

CAST: Ein System zur Entwicklung Computer-unterstützter Büroabläufe

J. Behrmann-Poitiers, M. Elixmann, Hamburg

Zusammenfassung

Ziel des MICADO-Projektes (Methods for Intelligent CAsE Design in Offices) ist es, Methoden auszuarbeiten zur Entwicklung Computer-unterstützter, kooperativer Arbeitsabläufe, wie sie im Bereich der Fall- oder Sachbearbeitung auftreten. Im Mittelpunkt des Projektes steht dabei die Spezifikation solcher Abläufe als Sachbearbeitungspläne. Diese Pläne sind die Grundlage eines Computer-unterstützten Bürosystems, das durch Interpretation der Pläne die Arbeitsschritte einer Sachbearbeitung weitgehend vorbereitet, alternative Bearbeitungsformen vorschlägt, über den Ablaufstand Buch führt und die notwendige Kooperation mehrerer Bearbeiter koordiniert.

Die vorliegende Arbeit stellt das Plan-Entwicklungswerkzeug CAST (Case plan Animation and Specification Tool) vor, mit dem Organisationsfachleute in Zusammenarbeit mit erfahrenen Sachbearbeitern Pläne Aufgaben-orientiert direkt am Bildschirm erstellen, verändern und animieren können.

Computer-unterstützte Sachbearbeitung im Büro

Computer-Unterstützung im Büro beschränkt sich heute noch weitgehend auf einzelne, isolierte Tätigkeiten wie Textverarbeitung und Archivierung von Dokumenten und Daten. Wünschenswert wäre eine weitergehende, umfassende Unterstützung ganzer Büroabläufe, einschließlich der Kommunikation und Kooperation zwischen Mitarbeitern [Bracchi, Pernici 84]. Dies gilt insbesondere für die strukturierten Routineaufgaben im Bereich der Fall- oder Sachbearbeitung.

Der Routinecharakter der Sachbearbeitung führt dazu, daß hier „nach Programm“ gearbeitet wird; die Arbeitsschritte einer Bearbeitung sind bekannt und durch Formulare stark vorgeprägt; Sachkompetenz und Verantwortlichkeit – und damit Kommunikation und Kooperation zwischen Mitarbeitern – sind klar geregelt [Hogg, Nierstrasz, Tschritzis 85].

Computer-unterstützte Sachbearbeitung geht von diesem Routinecharakter aus. Der Arbeitsablauf der Fälle eines Sachgebiets wird analysiert, erfaßt und als Computer-Anwendung so realisiert, daß der Gesamtvorgang an den entsprechenden Bildschirmarbeitsplätzen vollständig abgewickelt werden kann [Woo, Lochovsky 86]. Das Computer-System bereitet die Arbeitsschritte weitgehend vor, schlägt mögliche alter-

native Bearbeitungsformen vor, schreibt den Ablaufstand fort und koordiniert die Zusammenarbeit der Beteiligten [Kreifelts, Woetzel 86]. Eine solche Computer-Unterstützung kann die Sachbearbeitung übersichtlicher, weniger fehlerträchtig und effizienter gestalten.

Die Realisierungsmöglichkeiten für Computer-unterstützte Sachbearbeitungssysteme nach herkömmlichen Programmierverfahren sind bis jetzt unbefriedigend: Analyse- und Entwicklungsphasen dauern zu lange; das Ergebnis ist unübersichtlich und schwer anpaßbar an die sich häufig ändernden Anforderungen [Bullinger, Niemeier 87].

Bessere Verfahren zur Entwicklung solcher Systeme sind nötig. „Besser“ heißt, daß dabei die Phasen der Analyse, Spezifikation und Entwicklung zeitlich überlappend und nicht streng hintereinander ablaufen können. Zu diesem Zweck sollte der „programmierte Ablaufplan“ einer Sachbearbeitung explizit, in problemnaher Formulierung und übersichtlich dargestellt vorliegen, damit erfahrene Sachbearbeiter schon an der Planerstellung direkt beteiligt werden können.

