regn*
Verteilte Unix-Betriebssysteme
- 94-17 *Helmuth Partsch*
Again on Recognition and Parsing of Context-Free Grammars:
Two Exercises in Transformational Programming
- 94-18 *Helmuth Partsch*
Transformational Development of Data-Parallel Algorithms: an Example
- 95-01 *Oleg Verbitsky*
On the Largest Common Subgraph Problem
- 95-02 *Uwe Schöning*
Complexity of Presburger Arithmetic with Fixed Quantifier Dimension
- 95-03 *Harry Buhrman, Thomas Thierauf*
The Complexity of Generating and Checking Proofs of Membership
- 95-04 *Rainer Schuler, Tomoyuki Yamakami*
Structural Average Case Complexity
- 95-05 *Klaus Achatz, Wolfram Schulte*
Architecture Independent Massive Parallelization of Divide-And-Conquer Algorithms

- 95-06 *Christoph Karg, Rainer Schuler*
Structure in Average Case Complexity
- 95-07 *P. Dadam, K. Kuhn, M. Reichert, T. Beuter, M. Nathe*
ADEPT: Ein integrierender Ansatz zur Entwicklung flexibler, zuverlässiger kooperierender Assistenzsysteme in klinischen Anwendungsumgebungen
- 95-08 *Jürgen Kehrer, Peter Schulthess*
Aufbereitung von gescannten Röntgenbildern zur filmlosen Diagnostik
- 95-09 *Hans-Jörg Burtschick, Wolfgang Lindner*
On Sets Turing Reducible to P-Selective Sets
- 95-10 *Boris Hartmann*
Berücksichtigung lokaler Randbedingung bei globaler Zielloptimierung mit neuronalen Netzen am Beispiel Truck Backer-Upper
- 95-12 *Klaus Achatz, Wolfram Schulte*
Massive Parallelization of Divide-and-Conquer Algorithms over Powerlists
- 95-13 *Andrea Mößle, Heiko Vogler*
Efficient Call-by-value Evaluation Strategy of Primitive Recursive Program Schemes
- 95-14 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
A Generic Specification for Verifying Peephole Optimizations
- 96-01 *Ercüment Canver, Jan-Tecker Gayen, Adam Moik*
Formale Entwicklung der Steuerungssoftware für eine elektrisch ortsbediente Weiche mit VSE
- 96-02 *Bernhard Nebel*
Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class
- 96-03 *Ton Vullingsh, Wolfram Schulte, Thilo Schwinn*
An Introduction to TkGofer
- 96-04 *Thomas Beuter, Peter Dadam*
Anwendungsspezifische Anforderungen an Workflow-Management-Systeme am Beispiel der Domäne Concurrent-Engineering
- 96-05 *Gerhard Schellhorn, Wolfgang Ahrendt*
Verification of a Prolog Compiler - First Steps with KIV
- 96-06 *Manindra Agrawal, Thomas Thierauf*
Satisfiability Problems
- 96-07 *Vikraman Arvind, Jacobo Torán*
A nonadaptive NC Checker for Permutation Group Intersection
- 96-08 *David Cyrluk, Oliver Möller, Harald Rueß*
An Efficient Decision Procedure for a Theory of Fix-Sized Bitvectors with Composition and Extraction
- 96-09 *Bernd Biechele, Dietmar Ernst, Frank Houdek, Joachim Schmid, Wolfram Schulte*
Erfahrungen bei der Modellierung eingebetteter Systeme mit verschiedenen SA/RT-Ansätzen

