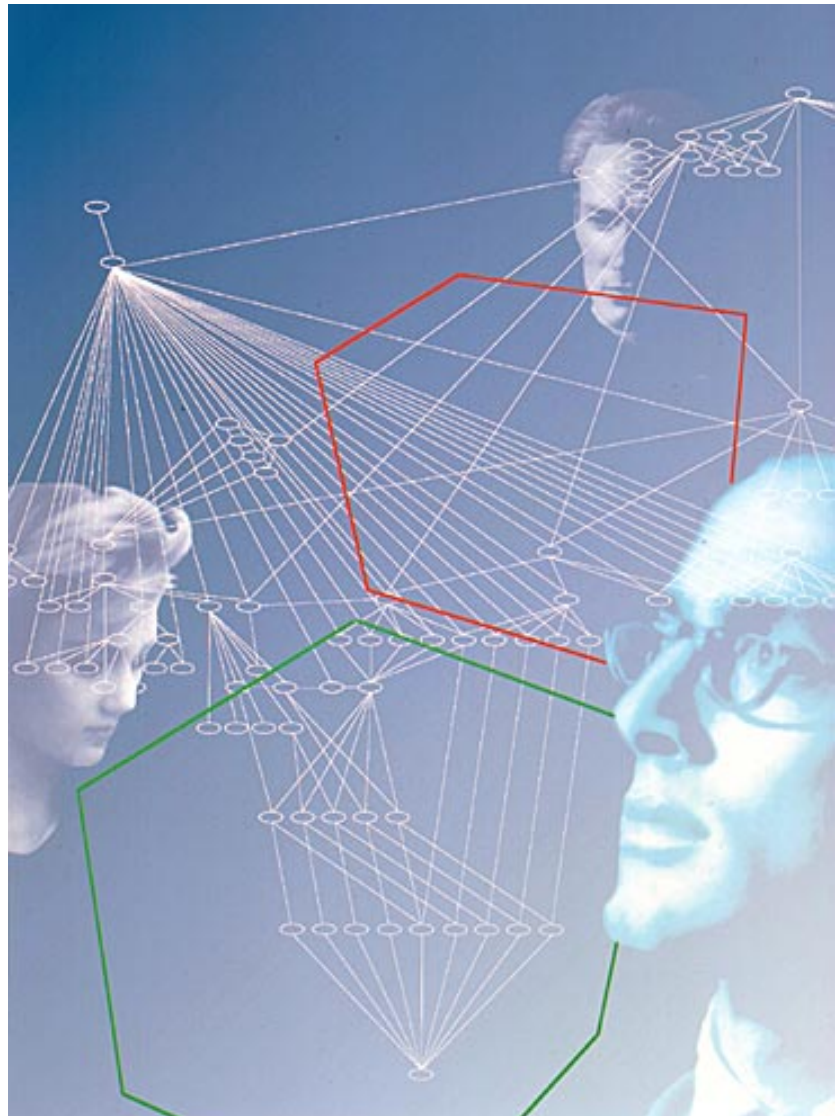




Fraunhofer-Einrichtung
für Software-
und Systemtechnik



Tätigkeitsbericht 1993

Fraunhofer-Gesellschaft

**Institut für Software-
und Systemtechnik**

Tätigkeitsbericht 1993

**Fraunhofer-Institut
für Software- und
Systemtechnik (ISST)**

Institutsleiter
Prof. Dr. Herbert Weber

Institutsteil Berlin:

Kurstr. 33
D-10117 Berlin

Tel.: (030) 20372-296/297
Fax: (030) 20372-207

Institutsteil Dortmund:

Postfach 520130
D-44207 Dortmund

Tel.: (0231) 9700-700/701
Fax: (0231) 9700-799

Inhaltsverzeichnis

	Vorwort.....	1
1	Das Institut für Software- und Systemtechnik.....	3
1.1	Aufgaben und Ziele	5
1.2	Organisation des Instituts und Fachbereiche.....	6
1.3	Leistungsangebot	9
1.4	Kuratorium.....	12
1.5	Personal und Sachausstattung	13
1.6	Haushalt und Finanzierung	15
2	Arbeiten der Fachbereiche	17
2.1	Projektüberblick.....	19
2.1.1	Öffentlich geförderte Projekte.....	19
2.1.2	Auftragsprojekte	20
2.1.3	Interne Projekte	20
2.2	Werkzeuggestützte Softwareentwicklung.....	22
2.2.1	Objektbasierte Softwareentwicklung	24
2.2.2	Lisp-Entwicklung	27
2.2.3	Re-Engineering	29
2.2.4	CASE-Einführung.....	31
2.2.5	CASE-Kompetenzzentrum	35
2.2.6	Software-Qualitätssicherung	36
2.3	Integrierte Software-Infrastrukturen.....	38
2.3.1	Software Factory Engineering	41
2.3.2	Daten- und Objektmanagement	43
2.3.3	Multimedia-Systeme.....	45
2.3.4	System- und Netzwerkmanagement	47
2.3.5	Computergestützte Kooperation.....	54
2.4	Prozeß-Engineering	56
2.4.1	Prozeßmanagement-Werkzeuge	58
2.4.2	Management kundenspezifischer Geschäftsprozesse.....	59
2.5	Aus- und Weiterbildung	63

Inhaltsverzeichnis

3	Öffentlichkeitsarbeit.....	65
3.1	ISST-Eröffnungsveranstaltung	67
3.2	Veröffentlichungen	69
3.2.1	Beiträge in (Fach-) Zeitschriften und Büchern.....	69
3.2.2	Vorträge auf Konferenzen und Tagungen	70
3.2.3	Sonstige Vorträge.....	72
3.2.4	Technische Berichte und Studien.....	73
3.3	Wissenschaftliche Arbeiten	76
3.3.1	Studien- und Diplomarbeiten.....	76
3.3.2	Dissertationen	77
3.4	Veranstaltungen des ISST	78
3.4.1	Seminare und Workshops	78
3.4.2	Kolloquia.....	78
3.4.3	Präsentationen	78
3.4.4	Sonstige Veranstaltungen	79
3.5	Mitarbeit in Gremien, Ausschüssen, Konferenz- und Programmkomitees.....	80
3.6	Vorlesungstätigkeit an Hochschulen	82
3.7	Teilnahme an Messen und Kongressen.....	83
3.8	Internationale Beziehungen.....	84
3.8.1	Internationale Industrie- und Wissenschaftskontakte	84
3.8.2	Internationale Besucher im ISST	85

