

Abschlußbericht DFG-Forschungsstipendium:
**Vermeidung von Feature-Interaktionen
in Intelligenten Netzwerken
durch ein modulares Architekturmodell**

Dr. Jan Brederke

Universität Oldenburg, FB Informatik · Postfach 2503 · D-26111 Oldenburg

DFG-Geschäftszeichen: Br 1797/1-1

1 Einführung und Zusammenfassung der Antragsziele

Zu den größten jemals realisierten Softwaresystemen im Kommunikationsbereich zählen die Telefonvermittlungssysteme. Durch die Liberalisierung der Telefonmärkte entsteht Druck, möglichst viele neue Dienste und Leistungsmerkmale anzubieten, und so werden diese Systeme zur Zeit in viele Richtungen erweitert. In vielen Telefonvermittlungssystemen existieren derzeit bereits Hunderte von Leistungsmerkmalen (Features), die über den Basis-Telefondienst hinausgehen. Daher ist die Wahrscheinlichkeit hoch, daß das Hinzufügen eines weiteren Leistungsmerkmals zu einem solchen System ein bereits vorhandenes Leistungsmerkmal beeinflussen wird, insbesondere auch auf unerwünschte Weise. Dies wird als *Feature-Interaktion* bezeichnet. Viele dieser Probleme werden erst in der Betriebsphase gefunden, und zwar von den (zu Recht verärgerten) Kunden. Dies hindert die Telefongesellschaften zunehmend, weitere neue Dienste einzuführen.

Neben der Anwendung von Verfahren zur Erkennung und Auflösung unerwünschter Feature-Interaktionen ist es sinnvoll, solche Probleme möglichst von vornherein zu vermeiden. Viele der Probleme sind bereits in der Anforderungsspezifikation eines Telefonvermittlungssystems und seiner zusätzlichen Leistungsmerkmale vorhanden. Da hier die Struktur des Systems noch gut zu erkennen und zu untersuchen ist, konzentrierte sich das Projekt hauptsächlich auf diese Ebene.

Feature-Interaktions-Probleme sind eine Ausprägung eines allgemeinen Software-Engineering-Problems: Wie kann man Software-Systeme adäquat erweitern, wenn diese so groß sind, daß eine einzelne Person sie nicht mehr überschauen kann? Ein allgemeiner Lösungsansatz für dieses schon lange bekannte Problem ist, das Software-System modular zu entwerfen, und zwar unter Beachtung des Geheimnisprinzips. Damit wird ein Teile-und-Herrsche-Ansatz möglich.

Ziel des hier beschriebenen Projektes war es daher zu untersuchen,

1. wie ein Basis-Telefonvermittlungssystem in möglichst unabhängige Komponenten strukturiert werden kann,
2. von welcher Art die Schnittstellen sein sollten, an denen Erweiterungen hinzugefügt werden können, und
3. mit welcher Methodik Erweiterungen vorgenommen werden sollten.

2 Durchgeführte Arbeiten und Ergebnisse

Das Projekt wurde im Communications Research Lab (CRL) an der McMaster University in Hamilton, Kanada, durchgeführt. Der Projektbearbeiter arbeitete dort insbesondere zusammen mit David Parnas und mit Dennis Peters im Zusammenhang mit Funktionaler Dokumentation, und mit Emil Sekerinski im Zusammenhang mit formaler Verfeinerung und Objektorientierung.

2.1 Zusammenfassung der Ergebnisse

In einer Fallstudie wurde eine formale Anforderungsspezifikation für ein vereinfachtes Telefonvermittlungssystem erstellt [1]. Dabei wurde darauf geachtet, dessen einzelne Aspekte separat zu spezifizieren (Punkt 1 der Ziele-Liste). Insbesondere stellte sich heraus, daß in den Spezifikationen in den Standards und in der meisten Literatur *bisher eine zu niedrige Abstraktionsebene für die Benutzerschnittstelle gewählt* wird. Daher wurde gezeigt, wie man deren Details inkapseln kann und damit viele Feature-Interaktionen vermeidet. Es wurde auch untersucht, wie sich dieser Ansatz auf die aktuelle Architektur des Intelligenten Netzwerkes anwenden läßt.