- 96-10 *Falk Bartels, Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Formalizing Fixed-Point Theory in PVS
- 96-11 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Mechanized Semantics of Simple Imperative Programming Constructs
- 96-12 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Generic Compilation Schemes for Simple Programming Constructs
- 96-13 *Klaus Achatz, Helmuth Partsch*
From Descriptive Specifications to Operational ones: A Powerful Transformation Rule, its Applications and Variants
- 97-01 *Jochen Messner*
Pattern Matching in Trace Monoids
- 97-02 *Wolfgang Lindner, Rainer Schuler*
A Small Span Theorem within P
- 97-03 *Thomas Bauer, Peter Dadam*
A Distributed Execution Environment for Large-Scale Workflow Management Systems with Subnets and Server Migration
- 97-04 *Christian Heinlein, Peter Dadam*
Interaction Expressions - A Powerful Formalism for Describing Inter-Workflow Dependencies
- 97-05 *Vikraman Arvind, Johannes Köbler*
On Pseudorandomness and Resource-Bounded Measure
- 97-06 *Gerhard Partsch*
Punkt-zu-Punkt- und Mehrpunkt-basierende LAN-Integrationsstrategien für den digitalen Mobilfunkstandard DECT
- 97-07 *Manfred Reichert, Peter Dadam*
 $ADEPT_{flex}$ - Supporting Dynamic Changes of Workflows Without Loosing Control
- 97-08 *Hans Braxmeier, Dietmar Ernst, Andrea Mößle, Heiko Vogler*
The Project NoName - A functional programming language with its development environment
- 97-09 *Christian Heinlein*
Grundlagen von Interaktionsausdrücken
- 97-10 *Christian Heinlein*
Graphische Repräsentation von Interaktionsausdrücken
- 97-11 *Christian Heinlein*
Sprachtheoretische Semantik von Interaktionsausdrücken
- 97-12 *Gerhard Schellhorn, Wolfgang Reif*
Proving Properties of Finite Enumerations: A Problem Set for Automated Theorem Provers

- 97-13 *Dietmar Ernst, Frank Houdek, Wolfram Schulte, Thilo Schwinn*
Experimenteller Vergleich statischer und dynamischer Softwareprüfung für eingebettete Systeme
- 97-14 *Wolfgang Reif, Gerhard Schellhorn*
Theorem Proving in Large Theories
- 97-15 *Thomas Wennekers*
Asymptotik rekurrenter neuronaler Netze mit zufälligen Kopplungen
- 97-16 *Peter Dadam, Klaus Kuhn, Manfred Reichert*
Clinical Workflows - The Killer Application for Process-oriented Information Systems?
- 97-17 *Mohammad Ali Livani, Jörg Kaiser*
EDF Consensus on CAN Bus Access in Dynamic Real-Time Applications
- 97-18 *Johannes Köbler, Rainer Schuler*
Using Efficient Average-Case Algorithms to Collapse Worst-Case Complexity Classes
- 98-01 *Daniela Damm, Lutz Claes, Friedrich W. von Henke, Alexander Seitz, Adelinde Uhrmacher, Steffen Wolf*
Ein fallbasiertes System für die Interpretation von Literatur zur Knochenheilung
- 98-02 *Thomas Bauer, Peter Dadam*
Architekturen für skalierbare Workflow-Management-Systeme - Klassifikation und Analyse
- 98-03 *Marko Luther, Martin Strecker*
A guided tour through *Typelab*
- 98-04 *Heiko Neumann, Luiz Pessoa*
Visual Filling-in and Surface Property Reconstruction
- 98-05 *Ercüment Canver*
Formal Verification of a Coordinated Atomic Action Based Design
- 98-06 *Andreas Küchler*
On the Correspondence between Neural Folding Architectures and Tree Automata
- 98-07 *Heiko Neumann, Thorsten Hansen, Luiz Pessoa*
Interaction of ON and OFF Pathways for Visual Contrast Measurement
- 98-08 *Thomas Wennekers*
Synfire Graphs: From Spike Patterns to Automata of Spiking Neurons
- 98-09 *Thomas Bauer, Peter Dadam*
Variable Migration von Workflows in *ADEPT*
- 98-10 *Heiko Neumann, Wolfgang Sepp*
Recurrent V1 – V2 Interaction in Early Visual Boundary Processing
- 98-11 *Frank Houdek, Dietmar Ernst, Thilo Schwinn*
Prüfen von C-Code und Statmate/Matlab-Spezifikationen: Ein Experiment