Vorwort

Die *Fraunhofer-Einrichtung für Software- und Systemtechnik ISST* kann nach dem zweiten Jahr ihres Bestehens eine positive Bilanz ziehen. Dies ist umso erfreulicher, als unsere Ausgangsposition Anfang 1992 alles andere als einfach war. Galt es doch, die beiden Institutsteile in Berlin und Dortmund auf eine organisatorisch wie wissenschaftlich gemeinsame Linie einzuschwören. Dies ist uns dank eines intensiven Technologieaustausches und kontinuierlicher Abstimmungen gut gelungen. Auch die Zusammenarbeit mit unserem Partnerinstitut für „Computergestützte Informationssysteme (CIS)“ am Fachbereich Informatik der Technischen Universität Berlin gestaltete sich im vergangenen Geschäftsjahr erfolgreich. Gemeinsame Forschungskolloquia wurden vorbereitet. Mit Beginn des Sommersemesters 1994 werden sich ISST und CIS über laufende Arbeiten in regelmäßigem Turnus informieren.

Seinem hohen Anspruch, der Informationstechnologie durch marktnahe Forschung und Entwicklung neue Impulse zu geben, konnte das Institut auch im Rezessionsjahr 1993 gerecht werden. Mit der Resonanz bei unseren Geschäftspartnern waren wir - und dies betraf unsere gesamte Leistungspalette - überaus zufrieden. Instanzen der öffentlichen Verwaltung machten vom Angebot des ISST besonders intensiv Gebrauch. Ein Beispiel hierfür ist die Kooperation mit dem Landesamt für Informationstechnik (LIT) Berlin, das die Informations- und Kommunikationsstruktur der Berliner Verwaltung komplett reorganisieren wird. Bei dieser komplexen Aufgabe fungierten die Experten des ISST als Berater und Konstrukteure. Unser Ziel ist es, verstärkt auch Industrieunternehmen bei der Gestaltung moderner IuK-Strukturen zu unterstützen. Erste Aktivitäten in dieser Richtung wurden 1993 im Rahmen von Auftragsprojekten bereits unternommen.

Insgesamt lassen uns die bisherigen Erfahrungen optimistisch in die Zukunft blicken. In der gegenwärtigen Lage kommt der Informations-, speziell Softwaretechnologie eine strategische Funktion im internationalen Wettbewerb zu. Für unsere Wirtschaftsunternehmen ein zwingender Grund, sich rechtzeitig mit zukunftsweisender Kompetenz auszustatten.

Das ISST wird sich dafür auch 1994 im Rahmen seines Wissenstransfers mit aller Kraft einsetzen.



Prof. Dr. Herbert Weber

1

**Das Institut für
Software- und
Systemtechnik**

1.1 Aufgaben und Ziele

Die Fraunhofer-Einrichtung für Software- und Systemtechnik (ISST) mit ihren beiden Institutsteilen in Berlin und Dortmund ist eine Forschungseinrichtung der Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V. (FhG). Gegenwärtig unterhält die FhG 46 Forschungseinrichtungen mit mehr als 7500 Mitarbeitern. Ihr Forschungsvolumen betrug 1993 über 940 Millionen DM; davon erwirtschaftete sie mehr als drei Viertel aus Aufträgen der Industrie und öffentlichen Forschungsaufträgen. Der Sitz der Zentralverwaltung ist in München.

Das Institut für Software- und Systemtechnik wurde im Rahmen der Ausdehnung der Fraunhofer-Gesellschaft auf die neuen Bundesländer zum 1. Januar 1992 als selbständige Einrichtung in Berlin (mit einer Außenstelle in Dortmund) gegründet. Das ISST verstärkt die Informatikforschung innerhalb des Fachbereiches „Informationstechnik“ der FhG.

Es sieht seine Hauptaufgabe in der Erforschung und Entwicklung systematischer, ingenieurtechnischer Methoden, Techniken und Verfahren auf den Gebieten der Software- und Systemtechnik. Der Schwerpunkt liegt dabei auf der Planung und Entwicklung moderner Software-Systeme und langlebiger „informations- und kommunikationstechnischer Infrastrukturen“ für deren Einsatz in Industrie und öffentlicher Verwaltung.



**Das Gebäude des ISST
am Spittelmarkt in Berlin**

1.2 Organisation des Instituts und Fachbereiche

Das ISST besteht aus den Institutsteilen Berlin und Dortmund. An beiden Standorten erfolgt die wissenschaftliche Arbeit innerhalb thematisch spezifischer Fachbereiche (vgl. Organisation der Institutsteile in Abb. 1 und Abb. 2). Während im Gründungsjahr 1992 und noch zu Beginn des Jahres 1993 der Technologie-Transfer von Dortmund nach Berlin im Vordergrund stand, kristallisierten sich bis zum Jahresende standortspezifische Schwerpunkte heraus.

Der Bereich „Allgemeine Geschäftsführung“ (GF) existiert separat in Berlin und Dortmund mit jeweils einem eigenverantwortlichen Bereichsleiter. Die Verwaltung, die technische Infrastruktur, das Projektcontrolling, die Koordination der Akquisition sowie die Öffentlichkeitsarbeit des Institutes fielen 1993 in dessen Zuständigkeit.

Die wissenschaftlichen Fachbereiche sind weitgehend - wenngleich mit jeweils unterschiedlichen Schwerpunkten - an beiden Standorten etabliert. Damit ist sichergestellt, daß Kompetenz auf den wichtigen Gebieten der Software- und Systemtechnik sowohl in Berlin als auch in Dortmund vorhanden ist. Im einzelnen nahmen die Bereiche im Rahmen von Arbeitsgebieten (AG) folgende Aufgaben wahr:

Der Bereich „Werkzeuggestützte Softwareentwicklung“ (WSE) umfaßte die Arbeitsgebiete „Objektbasierte Softwareentwicklung“, „Lisp-Entwicklung“ und „Software-Re-Engineering“ (AG in Berlin) sowie „CASE-Einführung“, „CASE-Kompetenzzentrum“ und - als neu geschaffenes Tätigkeitsfeld - „Software-Qualitätssicherung“ (AG in Dortmund). Vor allem im Arbeitsgebiet „Objektbasierte Softwareentwicklung“ gab es eine enge Zusammenarbeit zwischen Berlin und Dortmund, da man in Berlin mit der objektbasierten Spezifikationsprache II Entwicklungsarbeiten fortsetzte, die 1985 an der Universität Dortmund (am ehemaligen Lehrstuhl von Prof. Weber) begonnen wurden.