Erweiterungen der Anforderungen wurden formal als Verfeinerung spezifiziert (Ziel 3). Im Gegensatz zu Erweiterungen stellen Änderungen der vorhandenen Anforderungen bei diesem Vorgehen eine Rücknahme von Verfeinerungsschritten dar. In der Praxis sind Rücknahmen von Eigenschaften und Annahmen sowohl unvermeidlich als auch häufig. Daher wurde untersucht, wie man möglichst viele Verfeinerungsschritte, d.h. Leistungsmerkmale oder Teile davon, bei Änderungen unverändert wiederverwenden kann. Für solche *Familien von Anforderungsspezifikationen* ergaben sich drei neue wesentliche Vorgehensweisen: *Abhängigkeiten zwischen Verfeinerungsschritten für die Anforderungen werden explizit dokumentiert*, nicht nur die historische Reihenfolge der Verfeinerungsschritte; die Schritte werden entsprechend der Wahrscheinlichkeit geordnet, mit der sie später zurückgenommen werden müssen, mit den *grundlegendsten Anforderungen zuerst*; und für die syntaktische Überprüfung, ob die Rücknahme eines Schrittes eine wesensbestimmende Eigenschaft eines Leistungsmerkmals berührt, wird für jede Eigenschaft eines Leistungsmerkmals *explizit dokumentiert, ob sie fundamental oder änderbar ist* (“must – may”).

Als Schnittstelle, über die Erweiterungen zu einer Anforderungsspezifikation hinzugefügt werden, wurde eine bestimmte Art von *Vererbung* gewählt (Ziel 2). Setzt eine Erweiterung auf mehreren Leistungsmerkmalen auf, so wird entsprechend Mehrfach-Vererbung verwendet.

2.2 Detaillierte Darstellung

Im einzelnen wurden die folgenden Arbeiten durchgeführt beziehungsweise die folgenden Ergebnisse erzielt.

2.2.1 Anforderungsspezifikation mit Funktionaler Dokumentation

Es ist wichtig, daß die Anforderungen an ein Telefonvermittlungssystem vor Beginn der Implementationsarbeiten präzise und vollständig spezifiziert werden, um mit den Kunden zu klären, ob die Anforderungen ihren Wünschen entsprechen, und um die Konsequenzen des Kombinierens von Leistungsmerkmalen untersuchen zu können. Als Mittel der präzisen Dokumentation können dazu insbesondere formale Beschreibungstechniken verwendet werden.

Im Communications Research Lab wird zur formalen Spezifikation von Anforderungen *Funktionale Dokumentation* verwendet, auch bekannt als "Parnas-Tabellen" und eng verwandt mit den Varianten "A-7-Methode" und "SCR"-/"NRL"-Methode. Diese an die Kontrolltheorie angelehnte und mathematisch wohlfundierte Beschreibungstechnik erlaubt es, Anforderungen ausschließlich in Begriffen der Umgebung des zu konstruierenden Systems aufzuschreiben, und sie vermeidet damit, vorzeitig irgendwelche Implementationsentscheidungen zu treffen. Es werden Relationen ("Funktionen") zwischen den beobachteten und den kontrollierten Variablen spezifiziert, oft in einer kompakten und gut lesbaren Tabellenform. Diese Methode ist in der Praxis bereits unter anderem für die Spezifikation der gesamten Anforderungen an den Flugcomputer des A-7E-Kampfflugzeugs verwendet worden und für die Spezifikation des Notabschaltungssystems eines kanadischen Candu-Atomkraftwerks. Damit hat diese Technik ihre Anwendbarkeit für große Softwaresysteme unter Beweis gestellt. Weitere wichtige Aspekte der Funktionalen Dokumentation sind, daß sie gezielt vermeidet, dieselbe Information an verschiedenen Stellen der Spezifikation aufzuführen, um Inkonsistenzen bei späteren Änderungen zu vermeiden, und daß sie explizit zwischen Eigenschaften des zu konstruierenden Systems und Eigenschaften seiner Umgebung unterscheidet. Letzteres teilt nicht nur die Verantwortlichkeiten zwischen Entwickler und Kunde klar auf, es ist auch für die Komposition von Leistungsmerkmalen wichtig, die sich auf Teile eines vorhandenen Systems als Umgebung verlassen müssen.

