Kognitive Analyse Formaler Sicherheitskritischer Steuerungssysteme auf Basis eines integrierten Mensch-Maschine-Modells

Andreas Lüdtke

Safety Critical Systems
Oldenburger Forschungs- und Entwicklungsinstitut für
Informatik-Werkzeuge und -Systeme (OFFIS)
Escherweg 2
26121 Oldenburg
luedtke@offis.de

Abstract: In dieser Dissertation wurde eine innovative Methode zur Analyse sicherheitskritischer modaler interaktiver Steuerungssysteme aus kognitiver Sicht erstellt. Ziel ist es, potentielles menschliches Fehlverhalten bei der Entwicklung möglichst früh zu berücksichtigen und nahtlos im Entwurfsprozess für weitere Analysen zur Verfügung zu stellen. Dadurch soll ein Beitrag zum menschenzentrierten Entwurf technischer Systeme geleistet werden.

1 Einleitung

Statistiken weisen in 60-80% der Unfälle mit modernen Automatisierungssystemen menschliches Versagen als Hauptursache aus. Einige der tragischsten Katastrophen dieser Art sind im Bereich der Luftfahrt zu verzeichnen. Verstärkt setzt sich die Erkenntnis durch, dass nur selten die Piloten allein, sondern vielmehr wenig intuitive Bedienkonzepte der Cockpitsysteme verantwortlich sind. Im Zuge der Automatisierung wurde die menschliche Steuerung mehr und mehr durch eine automatische Steuerung in Form von Autopiloten und Flight Management Systemen ersetzt, wobei sich die Rolle des Piloten vom aktiv Handelnden zum passiv Überwachenden wandelte. Dadurch konnten unbestritten bestimmte Fehlerquellen eliminiert werden, dennoch wurden neue Fehlerarten hervorgebracht, die weitgehend einem Missverständnis der menschlichen Fähigkeiten entspringen [Bi97]. Wesentlich ist in diesem Zusammenhang die Ausstattung der Systeme mit einer Vielzahl von Modi, um dasselbe Gerät für unterschiedliche Manöver nutzen zu können. Ein *Modus* kann als eine Systemkonfiguration mit einer spezifischen Funktionalität verstanden werden. Modi ermöglichen einerseits ein hohes Maß an Systemflexibilität, andererseits erhöhen sie den kognitiven Bedienungsaufwand. Ein Bericht der FAA (Federal Aviation Association) [ASS96] bestätigt, dass mangelnde Modeawareness ein wesentlicher Faktor für die Entstehung so genannter *Modusfehler* ist. Hierbei führt der Pilot eine Aktion durch, die in bestimmten Modi korrekt im aktuellen aber inkorrekt ist, was zu überraschendem Verhalten der Automatik führen kann.

Potentielle Bedienungsfehler werden in der Praxis durch Beobachtung von Piloten im Flugsimulator antizipiert. Dieses Vorgehen stößt aufgrund der Komplexität der Geräte an seine Grenzen und kann zudem erst spät im Entwicklungsprozess durchgeführt werden. Die Forschung antwortet auf dieses Problem mit der Forderung nach einem menschenzentrierten Design und der Bildung des Forschungsbereichs *Cognitive Engineering*. Auf diesem Gebiet werden zahlreiche Ansätze verfolgt. Dennoch gibt es bisher nur wenige Tools, die Systeme aus kognitiver Sicht bewerten. Eine besondere Herausforderung stellt die Berücksichtigung inkorrekter Annahmen über das Systemverhalten auf Seiten der Bediener dar. Hier existiert derzeit keine befriedigende Lösung. Aufgrund dieses Mangels wurde in der Dissertation orientiert an einer bei der Lufthansa Verkehrsfliegerschule durchgeführten Simulatorstudie ein Bedienermodell erstellt, das den kognitiven Prozess „gelernter Sorglosigkeit" nachbildet, und in der Lage ist, so genannte Routinefehler vorherzusagen. Diese Fehlerart wurde in der Literatur als wahrscheinlichste Erklärung für zahlreiche Vorfälle und Unfälle identifiziert.

Die Darstellung beginnt im folgenden mit einer kurzen Beleuchtung des State of the Art und der theoretischen Grundlagen (Abschnitt 2). Nach der empirischen Studie (Abschnitt 3) wird aufbauend darauf die Konzeption des kognitiven Modells und ein Schema zur formalen Systemmodellierung dargestellt (Abschnitt 4). Anschließend wird die Anwendung der Methode innerhalb einer implementierten Human-Simulation-Plattform beschrieben (Abschnitt 5). Der Beitrag schließt mit einem kurzen Ausblick (Abschnitt 6).