Der Fachbereich „Integrierte Software-Infrastrukturen“ (ISIS) bearbeitete 1993 Projekte auf den Gebieten „Software-Fabrik-Engineering“, „Daten- und Objektmanagement“ (AG in Berlin und Dortmund) sowie „System- und Netzwerkmanagement“ und „Computergestützte Kooperation“ (AG in Berlin). Als neues Arbeitsfeld wurde in Dortmund „Multimedia-Systeme“ etabliert.

Die am Dortmunder Standort eingerichteten Fachbereiche „Prozeß-Engineering“ (mit den AG „Prozeß-Management-Werkzeuge“ und „Management kundenspezifischer Geschäftsprozesse“) und „Aus- und Weiterbildung“ (Schwerpunkt: Ausbildungsprogramm zum „European Professional Software Engineer“) konnten ihre Arbeit in 1993 in dem geplanten Umfang fortsetzen.

Durch die standortspezifischen Konzentrationen der Arbeiten haben die Verbindungen zwischen den Institutsteilen nicht gelitten. So wurden auch im Berichtsjahr gemeinsam Projekte bearbeitet, was eine enge Kooperation zwischen Berlin und Dortmund erforderte. Darüber hinaus hat diese Konzentration auch zu einer stärkeren Zusammenarbeit der einzelnen Fachbereiche am jeweiligen Standort geführt, so daß von einem Berlin- bzw. Dortmund-spezifischen Leistungsangebot gesprochen werden kann.