Außerdem sollte ein einfach zu bedienendes Werkzeug zur Planentwicklung vorhanden sein, mit dem Pläne am Bildschirm interaktiv erstellt, verändert und probeweise simuliert werden können. Am Ende der Planentwicklung sollte aus der Planbeschreibung das operationale Sachbearbeitungssystem automatisch erzeugbar sein.

Im folgenden Abschnitt beschreiben wir das Konzept für Sachbearbeitungspläne, das im Rahmen des MICADO-Projektes (Methods for Intelligent CAse Design in Offices) entwickelt wurde. Anschließend geben wir eine Übersicht über die Entwicklungsprinzipien und den Aufbau des Plan-Spezifikationswerkzeuges CAST (Case plan Animation and Specification Tool), mit dem solche Pläne bearbeitet und simuliert werden können.

Sachbearbeitungspläne

Ein Sachbearbeitungsplan beschreibt die einzelnen Schritte der Bearbeitung einer Klasse von thematisch ähnlichen Fällen. Dabei können die Handlungsabläufe durchaus von Fall zu Fall verschieden sein; z. B. kann die Entscheidung, ob ein Antrag eines Klienten angenommen oder abgelehnt wird, unterschiedliche Nachfolgeaktionen bedingen. Da zudem einzelne Aktionen zeitlich voneinander unabhängig sein können, beinhaltet ein Plan neben sequentiell abhängigen und alternativen auch parallele Aktionen.

Die einzelnen Tätigkeiten lassen sich darstellen als Bearbeitung von Formularen oder anderen Bürodokumenten, die in der Regel zumindest teilstrukturiert sind [Malone 87].

Im folgenden werden die verschiedenen Dokumente vereinfachend Formulare genannt. Der Bearbeitungsstand eines Falles ergibt sich aus der Art und dem Inhalt der zu diesem Fall vorliegenden Formulare („Aktenlage“). Der Handlungsablauf läßt sich somit durch Regeln beschreiben, die in Abhängigkeit von der Aktenlage festlegen, wie der Fall weiter zu bearbeiten ist.

Ein MICADO-Plan wird durch einen Verbund aus Arbeitsschritten und Formularen beschrieben (Bild 1). Die einzelnen Arbeitsschritte werden durch Regeln entsprechend

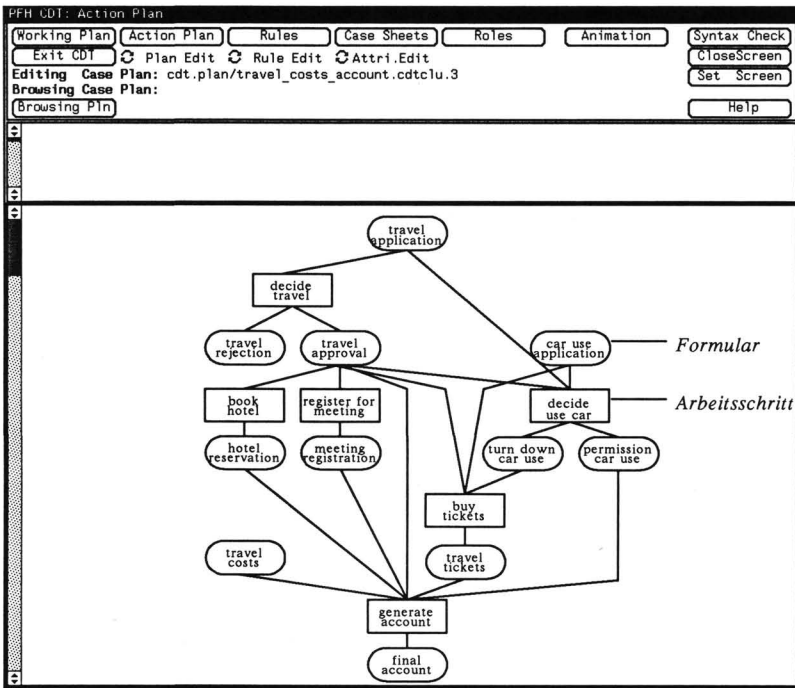


Bild 1: Die Plan-Übersicht

dem Produktions-System Paradigma spezifiziert [Nilsson 82]. [Zisman 78]. Eine Regel kann aus verschiedenen Regelvarianten bestehen, die jeweils verschiedene Situationen beschreiben, in denen dieser Arbeitsschritt durchgeführt werden kann (Bild 2).