2 State of the Art und Grundlagen

Ziel der Cognitive Engineering Ansätze ist es, durch formale Analysen wiederholbare Evidenzen über die Adäquatheit eines Systems zu liefern. Dabei wird vielfach von der Annahme ausgegangen, dass Menschen ein mentales Modell der Maschine, die sie bedienen, bilden und dass dieses maßgeblich die Interaktion mit dem Gerät bestimmt. Bestehende Ansätze lassen sich entweder als entwurfs- oder bedienerzentriert charakterisieren. Im Zentrum entwurfszentrierter Verfahren (z.B. [DH02], [MP99]) stehen Systemmodelle. Mittels *formaler Verifikation* können diese auf die Einhaltung bestimmter Benutzererwartungen analysiert werden. Der Vorteil ist die mathematisch vollständige Analyse. Die Herleitung realistischer Benutzererwartungen (mentales Modell) wird allerdings vorausgesetzt. Im Zentrum bedienerzentrierter Verfahren (z.B. [Co00]) stehen Bedienermodelle. Diese werden genutzt, um potentielles Benutzerverhalten durch *Human Simulation* zu antizipieren. Der Vorteil ist, dass komplexe kognitive Prozesse berücksichtigt werden können. Nachteil ist, neben der generellen Unvollständigkeit von Simulationen, auch hier, dass realistische mentale Modelle a-priori gegeben sein müssen. Der Beitrag der Dissertation besteht in der Entwicklung einer integrierten Methodik, die erstens kognitive Modellierung anwendet, um realistische mentale Modelle psychologisch plausibel durch Human Simulation zu antizipieren. Der Fokus liegt dabei auf kognitiven Lernmechanismen. Zweitens werden Teile des kognitiven Modells in Entwurfsnotationen übersetzt, um formale Verifikationstechniken zur vollständigen Fehleridentifikation anwenden zu können.

Das erstellte kognitive Modell basiert auf der Theorie der *gelernten Sorglosigkeit* von Frey und Schulz-Hardt [FSH97]. Menschen lernen Sorglosigkeit, wenn sich beim Handeln in sicherheitskritischen Umgebungen Erfolge einstellen, obwohl aus Gründen der Bequemlichkeit oder Zeitersparnis Sicherheitsvorkehrungen außer Acht gelassen werden. Je häufiger sich der Erfolg beim sicherheitswidrigen Handeln einstellt, desto mehr verfestigt sich ein Zustand der Sorglosigkeit. Ein gewisses Maß an Sorglosigkeit ist durchaus notwendig, denn übervorsichtiges Handeln kann in komplexen Umgebungen zur Handlungsunfähigkeit führen. Deshalb muss es bei der Analyse der Bedienung sicherheitskritischer Systeme darum gehen, die Interaktionabläufe aufzudecken, an denen sorgloses Handeln gefährlich sein kann. Aufgrund der kognitiven Inhärenz lässt sich die Entstehung von Sorglosigkeit nicht durch Training abstellen, sondern muss während der Systementwicklung durch geeignete Entwurfsmaßnahmen reduziert werden.

In der Dissertation wurde die Entstehung gelernter Sorglosigkeit innerhalb einer kognitiven Architektur modelliert. Die erstellte Architektur basiert auf etablierten Ansätzen der Kognitionspsychologie, wie ACT-R [AL98] und ISP-DL [MST95]. Das Ziel kognitiver Architekturen in der Kognitionspsychologie ist es, ein möglichst genaues Modell menschlichen Verhaltens zu entwickeln. Unterschiedlich hierzu müssen beim Cognitive Engineering die Verhaltensvorhersagen nur in soweit exakt sein, als dass sie eine Unterscheidung zwischen alternativen Systementwürfen hinsichtlich der Bediensicherheit ermöglichen. Aus diesem Grund wurden in dieser Arbeit aus unterschiedlichen Architekturen die Aspekte übernommen und adaptiert, die für das Modellierungsziel (gelernte Sorglosigkeit) relevant sind.

3 Empirische Flugsimulatorstudie

Als Grundlage für die Erstellung des kognitiven Modells wurde eine empirische Studie in den Full-Motion Simulatoren der Lufthansa Verkehrsfliegerschule in Bremen durchgeführt. Konkret wurde die Interaktion mit dem Autopiloten einer Piper Cheyenne PA42 IIIIA beobachtet und analysiert. Insgesamt wurde das Verhalten von vier Pilotenschülern videografiert (22,5 Stunden Videomaterial), Verlaufsdaten zur Interaktion mit dem Autopiloten transkribiert und in Aufgabenprotokolle segmentiert. Bedienungsfehler sind bei den Aufgaben zur Durchführung eines Steig- bzw. Sinkflugs (Change-Altitude) und zum Einschalten des Autopiloten mit anschließender Modusauswahl (Engage-Autoflight) zu verzeichnen. Für diese beiden Aufgaben liegen insgesamt 95 Protokolle mit 374 korrekten und 34 inkorrekten Aktionen vor. Die Fehler verteilen sich auf 29 Aufgaben, sodass also 30,53% der Aufgaben fehlerhaft durchgeführt wurden.