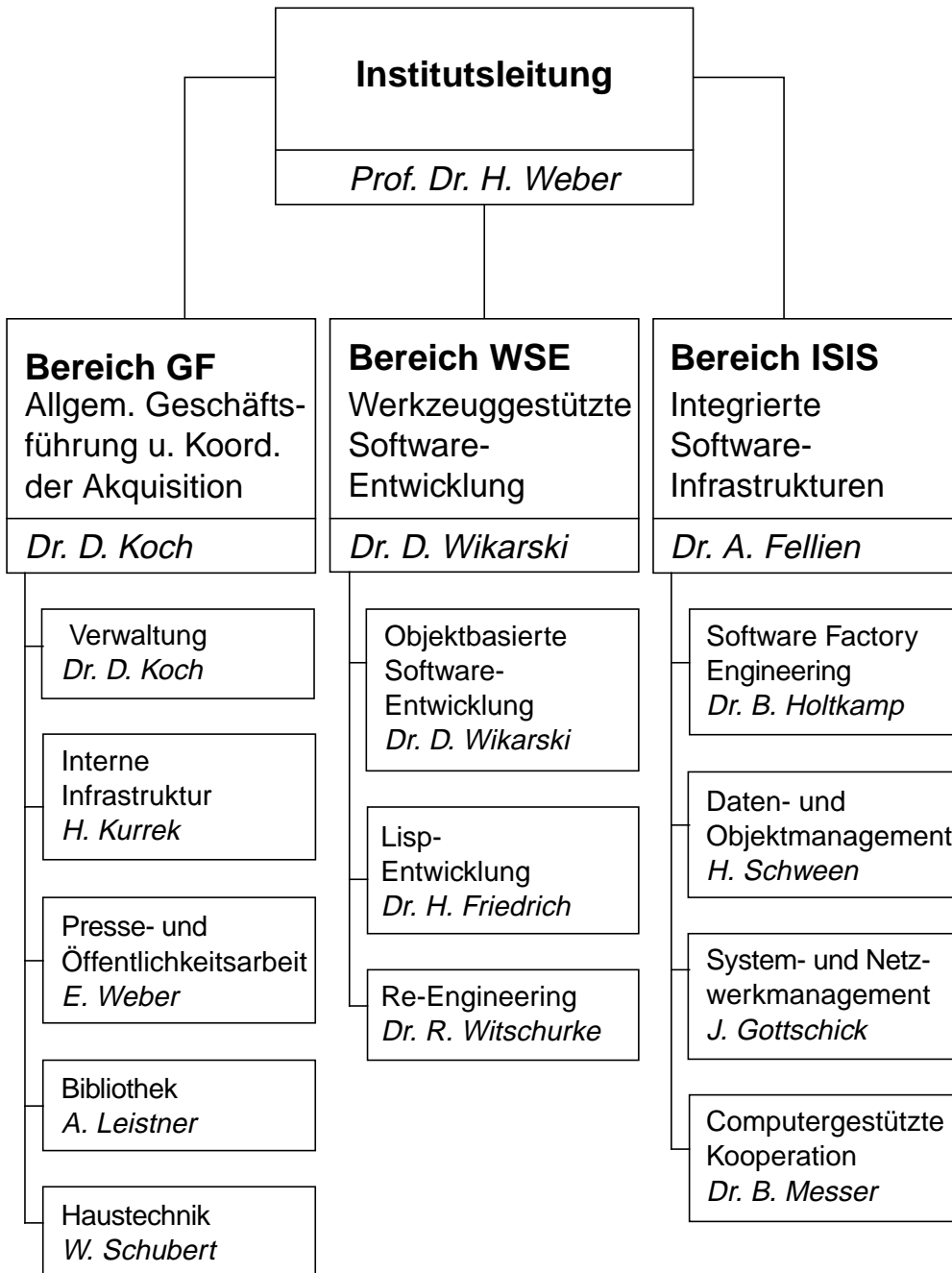


Abb. 1:
Organisation
Institutsteil
Berlin

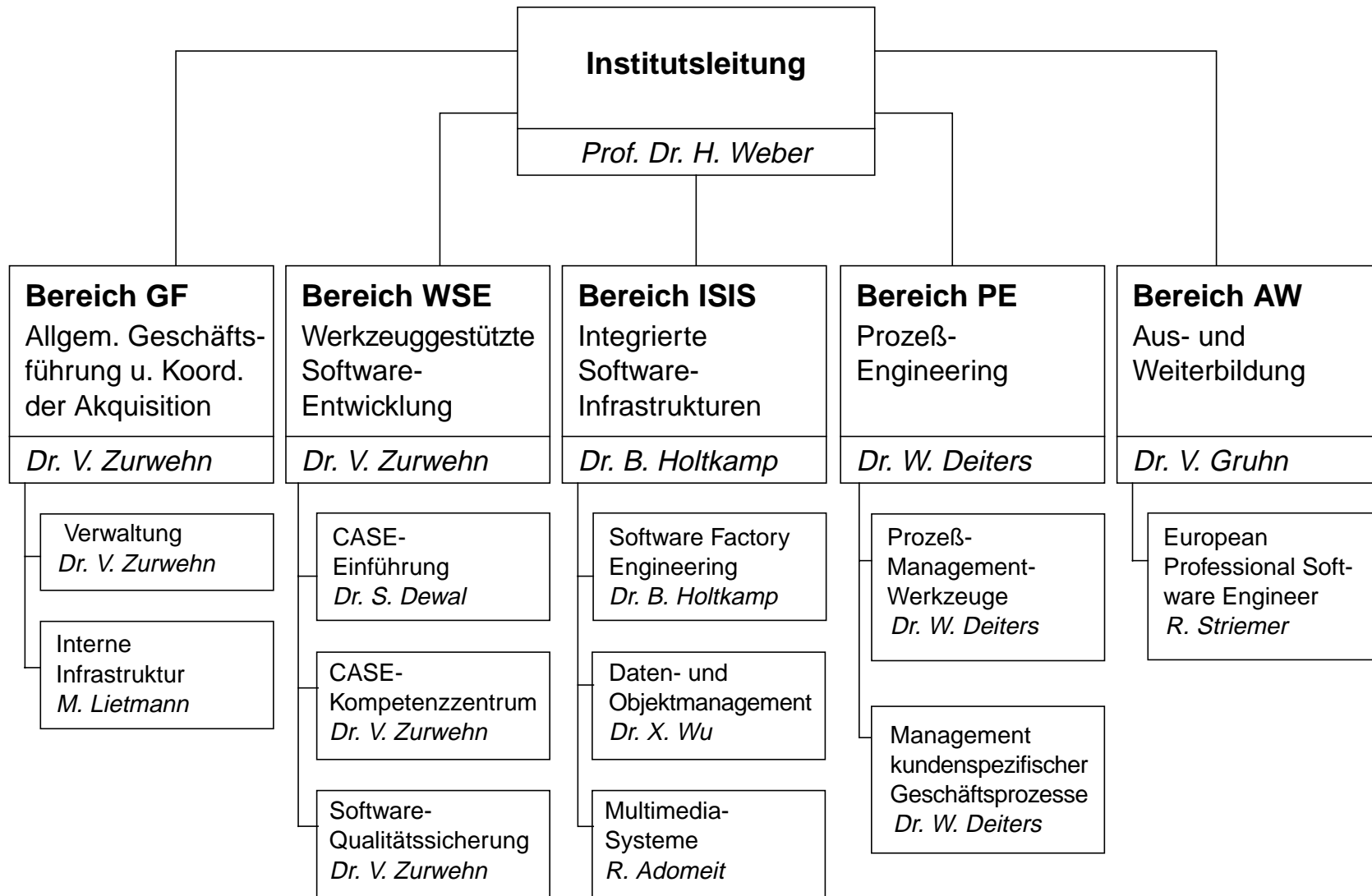


Abb. 2: Organisation Institutsteil Dortmund

1.3 Leistungsangebot

Das ISST arbeitet in den genannten Arbeitsbereichen als „Software-Bauhaus“ an innovativen Methoden, Techniken und Verfahren, um Software auf industrieller Grundlage entwickeln, pflegen, warten und kontinuierlich weiterentwickeln zu können. Sein breitgefächertes Dienstleistungsangebot reicht von der Erstellung von Gutachten, über Beratungs- und Unterstützungsleistungen bis hin zu strategischen Kooperationen zum Zwecke der Prototyp-Entwicklung. Mit diesem Servicepaket richtet sich das ISST an:

- Software- und Systemnutzer,
- Software- und Systementwickler und -integratoren
- Anwender von Systemen zur werkzeuggestützten Softwareentwicklung
- private Unternehmen und
- öffentliche Verwaltungen.

Im einzelnen bietet das ISST in seinen verschiedenen Fachbereichen folgende Dienstleistungen an:

Werkzeuggestützte Softwareentwicklung:

- Analyse von Methoden zur computergestützten Softwareentwicklung (CASE)
- Unterstützung bei der Entwicklung und dem gezielten, angepaßten Einsatz von Werkzeugen bei der Softwareentwicklung
- Unternehmensspezifische Auswahl von Methoden und Werkzeugen für die Softwareentwicklung
- Eignungsprüfung und Bewertung von kommerziellen Software-Entwicklungswerkzeugen und -umgebungen unter Berücksichtigung spezifischer Nutzeranforderungen
- Individuelle Produktberatung
- Methodenspezifische Werkzeugübersichten
- Beratung bei der Einführung einer ISO 9000-konformen Software-Qualitätssicherung
- „Software Factory Engineering“

Software-Re-Engineering:

- Entwicklung von Methoden und Techniken für die Renovierung bzw. Sanierung gealterter Software
- Konzeption unternehmensspezifischer Strategien zur Software-Sanierung
- Analyse von Schwachstellen in gealterten Software-Systemen
- Anpassung von Software an moderne Hard- und Software-Architekturen („Migration“)
- Methodische und technische Unterstützung von Unternehmen bei Re-Engineering-Projekten

Integrierte Software-Infrastrukturen:

- Entwicklung und Evaluierung von Integrationsstrategien und -technologien und deren Transfer in die industrielle Praxis
- Entwurf von Architekturen für Systeme und Infrastrukturen
- Konzepte und Verfahren zur Integration (Vernetzung) autonomer Systeme
- Realisierung von Infrastrukturen für Software-Fabriken zur industriellen Software-Fertigung
- Entwicklung von leistungsfähigen Kommunikationssystemen und multimedialen Anwendungen
- Beratung bei der Anwendung von formalen Spezifikationsmethoden in der Softwareentwicklung
- Konzeption und Einführung von Netzwerk- und Systemmanagement-Plattformen
- Entwicklung von Konzepten und Systemen:
 - zur koexistenten Datenhaltung und -verwaltung für kommerzielle und technische Anwendungen
 - zur rechnergestützten Dokumentenverwaltung und -bearbeitung
 - zur Unterstützung kooperativer Gruppenarbeit

Prozeß-Engineering:

- Management von Geschäftsprozessen (Workflow Management) durch Vorgangmodellierung
- Aufdeckung von Schwachstellen in betriebsinternen Abläufen durch Prozeßsimulation
- Dynamische Verbesserung von Prozessen
- Ressourcenplanung
- Gezielte Unterstützung bei der Vorgangsbearbeitung

Aus- und Weiterbildung:

- Konzeption und Durchführung von Schulungs- und Weiterbildungsveranstaltungen

Das ISST unter der Leitung von Prof. Dr. Herbert Weber verfügt durch seine erfahrenen Wissenschaftler in all diesen Bereichen über langjähriges Know-how in der Auftragsforschung für Industrie und öffentliche Verwaltung, in der Verbundforschung mit Industrieunternehmen im Rahmen diverser Forschungsprogramme (Beispiele: ESPRIT, COST) sowie in der internationalen Forschungskoooperation (z.B. EUREKA).

Das „Software-Bauhaus“ ISST ist keiner Technologie und keinem Produkt verpflichtet. Es räumt daher keinem bestimmten Produkt einen absoluten Vorrang ein. Dies hat den großen Vorteil, daß es gegenüber allen Partnern als *neutraler unvoreingenommener Berater* agieren kann. Das gilt uneingeschränkt auch dann, wenn das ISST Aufträge Dritter gemeinsam mit Hardwareherstellern,

Software- oder Systemhäusern bearbeitet. In solchen Partnerschaften bietet das ISST an, die Rolle des Architekten und Konstrukteurs für neue Systeme und Lösungen zu übernehmen.

Die neutrale Position des ISST ist wesentlich durch den gemeinnützigen Charakter der FhG bedingt. Dieser weist das ISST als „quasi öffentliche“ Institution aus und macht seine Dienstleistungen besonders auch für die öffentliche Verwaltung interessant.

Um potentielle Nutzer mit der Funktion und den Vorteilen neuer Integrationstechniken und neuer Technologien der werkzeuggestützten Softwareentwicklung vertraut zu machen, betreibt das ISST Demonstrations- und Trainingszentren, die kontinuierlich ausgebaut werden. Darüber hinaus gibt es eine enge Zusammenarbeit mit den überall für die Softwareindustrie entstehenden Dienstleistungszentren (z.B. „Software-Industrie Support Zentrum SISZ“ in Nordrhein-Westfalen und „Softwareindustrie Berlin/Brandenburg SIBB“).

1.4 Kuratorium

Das Institut wird von einem Kuratorium beraten, dessen Mitglieder aus Wirtschaft, Wissenschaft, Politik und Verwaltung kommen. Zur Zeit gehören ihm an:

Wirtschaft

- Heinz Paul Bonn
Geschäftsführender Gesellschafter der Firma GUS mbH, Köln
Vorsitzender des Kuratoriums
- Arnulf Ganser
Direktor in der Generaldirektion Telekom, Bonn
- Horst Rose
Generalbevollmächtigter IBM Deutschland GmbH

Wissenschaft

- Prof. Dr. Kurt Kutzler
Professor am Fachbereich Informatik der Technischen Universität Berlin

Politik und Verwaltung

- Jochen Stoehr
Abteilungsleiter in der Senatsverwaltung für Wissenschaft und Forschung des Landes Berlin
- Dr. Klaus Warnke-Gronau
Ministerialdirigent im Ministerium für Wirtschaft, Mittelstand und Technologie des Landes Nordrhein-Westfalen, Düsseldorf

1.5 Personal und Sachausstattung

Zum Ende des Jahres waren im ISST 50 wissenschaftliche (33 in Berlin/17 in Dortmund), 16 nicht wissenschaftliche (13/3) und 37 studentische (22/15) Mitarbeiterinnen und Mitarbeiter beschäftigt. In Berlin hat der größte Teil der wissenschaftlichen Mitarbeiter eine Ausbildung als Diplom-Informatiker; einige sind Physiker, Mathematiker oder Diplom-Ingenieure, alle aber mit dem Schwerpunkt Informatik in ihrer bisherigen Berufspraxis. Im Institutsteil Dortmund sind alle wissenschaftlichen Mitarbeiter - von einer Ausnahme abgesehen (Diplom-Kaufmann) - Diplom-Informatiker. Zwei Mitarbeiter sind habilitiert, 15 promoviert. Das Durchschnittsalter beträgt 37,4 Jahre (Berlin) bzw. 32 Jahre (Dortmund). Durch die Verbindungen zu den Universitäten (Technische Universität Berlin und Universität Dortmund) sowie durch interne und externe Fortbildungsveranstaltungen war eine kontinuierliche Verbesserung der fachlichen Qualifikation der Mitarbeiter möglich.