Jede Variante besteht aus einem Bedingungsteil und einem Aktionsteil. In dem Bedingungsteil werden Anforderungen an die Akte bezüglich des Vorhandenseins von Formularen und deren Einträge spezifiziert. Im Aktionsteil werden alternative Aktionsvarianten definiert, zwischen denen der Sachbearbeiter später auswählen kann. Jede Aktionsvariante legt die Formulare fest, die während der Aktion ausgefüllt oder

C-Rule: decide travel	
Rulevariant: decide travel	Role: head of department
Conditionpart Yes: travel_application TA No: travel_rejection TR No: travel_approval TP where: TA.name = TR.name where: TA.name = TP.name	Actionpart Actionvariant: approval Produce: travel_approval Delete: Update: Actionvariant: rejection Produce: travel rejection Produce: travel_rejection name = TA.name destination = TA.destination begin_date = TA.begin_date rejection_reason =
Rulevariant: redecide travel	Role: head of department

Bild 2: Das Regelformular

verändert werden sollen. Der Planersteller kann die Werte der Datenkomponente der zu erzeugenden Formulare an die Daten der existierenden Formulare des Bedingungsteils einer Regelvariante anbinden, so daß sie bei der Fallbearbeitung automatisch eingetragen werden können. Die verschiedenen Regelvarianten haben jeweils einen eigenen Aktionsteil, um sowohl eine der speziellen Situation angepaßte Formularbearbeitung als auch ein eventuell später mögliches Abändern eines Formulars in derselben Regel beschreiben zu können.

Jedem Arbeitsschritt ist eine Rolle zugeordnet. Die Rolle beschreibt die für die Durchführung kompetente oder verantwortliche Mitarbeiterklasse. Die spezielle System-Rolle wird bei den Arbeitsschritten angewandt, in denen Formulare ohne Einwirkung des Sachbearbeiters erstellt werden. Bei der Bearbeitung eines konkreten Falles werden einzelne Mitarbeiter den verschiedenen Rollen zugewiesen.

Eine solche Planbeschreibung wird automatisch in ein operationales Programm umgewandelt, das sowohl die Unterstützung des Sachbearbeiters als auch die Zugriffe auf die Datenbank steuert.

Ein graphisches Werkzeug zur Planerstellung

Das Plan-Spezifikationswerkzeug CAST besteht im wesentlichen aus fünf Komponenten. Sie dienen der Darstellung und Spezifikation der verschiedenen Bestandteile eines

Sachbearbeitungsplanes, der Integration existierender Plankomponenten in neue Pläne und der Animation einer Planbeschreibung (Bild 3).

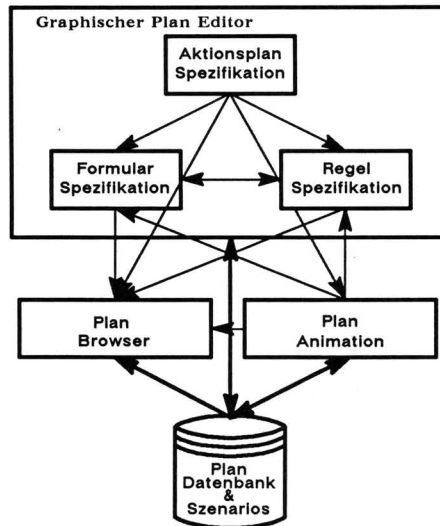


Bild 3: Die CAST Komponenten

Ein wesentlicher Aspekt eines solchen Werkzeuges betrifft die Präsentation der Planobjekte und die Funktionalität des Systems [Rohr 87]. Da sich die Benutzergruppe nicht nur aus Computerspezialisten zusammensetzt, sondern auch erfahrene Sachbearbeiter an der Plankonzeption mitwirken sollen, ist insbesondere die Unterstützung des Design-Prozesses durch das Werkzeug von Bedeutung. Aus diesem Grund wurde bei der Entwicklung des Spezifikationswerkzeuges CAST besonderer Wert gelegt auf moderne Hilfsmittel wie Pop-Up-Menüs, Bildschirm-Knöpfe und direkte Manipulation der Objekte mit Hilfe der Maus anstelle von textueller Kommando-Eingabe [Goldberg 84].