In Abbildung 1 ist ein Beispiel eines Fehlerszenarios veranschaulicht, das in ähnlicher Form bei mehreren Probanden zu beobachten war. Kurz nach dem Start wurde von den Fluglotsen die Clearance erteilt, auf 4000 Fuß zu steigen (Abbildung 1 ①). Der Pilot stellte den Alerter auf 4000 (②) und betätigte die ALTS-Taste (③) zur Armierung des ALTS-Modus. Anschließend erhöhte er die vertikale Geschwindigkeit (VG) mittels der ETRIM-Taste (④).

Um dabei sicherzustellen, dass die Indicated Airspeed (IAS) des Flugzeugs 160 Kn nicht unterschreitet, wird empfohlen, den IAS-Hold-Modus zur automatischen Stabilisierung der Geschwindigkeit bei Erreichen von 160 Kn zu aktivieren. Während der Pilot auf die sinkende Geschwindigkeit achtete und auf 160 Kn wartete, kam das Flugzeug der Zielhöhe bereits sehr nahe, sodass der Autopilot automatisch den ALTS-Modus von armiert auf aktiviert schaltete (⑤). In diesem Modus wird die VG automatisch verringert, um langsam auf die gewünschte Höhe „einzuschweben" (Capture-Phase). Als schließlich die Geschwindigkeit von 160 Kn erreicht war, drückte der Pilot die IAS-Taste (⑥). Überraschend für den Piloten schwebte das Flugzeug nicht auf die Zielhöhe von 4000 Fuß ein, sondern stieg darüber hinaus, d.h. die Höhe wurde „überschossen".

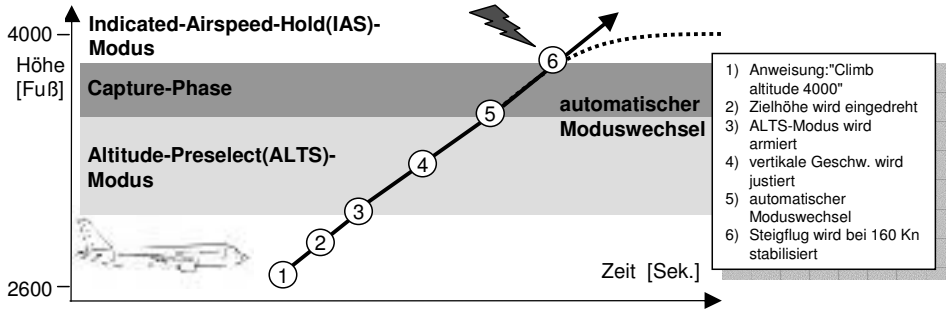

Abbildung 1: Flugbahn bei einem typischen Modusfehler

Der Fehler besteht in dem beschriebenen Szenario darin, dass der Pilot den IAS-Hold-Modus aktiviert, obwohl dies nach dem automatischen Wechsel in die Capture-Phase nicht mehr erlaubt ist. Bzgl. der IAS-Taste reagiert der Autopilot inkonsistent: Beim Betätigen der Taste *vor* der Capture-Phase wird die Aufgabe korrekt beendet, während beim Betätigen *in* der Capture-Phase ein Abbruch erfolgt. Somit liegt hier ein *Entscheidungspunkt* vor, wobei der Pilot jedes Mal vor der Aktion die Modusanzeige überprüfen muss, um sich situationsadäquat zu verhalten. Ein Großteil der beobachteten Fehler trat an solchen Entscheidungspunkten auf.