Aufgrund der beschriebenen Vorteile und aufgrund der Verfügbarkeit einiger der Entwickler dieser Technik als Experten wurde die Funktionale Dokumentation als Grundlage für eine Fallstudie zur Spezifikation von Anforderungen an ein Telefonvermittlungssystem gewählt.

2.2.2 Trennung von Benutzerschnittstelle und Funktionalität

Zu Beginn der Arbeiten an der Fallstudie zeigte sich, daß der Auswahl von geeigneten beobachteten und kontrollierten Variablen aus der Umgebung des zu entwickelnden Systems eine wichtige Rolle zukommt. Diese Wahl ist gleichzeitig die Wahl einer geeigneten Abstraktionsebene der Beschreibung.

Anforderungsspezifikationen von Telefonvermittlungssystemen in den Standards und in der meisten Literatur verwenden als Variablen (bzw. als Ereignisse bei ereignisorientierten Techniken) meist einzelne Tastendrucke, Signaltöne usw. Weiterhin haben typische Endgeräte nur wenige dieser Signale. Andererseits gehen die Spezifikationen von neuen Leistungsmerkmalen oft davon aus, daß sie die alleinigen Benutzer dieser Signale sind, was

ein Grund für viele Feature-Interaktionen ist.

Dies führte zu der Idee, daß auch in den Anforderungen für Telefonvermittlungssysteme die Details der Benutzerschnittstelle eingekapselt und von der Funktionalität des Basissystems und der zusätzlichen Leistungsmerkmale getrennt werden sollten. Für Computer unterscheidet die Forschung zur Mensch-Maschine-Schnittstelle bereits seit langem zwischen syntaktischen und semantischen Signalen. Entsprechend wurde in diesem Projekt vorgeschlagen, ein Information-Hiding-Modul zu verwenden, um diese Details zu kapseln und um die wenigen syntaktischen Signale in hinreichend viele semantische Signale hin- und zurück zu übersetzen. Bei Erweiterungen des Systems können so außerdem leicht weitere semantische Variablen hinzugefügt werden, zum Beispiel für einen Wunsch nach einem Benutzerrollentransfer. Syntaktische Variablen, z.B. eine neue Taste, müssen dafür nicht hinzugefügt werden. Interaktionen aufgrund gemeinsamer Benutzung syntaktischer Variablen durch verschiedene Leistungsmerkmale treten nur noch innerhalb des Benutzerschnittstellen-Moduls auf und können dort koordiniert behandelt werden.

Bei der Kapselung der Benutzerschnittstelle muß man unterscheiden, ob die Anforderungen des Kunden Details wie die zu verwendenden Tasten und Töne festschreiben oder ob sie sie dem Entwickler überlassen. Es zeigte sich, daß in beiden Fällen eine Kapselung möglich ist. Sollen die Details nicht in den Anforderungen festgeschrieben werden, kann man "Variablen im Kopf des Benutzers" verwenden. Einzelheiten zu dieser Kapselung sind im technischen Bericht [1] ausgearbeitet und wurden in [2] veröffentlicht.

2.2.3 Abbildung der Trennung auf den Standard für das Intelligente Netzwerk

Auch der internationale Standard der ITU-T für das Intelligente Netzwerk trennt bisher nicht zwischen der syntaktischen und der semantischen Ebene der Benutzerschnittstelle. In [1, 2] wird gezeigt, daß es möglich ist, im Rahmen der Architektur dieses Standards ein geeignetes Information-Hiding-Modul in der Form eines Service-Independent-Building-Blocks (SIB) einzuführen.

2.2.4 Methodik der Erweiterung und Änderung von Anforderungen

In diesem Abschnitt sind Empfehlungen und Erkenntnisse zur Methodik der Erweiterung und Änderung von Anforderungen aufgeführt, die im Projekt entwickelt wurden.