Um den Anforderungen an eine dem persönlichen Arbeitsstil angepaßte Umgebung zur Spezifikation von Plänen zu genügen, bietet CAST eine Vielzahl verschiedener Sichten auf die Pläne, die auf einer einheitlichen, logischen Planrepräsentation basieren. So ist es dem Benutzer zum Beispiel möglich, Pläne deskriptiv in einer Formularumgebung oder prozedural entsprechend einem graphisch-visuellen Formalismus zu beschreiben [Harel 88].

Allen Komponenten von CAST ist das Grundkonzept gemeinsam, dem Benutzer so viel an Information über den Plan wie vorhanden zu geben und ihm so wenig Schreibarbeit wie möglich zu überlassen. So können Inkonsistenzen vermieden werden. Zum Beispiel wird beim Eintragen eines Formularnamens in eine Regel über ein Menü die Liste der

existierenden Formulare angeboten, aus der man direkt mit der Maus auswählen kann. Zudem handelt es sich um Kontext-sensitive Syntax-gesteuerte Editoren [Neal 87], so daß der Benutzer von der Last der Erlernens einer Programmiersprache befreit ist.

Die Aktionsplan-Komponente von CAST stellt das Verhältnis zwischen Arbeitsschritten und Formularen graphisch dar (Bild 1). Aus einem solchen Aktionsplan können wichtige Teile herausgefiltert werden, z. B. spezielle Beziehungen zwischen Arbeitsschritten und Formularen. Zudem werden lokale Sichten auf spezielle Bestandteile und deren Einbindung in den Plan ermöglicht (Bild 4). Die Standarddarstellung

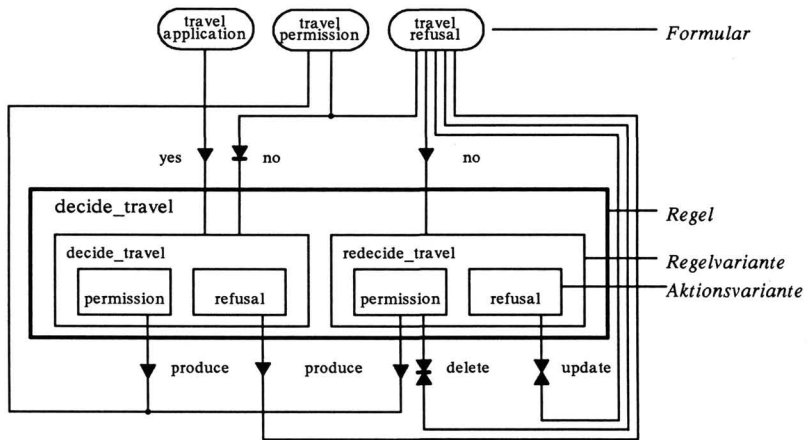


Bild 4: Aktionsplan Detailsicht

zeigt für einen Sachbearbeitungsplan die notwendigen Daten als Voraussetzung der Arbeitsschritte mit den jeweils zu produzierenden Formularen. Dieser Aktionsplan kann interaktiv durch Selektion mit der Maus über Menüs und Pop-Up-Spezifikationsfenster ediert werden.