Durch Analyse der Fehlerraten wurde ein Phänomen aufgedeckt, bei dem sich die Probanden an den Entscheidungspunkten anfangs durchaus korrekt verhielten. Fehler waren erst dann zu verzeichnen, wenn über eine gewisse Anzahl von Aufgabenwiederholungen immer dasselbe Verhalten verlangt war und plötzlich eine abweichende Situation eintrat, in der das Verhalten umgestellt werden musste. Dies lässt vermuten, dass das korrekte Wissen durch Routinebildung verlernt wird. Deshalb sollen diese Fehler als *Routinefehler* bezeichnet werden. Wird berücksichtigt, dass die Autopilotenbedienung eine sicherheitskritische Aufgabe ist, dann lässt sich die Routinebildung im Licht der oben beschriebenen Theorie gelernter Sorglosigkeit interpretieren: Das Prüfen des aktuellen Modus ist eine Sicherheitsmaßnahme, die einen zusätzlichen Aufwand bedeutet und der Gefahr unterliegt, als überflüssig betrachtet zu werden, wenn sich der Erfolg wiederholt auch ohne sie einstellt. In der Dissertation liegt der Fokus auf der Analyse der Routinefehler. Ergebnis der Studie ist eine empirisch abgeleitete Hypothese zur Erklärung von Routinefehlern auf Basis gelernter Sorglosigkeit zusammen mit einer Datenbasis, die im Rahmen der Fallstudie (s.u.) zur Evaluierung des kognitiven Modells verendet wurde.

4 Konzept

Die entwickelte Methode dient zur Analyse der Bedienbarkeit eines formalen Systementwurfs mittels eines ausführbaren kognitiven Bedienermodells. In diesem Abschnitt wird zunächst die Architektur des kognitiven Modells beschrieben. Anschließend werden Schablonen für den Aufbau modaler Systementwürfe und schließlich ein Verfahren zur Integration und Analyse der beiden Modelle vorgestellt.

4.1 Kognitive Architektur

Das Ziel der kognitiven Modellierung innerhalb der Dissertation ist es, die Entstehung gelernter Sorglosigkeit zu formalisieren, um diesen Prozess für die Fehlerprognose nutzbar zu machen. Die wesentlichen Komponenten sind eine Wissensbasis, eine Wissensverarbeitungs-, eine Gedächtnis- und eine Lernkomponente. Zur symbolischen Modellierung gelernter Sorglosigkeit in der Lernkomponente wurde der Mechanismus der Regelkomposition aus der ISP-DL-Architektur [MST95] adaptiert. Diese Wahl wurde insbesondere aufgrund der dort detailliert ausgearbeiteten empirischen Indikatoren getroffen. Für die subsymbolische Modellebene wurden Konzepte aus ACT-R [AL98] erweitert. Durch die Entscheidung für die ISP-DL-Regelkomposition ist auch das Format zur Wissensrepräsentation festgelegt. Die ISP-DL-Architektur wurde bisher zum Offline-Planen in statischen Umgebungen verwendet. In der Dissertation wurde das dabei verwendete Goal-Means-Format zur Verwendung für das Online-Planen in dynamischen Umgebungen erweitert. Das Ergebnis sind datengesteuerte Ziel-Mittel-Relationen in Form von Produktionsregeln, abgekürzt als Goal-State-Means(GSM)-Regeln. Abbildung 2a zeigt das generelle Schema dieser Regeln. Der State-Teil im Regelkopf besteht aus Booleschen Ausdrücken über Umgebungsvariablen und erlaubt dem Modell in einer dynamischen Umgebung zu agieren. Die Regel hat die Bedeutung: *„Wenn* das aktuelle Ziel mit dem Ziel im Goal-Teil übereinstimmt und die Umgebungsbedingungen im State-Teil erfüllt sind, *dann* führe die Aktionen im Means-Teil aus und bearbeite als nächstes die Subziele im Subgoal-Teil.“ Die Subzielen können partiell geordnet sein.

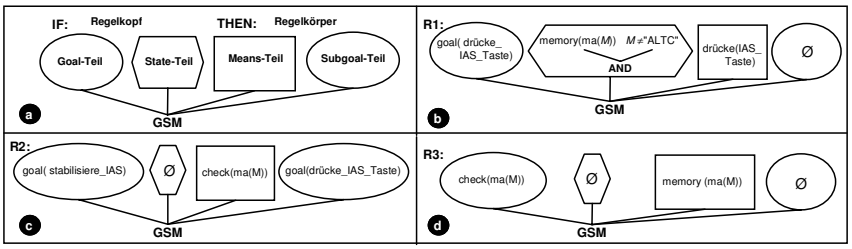

Abbildung 2: Schema und Beispiele für GSM-Regeln

Innerhalb der Wissensbasis wird mittels der GSM-Regeln das zur Durchführung einer Flugaufgabe notwendige Wissen formalisiert. Dabei entsteht das so genannte *Aufgabenmodell*. Ein Beispiel einer GSM-Regel ist in Abbildung 2b dargestellt. Die informelle Bedeutung lautet: *„Wenn* es das aktuelle Ziel ist, die IAS-Taste zu drücken und die Modusanzeige (ma) zeigt nicht „ALTC“ (Capture Modus), *dann* drücke die IAS-Taste.“