- 98-12 *Gerhard Schellhorn*
Proving Properties of Directed Graphs: A Problem Set for Automated Theorem Provers
- 98-13 *Gerhard Schellhorn, Wolfgang Reif*
Theorems from Compiler Verification: A Problem Set for Automated Theorem Provers
- 98-14 *Mohammad Ali Livani*
SHARE: A Transparent Mechanism for Reliable Broadcast Delivery in CAN
- 98-15 *Mohammad Ali Livani, Jörg Kaiser*
Predictable Atomic Multicast in the Controller Area Network (CAN)
- 99-01 *Susanne Boll, Wolfgang Klas, Utz Westermann*
A Comparison of Multimedia Document Models Concerning Advanced Requirements
- 99-02 *Thomas Bauer, Peter Dadam*
Verteilungsmodelle für Workflow-Management-Systeme - Klassifikation und Simulation
- 99-03 *Uwe Schöning*
On the Complexity of Constraint Satisfaction
- 99-04 *Ercument Canver*
Model-Checking zur Analyse von Message Sequence Charts über Statecharts
- 99-05 *Johannes Köbler, Wolfgang Lindner, Rainer Schuler*
Derandomizing RP if Boolean Circuits are not Learnable
- 99-06 *Utz Westermann, Wolfgang Klas*
Architecture of a DataBlade Module for the Integrated Management of Multimedia Assets
- 99-07 *Peter Dadam, Manfred Reichert*
Enterprise-wide and Cross-enterprise Workflow Management: Concepts, Systems, Applications. Paderborn, Germany, October 6, 1999, GI-Workshop Proceedings, Informatik '99
- 99-08 *Vikraman Arvind, Johannes Köbler*
Graph Isomorphism is Low for ZPP^{NP} and other Lowness results
- 99-09 *Thomas Bauer, Peter Dadam*
Efficient Distributed Workflow Management Based on Variable Server Assignments
- 2000-02 *Thomas Bauer, Peter Dadam*
Variable Serverzuordnungen und komplexe Bearbeiterzuordnungen im Workflow-Management-System ADEPT
- 2000-03 *Gregory Baratoff, Christian Toepfer, Heiko Neumann*
Combined space-variant maps for optical flow based navigation
- 2000-04 *Wolfgang Gehring*
Ein Rahmenwerk zur Einführung von Leistungspunktsystemen
- 2000-05 *Susanne Boll, Christian Heinlein, Wolfgang Klas, Jochen Wandel*

- Intelligent Prefetching and Buffering for Interactive Streaming of MPEG Videos
- 2000-06 *Wolfgang Reif, Gerhard Schellhorn, Andreas Thums*
Fehlersuche in Formalen Spezifikationen
- 2000-07 *Gerhard Schellhorn, Wolfgang Reif (eds.)*
FM-Tools 2000: The 4th Workshop on Tools for System Design and Verification
- 2000-08 *Thomas Bauer, Manfred Reichert, Peter Dadam*
Effiziente Durchführung von Prozessmigrationen in verteilten Workflow-
Management-Systemen
- 2000-09 *Thomas Bauer, Peter Dadam*
Vermeidung von Überlastsituationen durch Replikation von Workflow-Servern in
ADEPT
- 2000-10 *Thomas Bauer, Manfred Reichert, Peter Dadam*
Adaptives und verteiltes Workflow-Management
- 2000-11 *Christian Heinlein*
Workflow and Process Synchronization with Interaction Expressions and Graphs
- 2001-01 *Hubert Hug, Rainer Schuler*
DNA-based parallel computation of simple arithmetic
- 2001-02 *Friedhelm Schwenker, Hans A. Kestler, Günther Palm*
3-D Visual Object Classification with Hierarchical Radial Basis Function Networks
- 2001-03 *Hans A. Kestler, Friedhelm Schwenker, Günther Palm*
RBF network classification of ECGs as a potential marker for sudden cardiac death
- 2001-04 *Christian Dietrich, Friedhelm Schwenker, Klaus Riede, Günther Palm*
Classification of Bioacoustic Time Series Utilizing Pulse Detection, Time and
Frequency Features and Data Fusion
- 2002-01 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
Effiziente Verträglichkeitsprüfung und automatische Migration von Workflow-
Instanzen bei der Evolution von Workflow-Schemata
- 2002-02 *Walter Guttmann*
Deriving an Applicative Heapsort Algorithm
- 2002-03 *Axel Dold, Friedrich W. von Henke, Vincent Vialard, Wolfgang Goerigk*
A Mechanically Verified Compiling Specification for a Realistic Compiler
- 2003-01 *Manfred Reichert, Stefanie Rinderle, Peter Dadam*
A Formal Framework for Workflow Type and Instance Changes Under Correctness
Checks
- 2003-02 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
Supporting Workflow Schema Evolution By Efficient Compliance Checks
- 2003-03 *Christian Heinlein*
Safely Extending Procedure Types to Allow Nested Procedures as Values
- 2003-04 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
On Dealing With Semantically Conflicting Business Process Changes.