Die zweite Komponente dient der Spezifizierung der Formulare. Der Benutzer bestimmt dazu die Datenfelder und den Typ der einzutragenden Daten und unabhängig davon interaktiv das Layout der Formulare (WYSIWYG: What You See Is What You Get). Zur schnellen Integration bestehender Formulare in ein elektronisches Sachbearbeitungssystem ist es möglich, das Layout der Datenfelder auf das Bild eines eingescannten Papierformulars zu legen (Bild 5). Das Erlernen einer Formular-Beschreibungssprache ist unnötig. Aus der Formularbeschreibung wird automatisch ein entsprechendes Datenbank-Schema erzeugt.

In der dritten Komponente werden die einzelnen Arbeitsschritte durch Regeln spezifiziert. Dies kann durch Eintragen in ein Regel-Formular geschehen, das genau

Enclosure to travel costs sheet	
Name: <input type="text"/>	Dept: <input type="text"/> Phone No: <input type="text"/>
Begin of travel: <input type="text"/>	Time: <input type="text"/>
End of travel: <input type="text"/>	Time: <input type="text"/>
If returning from foreign countries: border crossing resp. 1st landing in the Federal Republic before 14.00 p.m.? Yes <input type="checkbox"/> No <input type="checkbox"/>	
Destination (s): <input type="text"/>	
1. Travel to training Yes <input type="checkbox"/> No <input type="checkbox"/>	
2. Travel to plant or organization of: Philips <input type="checkbox"/> Others <input type="checkbox"/>	
For combined trips: assumed splitting of costs Philips <input type="text"/> %, Others <input type="text"/> %	
3. Use of private car, id. no.: inland: <input type="text"/> km abroad: <input type="text"/> km	
4. No. of invitations: <input type="text"/>	
5. Car renting costs: enclosed <input type="checkbox"/> follows <input type="checkbox"/> none <input type="checkbox"/>	
6. If costs are cleared by organizer	
- Overnight accommodations - No. of nights: <input type="text"/>	
- Boardings - No. of days: <input type="text"/>	
Pure travel time - one way - to meeting place more than 3 hours: Yes <input type="checkbox"/> No <input type="checkbox"/>	
7. Have conference fees been paid before travel: Yes <input type="checkbox"/> No <input type="checkbox"/>	
8. Days of vacation during the trip: From <input type="text"/> Till <input type="text"/>	
9. Return of unused tickets: Yes <input type="checkbox"/> No <input type="checkbox"/>	
Kind of ticket: <input type="text"/>	No.: <input type="text"/> Unused travel line: <input type="text"/>

Assign Data Objects
- New Data Object -
Name
Department
Phone Number
Begin_Travel
End_travel
Begin_Time
End_Time
Foreign_Return
Destination
Training
Philips
Private_Car_Inland
Private_Car_Abroad
Invitations
Car_Renting
Cleared_Costs_Inland
Cleared_Costs_Abroad
Travel_Time
Conference_Fee
Vacation_From
Vacation_To
Unused Tickets
Kind_of_Ticket
Ticket_Number
Travel_Line

Bild 5: Formular-Layout

die Struktur der Regeln widerspiegelt und sich dynamisch dem Umfang einer Regel anpaßt (Bild 2).

Eine vierte Komponente dient der Integration existierender Pläne. In einer Fensterorientierten Umgebung sieht man diese Übersichtskomponente parallel neben dem eigentlichen Editor. Sie ermöglicht, Regeln und Formulare oder ganze Teilpläne aus einem bestehenden Plan in einen neuen zu kopieren.

Die fünfte Komponente dient der Simulation komplett oder teilweise spezifizierter Pläne. Diese arbeitet auf derselben logischen Plan-Repräsentation wie der Plan-Editor und deshalb ohne vorherige Übersetzung durch Interpretation des soweit festgelegten Planes. Der Benutzer kann dazu in der ihm bekannten graphischen Übersichtsdarstellung einzelne Formulare als existent kennzeichnen. Ausgehend von diesem Szenarium wird dann die Fallbehandlung gemäß dem erstellten Plan simuliert. Mögliche Arbeitsschritte werden angezeigt und, nach Auswahl einer Aktionsvariante durch den Benutzer, der neue Zustand des Falles bestimmt und dargestellt. Da zu jedem Zeitpunkt ein Edieren des Planes erfolgen kann und die Simulation mit den veränderten Regeln fortgesetzt wird, ist ein direkter Übergang von der Spezifikation zur Validierung durch Simulation und umgekehrt möglich.