Im State-Teil der Regel wird zunächst der aktuelle Wert der Modusanzeige durch das Prädikat „memory“ aus dem Gedächtnis abgerufen und erst anschließend ausgewertet. Der Wert muss vorher durch eine Perzeptaktion gelesen worden sein. Dies geschieht in R2 (Abbildung 2c). Hier wird das Ziel, die Geschwindigkeit zu stabilisieren, durch eine Perzeptaktion (check(ma(M))) auf der Modusanzeige und anschließender Ableitung des Subziels zum Drücken der IAS-Taste realisiert. Jeder gelesene Wert muss durch eine spezielle Regel explizit in das Gedächtnis geschrieben werden. Beispielsweise speichert R3 (Abbildung 2d) den Wert der zuvor durchgeführten Perzeptaktion check(ma(M). Zur Anwendung von R3 muss die Perzeptaktion mental in ein Ziel transformiert werden.

Bei der Wissensverarbeitung werden entsprechend der aktuellen Umgebungssituation und des aktuellen Ziels Regeln aus der Wissensbasis ausgewählt und gefeuert. Nach jeder *erfolgreichen* Aufgabendurchführung werden die Regeln innerhalb der Lernkomponente durch Regelkomposition modifiziert. Hierbei werden zwei mehrfach erfolgreich nacheinander angewendete Regeln zu einer einzigen verschmolzen, die Regelköpfe und Regelkörper werden vereinigt. Entscheidend ist, dass dabei Regelelemente, die sowohl im Körper der ersten als auch im Kopf der zweiten Regel vorhanden sind, wegfallen. Durch diesen Prozess werden Subziele und unter bestimmten Bedingungen auch Perzeptaktionen eliminiert. Zu Beginn enthält die Wissensbasis normative Regeln. Nach mehrmaliger erfolgreicher Durchführung von Flugaufgaben können sorglose Regeln entstehen. Das Kompositum C1 in Abbildung 3a entsteht aus R2 und R3. Die Perzeptaktion wurde eliminiert, sodass C1 direkt "ALTS" (den zuletzt gelesenen Wert) in das Gedächtnis speichert. C1 und R2 beziehen sich auf dasselbe Ziel und sind somit konkurrierende Regeln. Bei der Wissensverarbeitung erfolgt die Regelauswahl auf Basis von Regelstärken, sodass C1 erst dann angewendet wird, wenn die zugehörige Regelstärke höher ist. Ist dies der Fall und C1 wurde erfolgreich angewendet, dann kann das Kompositum weitergehend mit R1 zu C2 (Abbildung 3b) komponiert werden. Die Anwendung der Komposita C1 und C2 führt zu gelernter Sorglosigkeit, weil die Regeln implizieren, dass die Modusanzeige konstant anzeigt, dass die Capture-Phase noch nicht erreicht ist („ATLS“). In C2 wird die IAS-Taste ohne jegliche Vorbedingung betätigt. Das Modell würde in dem oben beschriebenen Steigflugszenario unter Verwendung dieser Regel denselben Fehler begehen, wie der Pilot dort.

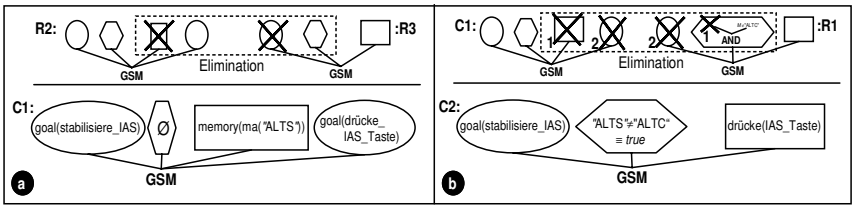

Abbildung 3: Beispiele für Regelkomposita

Die vorgestellte kognitive Architektur ist Aufgaben-unabhängig. Eine Spezialisierung erfolgt erst, wenn ein konkretes Aufgabenmodell in die Wissensbasis geladen wird. In diesem Sinne lässt sich die kognitive Architektur als generischer Interpreter für mentale Aufgabenmodelle verstehen.

Diese an etablierten kognitiven Architekturen orientierte Vorgehensweise ermöglicht, Erkenntnisse der Kognitionspsychologie anzuwenden und präzise Hypothesen über das Verhalten von Piloten zu generieren, die einer empirischen Überprüfung zugänglich sind.