Verfeinerung für die Erweiterung der Anforderungsspezifikation. Die formale Verfeinerung von Anforderungsspezifikationen ist ein wohlbekannter Ansatz. Es werden Schritt für Schritt mehr Details zur Spezifikation hinzugefügt, und jeder Schritt verringert den Nichtdeterminismus in der Spezifikation. In unserem Kontext ist es sinnvoll, zwischen *externer* und *interner* Verfeinerung zu unterscheiden. Bei externer Verfeinerung werden schrittweise Anforderungen zur Spezifikation hinzugefügt, bis diese vollständig ist. Die interne Verfeinerung beginnt mit einer vollständigen Anforderungsspezifikation, zu dieser werden schrittweise Implementationsdetails hinzugefügt, bis man eine ausführbare Implementation erhält. In beiden Fällen stellt man oft durch Validation oder auch per Kon-

struktion sicher, daß kein Schritt eine einmal vorhandene Anforderungseigenschaft wieder ungültig macht. Weiterhin hat man in beiden Fällen üblicherweise eine lineare Folge von Verfeinerungsschritten. Für die Erweiterung der Anforderungen an Telefonvermittlungssysteme sind wir nur an externer Verfeinerung interessiert.

Modifikation der Funktionalen Dokumentation für externe Verfeinerung. Bisher ist Funktionale Dokumentation dazu gedacht, die Anforderungen an ein *einzelnes* System zu beschreiben, wenn auch in einer Form, die Änderungen des Dokumentes erleichtert. Solche Änderungen sind allerdings irreversibel und die neuen Teile gehen im Dokument auf. Eine Gruppierung nach Leistungsmerkmalen und ein Verfahren zum Hinzufügen in sich abgeschlossener Beschreibungsstücke sind nicht vorgesehen. Für die Anwendung auf inkrementell erweiterbare Telefonvermittlungssysteme erschien es im Projekt notwendig, die Technik in diesem Punkt zu modifizieren. In der Fallstudie wurde jedes Stück an Funktionalität für sich nach dem Standardverfahren beschrieben, und anschließend wurden diese Teile nach Bedarf zu einem großen System komponiert. Damit wurde insbesondere die stufenweise Erweiterung von Telefonvermittlungssystemen abgebildet.

Grobkörnige Erweiterung und Änderung. Sowohl Erweiterungen als insbesondere auch Änderungen wurden in der Fallstudie ausschließlich auf der Ebene von ganzen (dort so genannten) “Eigenschaften” vorgenommen. Diese stellen in sich abgeschlossene, gewünschte Aussagen über das System oder seine Umgebung dar. In anderen Ansätzen ist es oft zulässig, daß Erweiterungen und Änderungen an beliebiger Stelle in die Anforderungsspezifikation eingreifen dürfen. Dies hat dort zur Folge, daß die Konsequenzen oft nur noch sehr schwer herauszufinden sind. Dadurch, daß in unserer Fallstudie Erweiterungen und Änderungen nur für ganze “Eigenschaften” vorgenommen wurden, konnte der Bereich, der von möglichen Konsequenzen betroffen war, syntaktisch leicht festgestellt werden.

Explizite, minimale Abhängigkeiten zwischen Verfeinerungsschritten. In der Spezifikation wurde explizit dokumentiert, welche anderen “Eigenschaften” von einer “Eigenschaft” für deren Definition vorausgesetzt wurden. So baut etwa die Eigenschaft, daß ein Eintrag in einer “Schwarzen Liste für unerwünschte Benutzer” Verbindungsversuche von diesem Benutzer unterbindet, auf der Definition der Schwarzen Liste auf, und diese wiederum auf dem Grundbegriff des Benutzers, der wiederum auf der Basis des Grundbegriffs des Menschen definiert ist. Andere Eigenschaften werden in diesem Beispiel nicht vorausgesetzt, wie etwa der Begriff des Endgerätes oder das Konzept der Weiterleitung von Verbindungen an andere Endgeräte oder an andere Benutzer. Damit sind die Abhängigkeiten syntaktisch explizit dokumentiert. So wurde ein gravierendes Problem der üblichen Verfeinerungsansätze im Hinblick auf Änderungen vermieden. Wenn in einer linearen Folge von Verfeinerungsschritten einer zurückgenommen wird, sind alle weiteren Schritte erst einmal ungültig und müssen zumindest erneut überprüft werden. Durch die expliziten Abhängigkeiten erhalten wir einen gerichteten, azyklischen Graphen. Sofern getrennte Anliegen in den Anforderungen getrennt spezifiziert werden, gibt es zwischen ihnen auch

keine Abhängigkeiten, was syntaktisch ersichtlich ist. Insbesondere hält dies den Spezifizierer dazu an, restliche Abhängigkeiten zu minimieren und nach geeigneten Gruppierungen zu suchen. Sofern eine Eigenschaft wirklich weitere Eigenschaften voraussetzen muß, lassen sich die Auswirkungen einer Änderung leichter herausfinden. Explizite Abhängigkeiten sind auch eine Verbesserung gegenüber dem Vorgehen bei dem Intelligenten Netzwerk: Dort wird jedes Leistungsmerkmal als eine Erweiterung nur des Basissystems angesehen, womit Abhängigkeiten zwischen den Leistungsmerkmalen erst einmal gar nicht dokumentiert werden.