Durch die Integration der verschiedenen eigenständigen Komponenten von CAST ist es gelungen, dem Benutzer die größtmögliche Freiheit bei der Spezifikation von Sachbearbeitungspläne zu bieten. So ist es zum Beispiel möglich, einen Plan schon zu animieren, wenn lediglich die Namen der Formulare bekannt sind, ohne Definition der Datenfelder. Andererseits können die Datenfelder der Formulare bzw. deren Layout unabhängig von den Regeln definiert werden.

CAST unterstützt somit nicht nur den individuellen Arbeitsstil des Benutzers. Insbesondere wird eine arbeitsteilige Spezifikation der Sachbearbeitungspläne auf diese Weise ermöglicht. So können verschiedene Mitarbeiter zuständig sein für die regelbasierte Struktur eines Planes, die Datenfelder der einzelnen Formulare und deren eventuelle automatische Berechnung, bzw. die Festlegung des Layout dieser Datenfelder für den Bildschirm des Sachbearbeiters oder auf Papier für Briefe etc.

Die enge Verzahnung der einzelnen Komponenten (Bild 3) erlaubt zu jedem Zeitpunkt den Zugriff auf alle relevanten Informationen über die einzelnen Planbestandteile. Zudem ist es möglich sich an Hand dieser in Menüs präsentierten Information interaktiv durch einen Plan zu navigieren und dessen Struktur zu erschließen.

Weitergehende Untersuchungen

Die derzeit laufenden Weiterentwicklungen von CAST gelten unter anderem der strukturierten Spezifikation von Sachbearbeitungsplänen: Einzelne Arbeitsschritte können selbst wieder durch einen Sachbearbeitungsplan näher beschrieben werden. Dieses Hauptplan/Unterplan-Schema kann über beliebig viele Stufen gehen. Durch diese modulare Beschreibung wird erreicht, daß einerseits die Detailspezifikation ausgesetzt bzw. delegiert werden kann, andererseits aber auch einzelne Unterpläne mehrfach in verschiedenen Sachbearbeitungsplänen zur Anwendung kommen können.

Auch für solche Arbeitsschritte existiert für den Hauptplan eine Regelbeschreibung, die die gemeinsamen Bedingungen für die Durchführbarkeit der im Unterplan enthaltenen Arbeitsschritte festlegt und die für den Hauptplan relevanten bearbeiteten Formulare spezifiziert. Auf diese Weise ist eine saubere Schnittstelle zwischen den Plan-Ebenen geschaffen, die das globale Verhalten eines Unterplanes beschreibt und den Einstieg in diese Ebene steuert.

Unter Berücksichtigung der modularen Planbeschreibung und der Möglichkeit der Animation nur teilweise spezifizierter Pläne bzw. einzelner Unterpläne ist es möglich, in CAST ein Konzept für „structured prototyping“ zu integrieren [Shoval, Pliskin 88].

Eine weitere Untersuchung gilt der Repräsentation von Plänen in einer Wissensbank. Ausgehend von einer solchen Darstellung sollen dann Pläne automatisch um Teilpläne und globale Konzepte, z. B. der Überprüfung durch einen Vorgesetzten, erweitert werden können. Letztendliches Ziel wäre die stärker automatisierte Erstellung neuer Pläne aus globalem Wissen über Pläne und aus speziellem Wissen über den Anwendungsbereich.

Die Integration einer mehr Beispiel-orientierten Spezifikation von Plänen ist ebenfalls Bestandteil der Forschung (programming-by-example) [Spenke, Beilken 87]. Das heißt, daß der Planersteller dem System verschiedene Vorgehensweisen bei der Fallbearbeitung vormacht, also für einen bestimmten Zustand angibt, welche voneinander unabhängigen Arbeitsschritte durchgeführt werden können, und welche Formulare dabei erstellt bzw. verändert werden müssen. Ebenso gehört dazu, wie sich einzelne Datenwerte dabei aus anderen berechnen lassen. Aus diesen beispielhaften Angaben sollen dann direkt die Regeln abgeleitet werden.