4.2 Systemmodell

In modalen Systemen besteht die Aufgabe des Bedieners darin, mehrere für die aktuelle Aufgabe korrekte Modi auf die richtige Weise auszuwählen. Diese Perspektive lässt sich in Form einer Moduslogik aus Black-Box-Sicht modellieren. Zur Unterstützung der Systementwickler wurden in der Dissertation Schablonen erstellt, die lediglich instantiiert werden müssen. Den Schablonen unterliegt eine konsequente Konzeptualisierung und Formalisierung des in der Literatur meist schwammig verwendeten Modus-Begriffs.

Auf Basis der Produktfamilien-Architektur von Miller und Potts [MP99] wurde eine Formalisierung innerhalb der Modellierungsnotation des industriellen Case Tools Statemate (Activity- und Statecharts) vorgenommen. Dabei werden die einzelnen Modi modular definiert und in Modusgruppen zueinander in Beziehung gesetzt, um eine leicht erweiterbare und änderbare Struktur zu erzielen. Zur automatischen Überprüfung mittels des Modelchecker-Plug-Ins von Statemate (ModelCertifier), wurden notwendige Moduseigenschaften, z.B. dass zu jedem Zeitpunkt genau ein Modus aktiv sein muss, formal in LTL-Syntax (Linear Temporal Logic) spezifiziert.

4.3 Integration und Analyse

Das kognitive Modell wird innerhalb einer Simulationsplattform mit Statemate und einer Flugsimulatorsoftware gekoppelt und ausgeführt. Ein Schnappschuss des dabei gelernten Aufgabenmodells wird zwecks vollständiger Fehleridentifikation manuell in die Entwurfsnotation übersetzt und mit dem System integriert. Um valide Ergebnisse zu erhalten, muss eine semantische Äquivalenz zwischen Original und Übersetzung in Bezug auf die erzeugten Aktionssequenzen gewährleistet werden. Der Ansatz besteht darin, das symbolische Aufgabenmodell weitgehend strukturerhaltend auf äquivalente State- und Activitychart-Strukturen zu übersetzen. Die subsymbolische Lösung von Regelkonflikten auf Basis der Regelstärken wird in einer Vorverarbeitungsstufe durch Erweiterung der symbolischen Regelköpfe aufgelöst. Das integrierte Mensch-Maschine-Modell wird schließlich durch ein abstraktes Umweltmodell ergänzt und auf Einhaltung des obersten Aufgabenziels hin analysiert. Bei einer korrekten Aufgabendurchführung nähert sich das Flugzeug einem vorgegeben Zielwert, erreicht ihn und behält ihn bei. Formale LTL-Varianten dieses Ablaufs werden mit dem Statemate Modelchecker automatisch geprüft. Im negativen Fall, d.h. der Zielwert wird über- bzw. unterschossen oder erst gar nicht erreicht, liefert der Modelchecker das Analyseergebnis „false“. Zusätzlich wird ein Simulationsfile als Beleg generiert, welches innerhalb von Statemate visualisiert werden kann. Bei der Analyse wird vorausgesetzt, dass die zu untersuchende Aufgabe bei Anwendung des normativen Aufgabenmodells immer korrekt verläuft. Da dies vorher durch Verifikation sichergestellt werden kann, lassen sich identifizierte Verletzungen auf den strukturellen Unterschied zum gelernten sorglosen Modell zurückführen.

5 Fallstudie

Die Fallstudie soll einerseits die Machbarkeit der Methode demonstrieren und andererseits die Plausibilität des kognitiven Modells überprüfen. Analysiert wurde der Autopilot (AP), dessen Bedienung bereits Gegenstand der empirischen Studie war. Es wurde ein siebenstufiges Vorgehensmodell erarbeitet. In *Schritt 1* (*Systemmodellierung*) wurde der Entwurf des AP unter Verwendung der Modusschablonen rekonstruiert. In *Schritt 2* (*Strukturanalyse*) wurden Entscheidungspunkte im Entwurf aufgedeckt, die hinsichtlich bestimmter Flugaufgaben anfällig für gelernte Sorglosigkeit sein können und deshalb weiter untersucht werden müssen. Aufbauend darauf wurde in *Schritt 3* (*Aufgabenanalyse*) ein Aufgabenmodell für die korrekte AP-Bedienung erstellt. In *Schritt 4* (*Szenariendefinition*) erfolgte eine Formalisierung der von den Probanden der empirischen Studie geflogenen Szenarien (initiale Flugzustände und Ereignisse). Zur Realisierung der *Human Simulation* in *Schritt 5* wurde das kognitive Modell in PROLOG und eine Simulationsplattform in JAVA implementiert. Die Plattform koppelt das kognitive Modell mit Statemate und dem Microsoft Flight Simulator (für die Flugdynamik), sodass eine Closed-Loop-Simulation möglich ist. Für jedes definierte Szenario wird ein Simulationslauf mit jeweils anschließender Komposition der angewendeten Regeln durchgeführt.