In Berlin nutzt das ISST Berlin gemeinsam mit anderen Forschungseinrichtungen der Fraunhofer-Gesellschaft ein Institutsgebäude und verfügt dort über etwa 1300 qm Bürofläche. Die rechen-technische Infrastruktur in Berlin besteht aus einem Rechnernetz mit einer Client-Server-Architektur. Dazu sind alle Arbeitsräume in ein lokales Netzwerk auf der Basis von Thinwire-Ethernet (Übertragungsrate 10 Megabit/s) integriert, das durch Repeater und Bridges strukturiert ist. Zusätzlich hat ein Teil der Arbeitsräume Anschluß an einen sternförmig von einem Konzentrador installierten FDDI-Ring (Übertragungsrate 100 Megabit/s) auf der Basis einer Glasfaserverkabelung bis zum Arbeitsplatz. Auf diesen durch Bridges gekoppelten Netzwerken werden als Protokolle TCP/IP und Appletalk eingesetzt. Für Wide-Area-Verbindungen steht ein 64 Kilobit/s schneller Anschluß an das Wissenschaftsnetz (WIN) des Vereins zur Förderung eines Deutschen Forschungsnetzes (DFN) mit Verbindung zum weltweiten Internet zur Verfügung. Weiterhin verfügt das Institut über Dienste wie elektronische Post, Dateitransfer, Zugriff auf andere Rechner und Informationsdienste nach Internet- und OSI-Standards. Die benötigte hohe Rechenleistung wird durch ein UNIX-Rechnernetz mit sechs SUN-Sparc10-Servern und einem SUN-4/470-Server erbracht. Als Arbeitsplätze dienen Hochleistungs-X-Terminals, SUN-Farbgraphik-Workstations, Apple-Macintosh-Rechner und PCs, die teilweise mit Audio- und Video-Erweiterungen ausgestattet sind. Durch Integration in das Netzwerk werden an jedem Arbeitsplatz gleiche Arbeitsumgebungen, der Zugriff auf alle Ressourcen und multimediale Informations- und Kommunikationssysteme bereitgestellt. Die Softwareausstattung umfaßt modernste Werkzeuge und Umgebungen zur Softwareentwicklung.

Die rechen-technische Infrastruktur wird von zwei Mitarbeitern, unterstützt von studentischen Hilfskräften, betreut. Dem Personal steht eine eigene, gutsortierte Fachbibliothek zur Verfügung (rund 5200 Einzeltitel und 90 (Fach-)Zeitschriften).

Für den Dortmunder Institutsteil war das einschneidende Ereignis im Jahr 1993 der Auszug aus den Räumen der Universität Dortmund im Oktober. Im Gebäude der Forschungs- und Entwicklungsgesellschaft mbH im Dortmunder Techno-



**Das neue Gebäude des ISST
im Dortmunder Techno-Park**

Park verfügt das Institut nun über ca. 800 qm Bürofläche. Neben den Büroräumen stehen Rechnerpools, Besprechungszimmer und ein großer Schulungsraum zur Verfügung. Die Räume sind sternförmig über UTP-Kabel verbunden. Eine Kopplung der einzelnen Segmente erfolgt durch mit Broutern verbundene HUBs. Als Protokoll wird TCP/IP auf Ethernet eingesetzt. Eine Internet-Anbindung ist über eine semi-permanente ISDN-Verbindung zu einem Xlink-Pop in Castrop-Rauxel realisiert. Für rechenintensive Arbeiten stehen SUN Sparc 10, als Universalworkstation Sparc LX und als einfache Workstation SUN Classic zur Verfügung. Dazu kommen IBM RS 6000/250 Workstations als Entwicklerarbeitsplätze für den IBM 9076, einen Parallelrechner. Eine größere Anzahl PCs ist zum einen im Bereich der Multimediaanwendungen und zum anderen im Bereich der Softwareentwicklung im Einsatz. Als Server werden eine SUN 690 MP, eine IBM RS 6000/570, eine SUN Sparc 10 sowie ein HP Vectra 486/33 ST eingesetzt. Im UNIX-Bereich steht dem Anwender ein Single System Image zur Verfügung. Dies bedeutet, daß dem Anwender unabhängig von der benutzten Workstation immer die gleiche Arbeitsumgebung bereitsteht. Dies wird durch eine Client-Server-Architektur und durch die interne Vernetzung möglich. In das lokale Netz sind alle Workstations, Server, PCs, Drucker etc. integriert. Hiermit wird ein Zugriff auf die unterschiedlichen Ressourcen im Netz ermöglicht. In das lokale Netz sind auch der IBM Parallelrechner sowie eine IBM AS 400 eingebunden.

1.6 Haushalt und Finanzierung

Der Betriebshaushalt des ISST Berlin stieg im Vergleich zum Vorjahr um 30% auf 3,6 Mio. DM an. Davon entfielen 2,8 Mio. DM auf Personal- und 0,8 Mio. DM auf Sachkosten. Etwa 45% der Aufwendungen (1,6 Mio. DM) konnten durch Projekterträge erwirtschaftet werden (Steigerungsrate bei Industrie- und öffentlichen Aufträgen gegenüber 1992: 190%; Stagnation bei öffentlichen Projektfördermitteln). Der Rest wurde durch die FhG-Grundfinanzierung abgedeckt.

Im ISST Dortmund hat sich der Betriebsaufwand - bedingt durch den Personal- aufbau - im Vergleich zum Vorjahr auf rund 2,1 Mio. DM verdreifacht (davon waren 1,5 Mio. DM Personal- und rund 0,6 Mio. DM Sachkosten). Allerdings konnten in 1993 auch die dreifachen Erträge (Gesamtsumme 1,4 Mio. DM) erwirtschaftet werden. Dabei wurde der Industrieertrag um über 1000% auf 0,5 Mio. DM gesteigert; er beträgt damit knapp 25% der Kosten. 0,8 Mio. DM wurden über öffentlich geförderte Projekte erzielt. Der Restbetrag wurde durch die Grundfinanzierung vom Land Nordrhein-Westfalen zur Verfügung gestellt. Tabelle 1 zeigt die Zahlen im Detail.

In Berlin wurden 1993 insgesamt über 0,8 Mio. DM investiert, davon über 80% in die rechtechnische Infrastruktur. Für die Renovierung des Berliner Institutsgebäudes sind aus zentralen Mitteln der Fraunhofer-Gesellschaft nochmals rund 0,8 Mio. DM eingesetzt worden. Das Investitionsvolumen in Dortmund betrug im Geschäftsjahr 1993 2,0 Mio. DM. Der größte Teil wurde dabei für die Anschaffung der IBM Hardware ausgegeben.

ISST	Berlin		Dortmund	
	in TDM	Anteil	in TDM	Anteil
Betriebsaufwand	3.612		2.093	
Personalkosten	2.555	63%	1.375	66%
Betriebsfremde	278	5%	183	9%
Sachaufwand	570	27%	385	18%
Leistungsverrechnung innerhalb der FhG	208	5%	150	7%
Finanzierung				
Projekteinnahmen insgesamt	1.618	45%	1.393	67%
- öffentl. Projektförderung	1.061	29%	815	39%
- sonstige Projekterträge	557	15%	578	28%
Grundfinanzierung	1.995	55%	700	33%

Tab. 1:
Betriebsaufwand
und Finanzierung in
den Institutsteilen

2

Arbeiten der Fachbereiche

2.1 Projektüberblick

Die wissenschaftliche Arbeit in den Fachbereichen und Arbeitsgebieten erfolgte in der Hauptsache im Rahmen von Projekten; d.h. sie war hinsichtlich der angestrebten Ziele und Ergebnisse sowie der eingesetzten Potentiale und des Zeitaufwands überschaubar strukturiert. In der Regel wurden in einem Arbeitsgebiet mehrere Projekte bearbeitet; dies erfolgte zum Teil standortübergreifend, also über Berlin und Dortmund verteilt.