[Bracchi, Pernici 84]

Bracchi, G. and B. Pernici: The design requirements of office systems. ACM Transactions on Office Information Systems, Vol. 2, No. 2, April 1984, pp. 151 – 170.

[Bullinger, Niemeier 87]

Bullinger, H.-J. and J. Niemeier: Analyse und Gestaltungsmethoden im Bürobereich.
GI 17. Jahrestagung 1987: Computerintegrierter Arbeitsplatz im Büro, Springer Informatik Fachberichte 156, Berlin 1987, pp. 36 – 53.

[Goldberg 84]

Goldberg, A.: Smalltalk-80: The interactive programming environment. Addison-Wesley, Reading, Massachusetts 1984.

[Harel 88]

Harel, D.: On visual formalisms. Communications of the ACM, Vol. 31, No. 5, May 1988, pp. 514 – 529.

[Hogg, Nierstrasz, Tsichritzis 85]

Hogg, J., Nierstrasz, O. M., and D. Tsichritzis: Office procedures. In: Tsichritzis, Dionysios (Ed.): Office automation, Springer Verlag, Topics in Information Systems, Berlin Heidelberg 1985, pp. 137 – 165.

[Kreifelts, Woetzel 86]

Kreifelts, Th., and G. Woetzel: Distributed and error handling in an office procedure system. Proc. IFIP WG 8.4. Working Conference: Office systems: Methods and tools, Pisa 1986, pp. 197 – 208.

[Malone 87]

Malone, T. W. et al.: Semistructured messages are surprisingly useful for computer-supported coordination.
ACM Transactions in Office Information Systems, Vol. 5, No. 2, April 1987, pp. 115 – 131.

[Neal 87]

Neal, L. R.: Cognition-sensitive design and user modeling for syntax-directed editors.
CHI + GI 87: Human factors in computing systems and graphics interface. Toronto, April 1987, SIGCHI Bulletin, ACM, New York 1987, pp. 99 – 102.

[Nilsson 82]

Nilsson, N. J.: Principles of artificial intelligence.
Springer Verlag, Berlin Heidelberg New York 1982.

[Rohr 87]

Rohr, G.: How people comprehend unknown system structures: Conceptual primitives in systems' surface presentations.
5th Int. Workshop in Inf. and Psy. 1986: Visualization in programming, Springer Lecture Notes 282, Berlin 1987, pp. 89 – 105.

[Shoval, Pliskin 88]

Shoval, P., and N. Pliskin: Structured prototyping: Integrating prototyping into structured system development.
Information & Management, Vol. 14, No. 1, January 1988, pp. 19 – 30.

[Spenke, Beilken 87]

Spenke, M., and C. Beilken: Programming-by-example in einer formularorientierten Umgebung.
Expertensystem '87, Nürnberg 1987, Berichte des German Chapter of the ACM, Vol. 28, Teubner Verlag, Stuttgart 1987, pp. 19 – 38.

[Woo, Lochovsky 86]

Woo, Carson C. and Frederick H. Lochovsky: Integrating procedure-automation and problem-solving approaches to supporting office work.
IFIP WG 8.4 Working Conference: Office systems: Methods and tools, Helsinki 1985, North-Holland, Amsterdam 1986, pp. 15 – 30.

[Zisman 78]

Zisman, Michael D.: Use of production systems for modeling asynchronous, concurrent processes.
Waterman, D. and F. Hayes-Roth, (Eds.): Pattern directed inference systems, Academic Press, New York 1978, pp. 53 – 68.

Dr. Jens Behrmann-Poitiers
Philips GmbH
Forschungslaboratorium Hamburg
Vogt-Kölln-Straße 30
D-2000 Hamburg 54

Dr. Martin Elixmann
Philips GmbH
Forschungslaboratorium Hamburg
Vogt-Kölln-Straße 30
D-2000 Hamburg 54