- 2003-05 *Christian Heinlein*
Dynamic Class Methods in Java
- 2003-06 *Christian Heinlein*
Vertical, Horizontal, and Behavioural Extensibility of Software Systems
- 2003-07 *Christian Heinlein*
Safely Extending Procedure Types to Allow Nested Procedures as Values
(Corrected Version)
- 2003-08 *Changling Liu, Jörg Kaiser*
Survey of Mobile Ad Hoc Network Routing Protocols)
- 2004-01 *Thom Frühwirth, Marc Meister (eds.)*
First Workshop on Constraint Handling Rules
- 2004-02 *Christian Heinlein*
Concept and Implementation of C+++, an Extension of C++ to Support User-Defined
Operator Symbols and Control Structures
- 2004-03 *Susanne Biundo, Thom Frühwirth, Günther Palm(eds.)*
Poster Proceedings of the 27th Annual German Conference on Artificial Intelligence
- 2005-01 *Armin Wolf, Thom Frühwirth, Marc Meister (eds.)*
19th Workshop on (Constraint) Logic Programming
- 2005-02 *Wolfgang Lindner (Hg.), Universität Ulm , Christopher Wolf (Hg.) KU Leuven*
2. Krypto-Tag – Workshop über Kryptographie, Universität Ulm
- 2005-03 *Walter Guttmann, Markus Maucher*
Constrained Ordering
- 2006-01 *Stefan Sarstedt*
Model-Driven Development with ACTIVECHARTS, Tutorial
- 2006-02 *Alexander Raschke, Ramin Tavakoli Kolagari*
Ein experimenteller Vergleich zwischen einer plan-getriebenen und einer
leichtgewichtigen Entwicklungsmethode zur Spezifikation von eingebetteten
Systemen
- 2006-03 *Jens Kohlmeyer, Alexander Raschke, Ramin Tavakoli Kolagari*
Eine qualitative Untersuchung zur Produktlinien-Integration über
Organisationsgrenzen hinweg
- 2006-04 *Thorsten Liebig*
Reasoning with OWL - System Support and Insights –
- 2008-01 *H.A. Kestler, J. Messner, A. Müller, R. Schuler*
On the complexity of intersecting multiple circles for graphical display
- 2008-02 *Manfred Reichert, Peter Dadam, Martin Jurisch, Ulrich Kreher, Kevin Göser,
Markus Lauer*
Architectural Design of Flexible Process Management Technology

2008-03 *Frank Raiser*

Semi-Automatic Generation of CHR Solvers from Global Constraint Automata

2008-04 *Ramin Tavakoli Kolagari, Alexander Raschke, Matthias Schneiderhan, Ian Alexander*

Entscheidungsdokumentation bei der Entwicklung innovativer Systeme für
produktlinien-basierte Entwicklungsprozesse

Ulmer Informatik-Berichte
ISSN 0939-5091

Herausgeber:
Universität Ulm
Fakultät für Ingenieurwissenschaften und Informatik
89069 Ulm