Der Charakter der Projekte war unterschiedlich. Einerseits wurden externe Projekte aufgrund institutioneller Förderung oder im Auftrag privater Unternehmen bzw. öffentlicher Institutionen bearbeitet. In diesen Fällen erhielt der Auftraggeber eine bestimmte Dienstleistung. Andererseits wurden Projekte vom ISST selbst - als Eigenforschungsprojekte - aus Mitteln der Trägerorganisation (Grundfinanzierung der Fraunhofer-Gesellschaft) getragen.

Eigenforschungsprojekte haben den Zweck, wissenschaftlichen Vorlauf für die Akquisition neuer Auftragsprojekte zu schaffen und innovative Prototypen für spätere Aufträge zu entwickeln. Ein spezieller Aspekt der Eigenforschungsprojekte im ISST ist der interne Technologietransfer zwischen den Institutsteilen Dortmund und Berlin, um eine gemeinsame wissenschaftlich-technische Basis herzustellen.

2.1.1 Öffentlich geförderte Projekte

Das ISST hat 1993 an folgenden Verbundprojekten mitgearbeitet:

- BMFT-Verbundprojekt APPLY (*A Practical and Portable Lisp „Ynvironment“*): Entwicklung eines bedarfsgerechten und effizienten Lisp-Systems (siehe *Werkzeuggestützte Softwareentwicklung*, Arbeitsgebiet *Lisp-Entwicklung*)
- Einführung von CASE-Technologie in die industrielle Praxis (Auftraggeber: Ministerium für Wirtschaft, Mittelstand und Technologie des Landes Nordrhein-Westfalen; siehe *Werkzeuggestützte Softwareentwicklung*, Arbeitsgebiet *CASE-Einführung*)
- Eureka-Projekt ESF¹: Internationales Verbundprojekt zur Entwicklung von Konzepten, Werkzeugen und Softwareinfrastrukturen zur industriellen Produktion von Software (siehe *Integrierte Software-Infrastrukturen*, Arbeitsgebiet *Software-Fabrik-Engineering*)
- Eureka-Projekt ECMA-PCTE (*Portable Common Tool Environment*): Implementierung des PCTE-Standards und objektorientierter Erweiterungen (siehe *Integrierte Software-Infrastrukturen*, Arbeitsgebiet *Daten- und Objektmanagement*)

1. EUREKA Software Factory: Größtes internationales Forschungsprojekt zum Software Engineering und zu Software-Entwicklungsumgebungen; Laufzeit 1986-1994

2.1.2 Auftragsprojekte

Durch private oder öffentliche Auftraggeber finanzierte Projekte waren:

- MAS90: Analyse der Entwicklungsperspektive des Mittelstands-Anwendungssystems MAS90 (Auftraggeber: IBM Deutschland GmbH; siehe *Werkzeuggestützte Softwareentwicklung*, Arbeitsgebiet *Re-Engineering*)
- MAS90-TIS: Entwicklung des Tabelleninformationssystems (TIS) zur Unterstützung des Installationsprozesses des Mittelstands-Anwendungssystems MAS90 (Auftraggeber: IBM Deutschland GmbH; siehe Arbeitsgebiet *Re-Engineering*)
- LIT: Konzeption einer Systemverwaltung für ein Service- und Administrationszentrum (SAZ) im Landesamt für Informationstechnik Berlin (Auftraggeber: LIT Berlin; siehe *Integrierte Software-Infrastrukturen*, Arbeitsgebiet *System- und Netzwerkmanagement*)
- WAM: Realisierung einer computergestützten Vorgangsbearbeitung für räumlich getrennte Arbeitsgruppen (Auftraggeber: DeTeBerkom; siehe *Integrierte Software-Infrastrukturen*, Arbeitsgebiet *Computergestützte Kooperation*)
- FISCUS: Neukonzeption der EDV-technischen Abwicklung des Steuerwesens (Auftraggeber: Finanzministerien des Bundes und der Länder; siehe *Prozeß-Engineering*)
- KSG: Vorgangsmodellierung für die Bearbeitung von Kraftwerks-Simulatoränderungen (Auftraggeber: Kraftwerks-Simulator-Gesellschaft (KSG) mbH; siehe *Prozeß-Engineering*)
- DIS: Entwicklung eines Dateninformationssystems (Auftraggeber: Ministerium für Wirtschaft, Mittelstand und Technologie des Landes Nordrhein-Westfalen; siehe *Prozeß-Engineering*)

2.1.3 Interne Projekte

Folgende internen bzw. Eigenforschungsprojekte wurden 1993 am ISST bearbeitet:

- II-Konzepte: Entwicklung, Weiterentwicklung und prototypische Anwendung der objektbasierten Spezifikationssprache II und Erstellung einer dazugehörigen Software-Entwicklungsumgebung (siehe *Werkzeuggestützte Softwareentwicklung*, Arbeitsgebiet *Objektbasierte Softwareentwicklung*)
- Interne Softwareentwicklungstechnik: Aufbauend auf den II-Konzepten und -Werkzeugen wurde damit begonnen, eine ISST-eigene Softwareentwicklungsmethodik zu erarbeiten (siehe *Werkzeuggestützte Softwareentwicklung*, Arbeitsgebiet *Objektbasierte Softwareentwicklung*)