Zur Überprüfung des kognitiven Modells wurde das Modellverhalten, insbesondere die Aktionen, mit dem Verhalten der vier Probanden der Simulatorstudie verglichen. Die Aufgabenprotokolle von Proband A wurden mit einem Model-Tracing-Verfahren rekonstruiert, um die *Vollständigkeit* zu prüfen. Die Rekonstruktion wurde zunächst ohne und anschließend mit Regelkomposition durchgeführt. Beim zweiten Versuch zeigte sich eine zum Teil signifikant (McNemar-Test) bessere Übereinstimmung mit den Daten, wobei sämtliche Routinefehler (sieben) korrekt rekonstruiert werden konnten. Die Protokolle der Probanden B,C und D wurden zur Überprüfung der Prognosefähigkeit/*Validität* des Models verwendet. Der Vergleich der prognostizierten und tatsächlichen Sorglosigkeit zeigt, dass die Routinefehler von Proband B bei der Aufgabe Change-Altitude und von Proband D bei Engage-Autoflight korrekt vorhergesagt werden. Die Ergebnisse der Modellprüfung zeigen also, dass das Modell Fehler begeht, die mit den Routinefehlern der Probanden A, B und D übereinstimmen. Darüber hinaus zeigen sich sowohl im Modell- als auch im Pilotenverhalten typische Indikatoren (schnellere Performanz und Interleaving der Aktionen) für Regelkomposition. Verhaltensabweichungen lassen sich insbesondere dadurch erklären, dass das Modell derzeit keine Multitasking Funktionalität besitzt. Es sei betont, dass es aufgrund der Methodik der Einzelfalluntersuchungen hier nicht möglich ist, die Vollständigkeit und Validität zu *beweisen* (Hypothesentest). Vielmehr geht es im Sinne von Dörner [Dö89] darum, die kognitive Adäquatheit der Modellierung empirisch zu überprüfen (Hypothesenbildung).

In *Schritt 6* (*Fehleridentifikation*) wird ein Schnappschuss des gelernten potentiell sorglosen Aufgabenmodells manuell in die Entwurfsnotation übersetzt und mit dem Systemmodell integriert. Mittels formaler Verifikation sollen automatisch Szenarien generiert werden, in denen das gelernte Modell zu Bedienungsfehlern und in der Folge zur Verletzung des Aufgabenziels führt. Das Ziel der Aufgabe Change-Altitude besteht darin, im Falle einer neuen Fluglotsenanweisung (Clearance) die intendierte Höhe zu erreichen und beizubehalten.

Die Verifikation der entsprechenden LTL-Formel ergibt “false”. Das resultierende Simulationsfile kann als Waveform (Abbildung 4) visualisiert werden.

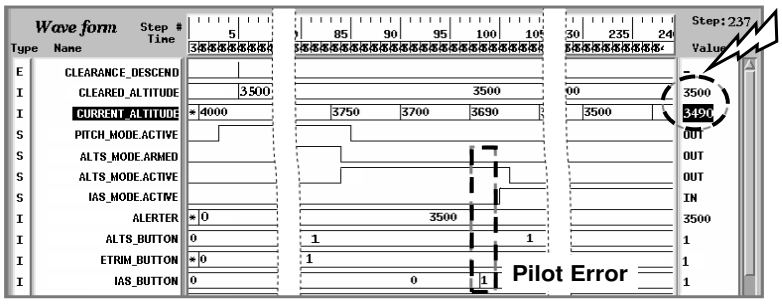

Abbildung 4: Waveform eines automatisch prognostizierten Bedienungsfehlers

Die Waveform zeigt die Veränderung der Variablen über die Zeit. In dem generierten Szenario befindet sich das Flugzeug anfangs bei 4000 Fuß (CURRENT_ALTITUDE) und soll auf 3500 Fuß sinken (CLEARED_ALTITUDE). Ab Step 237 wird die freigegebene Höhe unterschossen. In Step 98 wird die IAS-Taste gedrückt, obwohl der ALTS-Mode aktiv ist (ALTS_MODE.ACTIVE). Da dies im normativen Modell nicht möglich ist, muss eine sorglose Regel entstanden und angewendet worden sein. Durch die Verifikation konnte ein Szenario aufgedeckt werden, in dem diese Sorglosigkeit zur Verletzung des Aufgabenziels führt. In *Schritt 7* (*Entwurfsmodifikation*) wurde der AP-Entwurf so geändert, dass die IAS-Taste im ALTS-Modus keinen Effekt hat. Bei einer erneuten Analyse mit dem Modelchecker tritt der zuvor identifizierte Fehler nicht mehr auf. Allerdings wird eine weitere potentielle Verletzung generiert, die ebenfalls durch Modifikationen verhindert werden muss. Anschließend wird iterativ fortgefahren, bis sich das Modell robust gegenüber der prognostizierten gelernten Sorglosigkeit erweist.