Grundlegende Anforderungen zuerst. Weiterhin wurde in dem Projekt als wichtig erkannt, in welcher Reihenfolge Eigenschaften aufeinander aufbauen. Um den Aufwand bei Änderungen zu minimieren, sollten diejenigen Eigenschaften zuerst spezifiziert werden, von denen erwartet wird, daß sie sich am wenigsten wahrscheinlich ändern. Die Eigenschaften, die sich leicht ändern können, sollten im Abhängigkeitsgraphen so angeordnet werden, daß sie dicht bei den Blättern angeordnet sind, von denen nichts mehr abhängt. Die erweiterten Varianten einer Anforderungsspezifikation bilden eine *Familie von Anforderungsspezifikationen*, die große Teile des Abhängigkeitsgraphen gemeinsam hat. Die Idee, externe Verfeinerungsschritte nach ihrer Änderungswahrscheinlichkeit zu ordnen, ist daher verwandt mit der Idee von Parnas, für Familien von Programmen die internen Verfeinerungsschritte in Richtung auf eine Implementation nach der Wahrscheinlichkeit zu ordnen, mit der sich die zugehörigen Implementationsdetails ändern.

Fundamentale Eigenschaften eines Leistungsmerkmals explizit – “must and may”. In den Anforderungen für ein Leistungsmerkmal gibt es Eigenschaften, die für dieses Leistungsmerkmal wesensbestimmend sind, und es gibt oft auch Eigenschaften, die vom Spezifizierer aus praktischen Gründen mit spezifiziert werden, aber vom Kunden auch in anderer Form akzeptiert würden. Zum Beispiel ist es oft wünschenswert, daß das Verhalten eines Leistungsmerkmals bei jeder Aktivierung gleich ist, obwohl es mehrere sinnvolle Verhaltensvarianten gäbe. Hier werden Spezifizierer und Kunde eine geeignete Variante auswählen. Kommt es später zu Inkonsistenzen (d.h. Feature-Interaktionen) mit einem anderen Leistungsmerkmal, so wäre es kein Problem, die nicht fundamentalen Anforderungen des ersten Leistungsmerkmals geeignet zu ändern. Allerdings ist aus einer unstrukturierten Spezifikation nicht mehr ersichtlich, welche Anteile wesensbestimmend sind und welche nicht. Der Spezifizierer kann dies nicht zuverlässig entscheiden, da ihm oft das Anwendungswissen über das interagierende Leistungsmerkmal fehlt. Daher wurde in diesem Projekt systematisch und explizit zwischen fundamentalen und änderbaren Eigenschaften eines Leistungsmerkmals unterschieden.

2.2.5 Struktur und Erweiterungsschnittstellen der Fallstudie

Vom “Anruf” zur “Verbindung”. Ein großes konzeptuelles Problem aktueller Telefonvermittlungssysteme ist der Begriff des *Anrufs* (engl. “call”). Ein Anruf kann mittlerweile so viel Funktionalität umfassen, und der Begriff wird so unterschiedlich definiert,

daß er als Grundbegriff unbrauchbar ist. Daher verzichtet die Fallstudie ganz auf diesen Begriff und führt als einen Grundbegriff die *Verbindung* zwischen zwei Teilnehmern ein. Dieser Begriff wird wiederum getrennt von der Art der Verbindung, so daß ein Sprachkanal-Leistungsmerkmal separat auf der Basis des Verbindungs-Leistungsmerkmals spezifiziert wird. Auf die gleiche Weise ließe sich ein Bildkanal-Leistungsmerkmal ohne Duplizierungen und daher mit wenig Aufwand spezifizieren.