6 Zusammenfassung und Ausblick

Ergebnis der Dissertation ist eine Methode zur kognitiven Analyse mit einem Vorgehensmodell, einem lernenden kognitiven Modell, Schablonen für Moduslogiken, einem Verfahren zur Übersetzung mentaler Aufgabenmodelle und einer werkzeugseitigen Unterstützung in Form einer Human-Simulation-Plattform. Die Analyse ist anwendbar für modusbasierte sicherheitskritische Steuerungssysteme, deren Bedienung durch eine endliche Menge von Regeln angegeben werden kann. Aktuell wird in Kooperationen eine Erweiterung des kognitiven Modells um zusätzliche entwurfsrelevante Prozesse wie Multitasking, Aufmerksamkeitskontrolle und Workload angestrebt. In dem EU-Projekt ISAAC wird derzeit die industrielle Anwendung der Methode in Zusammenarbeit mit Airbus-France, Alenia Aeronautica und Saab evaluiert.

Literaturverzeichnis

[AL98] Anderson, J.R.; Lebiere, C.: Atomic Components of Thought. Erlbaum, Hillsdale, 1998.

[ASS96] Abbott, K.; Slotte, S.; Stimson, D.: The Interface between Flightcrews and Modern Flight Deck Systems. Federal Aviation Administration, Seattle, WA, 1996.

[Bi97] Billings, C.E.: Aviation Automation: the Search for a Human-Centered Approach. Erlbaum, Mahwah, NJ, 1997.

[Co00] Corker, K. M.: Cognitive Models and Control. In (Sarter, N.B.; Amalberti, R. Hrsg.): Cognitive Engineering in the Aviation Domain. Erlbaum, Mahwah, NJ, 2000. S. 13-42.

[DH02] Degani A.; Heymann M.: Formal Verification of Human-Automation Interaction. In: Human Factors, 2002, Band 44, Nr. 1, S. 28-43.

[Dö89] Dörner, D.: Die kleinen grünen Schildkröten und die Methoden der experimentellen Psychologie. In: Sprache und Kognition, 1989, Band 8, Nr. 2, S. 86-97.

[FSH97] Frey, D.; Schulz-Hardt, S.: Eine Theorie der gelernten Sorglosigkeit. In (Mandl, H. Hrsg.), Bericht über den 40. Kongress der Deutschen Gesellschaft für Psychologie. Hogrefe Verlag für Psychologie, 1997, S.604-611.

[MP99] Miller, S. P.; Potts, J. N.: Detecting Mode Confusion through Formal Modeling and Analysis. Advanced Technology Center, Rockwell Collins, Inc., 1999.

[MST95] Möbus, C.; Schröder, O.; Thole, H.J.: Online Modeling the Novice-Expert Sift in Programming Skills on a Rule-Schema-Case Partial Order. In (Wender, K.F.; Schmalhofer, F.; Böcker, H.-D. Hrsg.): Cognition and Computer Programming. Ablex, Norwood, 1995.

Andreas Lüdtke wurde 1969 in Bramsche geboren. Er nahm im Wintersemester 1989 ein Informatikstudium an der Carl von Ossietzky Universität in Oldenburg auf, das er im Januar 1997 mit dem Diplom abschloss. 1997-2001 arbeitete Herr Lüdtke als wissenschaftlicher Mitarbeiter am Oldenburger Forschungs- und Entwicklungsinstitut für Informatik-Werkzeuge und -Systeme (OFFIS) an den Themen Sicherheitsanalyse und Cognitive Engineering. Die Dissertation wurde im November 2000 mit einer empirischen Vorstudie bei der Lufthansa Verkehrsfliegerschule in Bremen begonnen. 2002 wechselte Herr Lüdtke als Assistent an die Universität Oldenburg, in die Abteilung Lehr- und Lernsysteme des Fachbereichs Informatik, wo er die Arbeiten im Themenfeld Cognitive Engineering intensivierte und schließlich am 15. Oktober 2004 seine Promotion absolvierte. Seit 2004 bearbeitet Herr Lüdtke das Thema Human Centred Design in Kooperation mit führenden Flugzeugherstellern in der Abteilung Safety Critical Systems am OFFIS.