Explizite Konzepte. In der Fallstudie wurde darauf geachtet, Konzepte möglichst explizit einzuführen, um sie sauber voneinander zu trennen. Ein Beispiel ist die Unterscheidung von Endgeräten, Menschen und Benutzerrollen. Im einfachen alten Telefondienst (plain old telephone service, POTS) werden diese Begriffe implizit miteinander identifiziert, indem zum Beispiel eine Telefonnummer Instanzen aus allen drei Kategorien bezeichnen kann. Dies führt zu vielen Feature-Interaktionen, zum einen durch Verwirrung der Benutzer und zum anderen, weil für manche Leistungsmerkmale notwendige Information nicht zur Verfügung steht. In der Fallstudie wird die Annahme, daß es eine feste Zuordnung zwischen den Instanzen der drei Kategorien gibt, explizit in mehreren Eigenschaften spezifiziert und später für ein Weiterleitungs-Leistungsmerkmal explizit zum Teil wieder aufgehoben.

Hinzufügen von Konzepten nur explizit. Entsprechend wurden bei dem Hinzufügen von Leistungsmerkmalen notwendige neue Konzepte zuerst explizit zur Spezifikation hinzugefügt, bevor das eigentliche Leistungsmerkmal spezifiziert wurde. So wurde zunächst ein "Leistungsmerkmal" spezifiziert, das die Möglichkeit einführt, daß eine Verbindung blockiert werden kann, zusammen mit der (explizit änderbaren) Eigenschaft, daß eine solche Blockierung nie stattfindet. Danach wurden dann verschiedene konkrete Sperr-Leistungsmerkmale auf dieser Basis spezifiziert. (Das fundamentale Sperr-Leistungsmerkmal wiederum baute auf der bereits in dem fundamentalen Verbindungs-Leistungsmerkmal vorgesehenen Möglichkeit auf, daß das Fehlen oder Verlorengehen von notwendigen Ressourcen eine Verbindung verhindern oder abbrechen kann.)

Vererbung für die Erweiterungsschnittstellen. Erweiterungen wurden durch Hinzufügen von weiteren Eigenschaften zu den vorhandenen Eigenschaften vorgenommen (Konjunktion, Superposition), wobei die vorhandenen nur referenziert, aber nicht kopiert wurden. Dies ist eine Form von Vererbung. Da dabei z.T. auch mehrere, unabhängige Beschreibungen referenziert werden, handelt es sich dann um Mehrfach-Vererbung. Interessant ist hier insbesondere die Möglichkeit, nur einen Teil einer Beschreibung zu referenzieren, um damit die Abhängigkeiten explizit zu machen und um sie zu minimieren. Weiterhin erlaubt es die Methode, daß eine neue Eigenschaft eine der geerbten Eigenschaften explizit ersetzt. Dies ist eine relativ grobkörnige Art der Veränderung, die es erleichtert, die Konsequenzen herauszufinden, da zusammenhängende Teile nur zusammen ersetzt werden. Außerdem ist die explizite Unterscheidung zwischen fundamentalen und änderbaren Eigenschaften eines Leistungsmerkmals hier hilfreich, weil dadurch syntaktisch offensichtlich wird, ob eine Erweiterung mit einem anderen Leistungsmerkmal unerwünscht interferiert. Die Anzahl

der konkreten Punkte, an denen Erweiterungen vorgenommen werden können, hängt von den bereits vorhandenen Leistungsmerkmalen ab, da diese solche Punkte einführen. Damit wächst die Zahl der Erweiterungspunkte mit der Zahl der Leistungsmerkmale, und es kommt nicht mehr zu den bekannten Engpässen an den wenigen, ggf. nur schlecht für den jeweiligen Zweck passenden “Detection Points” des Intelligenten Netzwerks.

2.3 Weitere Aktivitäten

Die Idee der Trennung der Details der Benutzerschnittstelle von der Funktionalität des Telefonvermittlungssystems, unter Einführung geeignet vieler “semantischer” Variablen, wurde auf dem für die behandelte Problematik zentralen internationalen *Feature-Interaction-Workshop* [2] vorgestellt, mit den dortigen Experten aus Industrie und Forschung diskutiert und von ihnen gut aufgenommen.

Aufgrund der Behandlung auch von Software-Engineering-Aspekten in dem Projekt ergab sich eine Einladung für das *Dagstuhl-Seminar* für “eine neue Agenda der Software-Engineering-Forschung und -Lehre” [3] im Februar 1999.

Weiterhin wurden an folgenden Orten *eingeladene Vorträge* zur Thematik des Projekts gehalten:

Naval Research Laboratory, Software Engineering section of CHACS, Washington DC, USA, 17. Dez. 1997.

Univ. of Ottawa, School of Information Technology and Engineering, Ontario, Kanada, 5. März 1998.

Bell Labs, Software Production Research Dept., Naperville IL, USA, 18. Aug. 1998.

Während des Projektzeitraumes wirkte der Projektbearbeiter im *Programmkomitee* für den internationalen Estelle-Workshop mit, der sich mit allen Aktivitäten rund um die formale Beschreibungstechnik Estelle befaßt.

3 Ausblick

Die erarbeiteten Ideen zur Methodik der Erweiterung und Änderung einer Anforderungsspezifikation bedürfen einer weiteren Präzisierung. Insbesondere konnte im Projektzeitraum nur eine Skizze einer Semantik für die Komposition von Leistungsmerkmalen entwickelt werden. Nach dem Ende des Projektes begann der Projektbearbeiter an der Universität Oldenburg in der Arbeitsgruppe von Prof. Olderog mit der Fortsetzung dieser Arbeit. Es ist geplant, die Methodik formal zu unterstützen, indem auf einen bereits fertig ausgearbeiteten Verfeinerungsansatz zurückgegriffen wird. Hierbei handelt es sich um die in Oldenburg entwickelte Sprache CSP-OZ, die ein ähnliches semantisches Modell wie die Funktionale Dokumentation besitzt, und die zusätzlich die formalen Verfeinerungsbegriffe für Daten und Kontrolle von Object-Z bzw. CSP mitbringt. Auch der Formalismus der

Parnas-Tabellen soll weiter verwendet werden, da er den Modularisierungsgedanken besonders betont und da diese Notation gut zur Kommunikation mit technisch nur allgemein gebildeten Anwendern geeignet ist. (Z-basierte Sprachen dagegen erfordern ein umfangreiches mathematisches Handwerkszeug.) Die im Projekt entwickelte Anforderungsspezifikation der Fallstudie soll weiter genutzt werden, indem die Beschreibungsform geeignet angepaßt wird. Mit der anvisierten Methodik soll dann der Vorschlag für eine Architektur von Telefonvermittlungssystemen weiter ausgearbeitet werden. Auch die Möglichkeit der Migration von aktuellen Architekturen wie zum Beispiel dem Intelligenzen Netzwerk soll untersucht und beachtet werden.

4 Veröffentlichungen

Im Zusammenhang mit dem Projekt entstanden folgende Veröffentlichungen:

- [1] J. BREDEREKE. Requirements specification and design of a simplified telephone network by functional documentation. CRL Report 367, McMaster University, Hamilton, Ontario, Kanada (Dez. 1998). 86 Seiten.
- [2] J. BREDEREKE. Avoiding feature interactions in the users' interface. In K. KIMBLER UND L. G. BOUMA (Hrsg.), "Feature Interactions in Telecommunications and Software Systems V", S. 305–317. IOS Press, Amsterdam (Sep. 1998).
- [3] J. BREDEREKE. Maintaining telephone switching system requirements. Draft for the Dagstuhl seminar 99071 software engineering research and education: seeking a new agenda (Okt. 1998).

Diese Arbeiten sind dem Abschlußbericht beigelegt. Es ist geplant, den Teil der Ergebnisse, der bisher nur im technischen Bericht veröffentlicht ist, in Form von weiteren Konferenzbeiträgen im Jahre 1999 zu publizieren.

Während des Projektzeitraumes, aber nicht unbedingt in direkter Beziehung zum Projekt, entstanden die folgenden Veröffentlichungen:

- [4] R. GOTZHEIN UND J. BREDEREKE. Editorial: Trends in formal description techniques. *Comp. Networks and ISDN Syst.* **30**(9–10), 863–864 (Mai 1998).
- [5] J. BREDEREKE. Specification style and efficiency in Estelle. In S. BUDKOWSKI, S. FISCHER UND R. GOTZHEIN (Hrsg.), "Proc. of the 1st Int'l. Workshop on the Formal Description Technique Estelle – Estelle'98", S. 165–186, Evry, Frankreich (2. Nov. 1998). Institut National des Télécommunications.