- Re-Engineering-Kit: Entwicklung eines „Methoden- und Werkzeugkastens“ zum Re-Engineering von Software-Systemen. Dieser soll die methodische und technische Basis für Re-Engineering-Projekte liefern (siehe *Werkzeuggestützte Softwareentwicklung, Arbeitsgebiet Re-Engineering*)
- CASE-Development-Kit: Entwicklung eines „Methoden- und Werkzeugkastens“ zur Vorbereitung und Realisierung einer werkzeuggestützten Softwareentwicklung in Unternehmen (siehe *Werkzeuggestützte Softwareentwicklung, Arbeitsgebiet CASE-Einführung*)
- MEDIAWORKS: Realisierung eines Multimedia-Datenbanksystems mit einfacher Bedienungsoberfläche und integriertem Desktop-Publishing-System (siehe *Integrierte Software-Infrastrukturen, Arbeitsgebiet Daten- und Objektmanagement*)
- TELIS: Entwicklung von Software-Infrastrukturen für integrierte Point-of-Information- und Point-of-Sale-Systeme (siehe *Integrierte Software-Infrastrukturen, Arbeitsgebiet Multimedia-Systeme*)
- SINUS: Konzepte und Prototypen für die Systemadministration in mittleren und großen Organisationen (siehe *Integrierte Software-Infrastrukturen, Arbeitsgebiet System- und Netzwerkmanagement*)
- HOTCON: Hotline- und Consulting-System zur Unterstützung von Benutzern in Problem- und Fehlersituationen (siehe *Integrierte Software-Infrastrukturen, Arbeitsgebiet System- und Netzwerkmanagement*)
- ICOMA: Konzepte und Prototypen zur Konfiguration von verteilten Systemen (siehe *Integrierte Software-Infrastrukturen; Arbeitsgebiet System- und Netzwerkmanagement*)
- COBRA-1: Aufbau und Erprobung moderner Kommunikationsinfrastrukturen zur Unterstützung der kooperativen Arbeit zwischen Fraunhofer-Instituten und deren Außenstellen (siehe *Integrierte Software-Infrastrukturen, Arbeitsgebiet System- und Netzwerkmanagement*)
- Prozeßmanagement-Kit: Entwicklung eines „Methoden- und Werkzeugkastens“ zum systematischen Management von Geschäftsprozessen (siehe *Prozeß-Engineering*)
- European Professional Software Engineering Program (EPSE): Ausbildungsprogramm zum Software-Ingenieur bzw. industriellen Software-Entwickler (siehe *Aus- und Weiterbildung*)

2.2 Werkzeuggestützte Softwareentwicklung

Zielsetzung des Bereichs

Der derzeitige „Stau“ bei Anwendungssoftware erfordert, daß in der Softwareentwicklung in Zukunft verbesserte, vor allem werkzeuggestützte Fertigungstechnologien zum Einsatz kommen. Das ISST konzentriert sich daher in diesem Bereich auf die Entwicklung leistungsfähiger Software-Entwicklungsmethoden und -werkzeuge.

Ein strategisches Ziel des ISST besteht darin, für ein breites Spektrum von software- und systemtechnischen Anwendungen integrierende Methoden und weitgehend hardwareunabhängige Werkzeuge und Plattformen bereitzustellen, die eine *ingenieurmäßige Entwicklung wiederverwendbarer Softwarebausteine* ermöglichen. Die verwendeten Methoden und Werkzeuge sollen dabei gewährleisten, daß Software

- mit höherer Produktivität hergestellt werden kann,
- nachweisbar bestimmten Qualitätsanforderungen genügt,
- noch im Entwicklungsprozeß im Hinblick auf ihre Funktionalität erprobt (Prototyping) und
- möglichst unproblematisch auch auf parallele und verteilte Hardware portiert werden kann.

Die Entwicklung von Software mit Bausteincharakter hat eine *geplante* Wiederverwendung zum Ziel. Daneben gewinnt die *ungeplante* Wiederverwendung zunehmend an Bedeutung. Dabei geht es um die Anpassung existierender Systeme an neue Hard- und Software-Architekturen. Dies erfordert z.B. Methoden und Werkzeuge für den Umkehrungsprozeß (Reverse Engineering) zur eigentlichen Softwareentwicklung (Forward Engineering).

Das ISST entwickelt auf der Basis eines Kerns fundamentaler Konzepte und Sprachen neue Methoden des Software Engineering und stellt dafür adäquate Werkzeuge bereit.

Leistungsfähige Software-Entwicklungsmethoden und -werkzeuge allein reichen aber nicht aus. Dies illustriert die heutige Situation bei der Anwendung von CASE-Systemen (*Computer Aided Software Engineering*) sehr deutlich. Eine ganze Reihe leistungsfähiger CASE-Systeme ist bereits auf dem Markt verfügbar. Der bisher durch CASE und CASE-Werkzeuge erreichte Nutzen ist jedoch bisher eher gering. Dies hat verschiedene Ursachen. Einerseits weisen die am Markt erhältlichen Werkzeuge eine Reihe von Mängeln auf, andererseits fehlen wichtige Voraussetzungen, um sie in der Praxis wirksam anzuwenden. Damit die Einführung von Software-Entwicklungswerkzeugen und -umgebungen in die industrielle Praxis gelingt, müssen die erforderlichen Migrationsprozesse analysiert, praktikable Konzepte für die Einführung definiert und Unternehmen bei ihrem Strukturwandel unterstützt werden.

Arbeitsgebiete im Bereich Werkzeuggestützte Softwareentwicklung

Das ISST hat seine Ziele im Bereich „Werkzeuggestützte Softwareentwicklung“ im Berichtsjahr auf folgenden Arbeitsgebieten verfolgt:

- **Objektbasierte Softwareentwicklung**

Dieses Arbeitsgebiet zielt auf die Ausarbeitung und Weiterentwicklung von Basistechniken und Werkzeugen für die Entwicklung modularer Software-Systeme. Zentraler Gegenstand der Arbeiten war die objektbasierte Spezifikationssprache II und die Bereitstellung entsprechender Softwarewerkzeuge.
- **Lisp-Entwicklung**

Innerhalb des BMFT-Verbundprojektes APPLY war das ISST an der Schaffung von bedarfsgerechten und effizienten Lisp-Systemen beteiligt. Hier sollte es möglich werden, Lisp-Anwendungen problemlos in die übrige Softwarewelt zu integrieren (z.B. gegenseitige Aufrufbarkeit von Lisp- und C-Programmen) und dabei eine Speicher- und Laufzeiteffizienz zu erreichen, die der von C-Programmen vergleichbar ist.
- **Re-Engineering**

In diesem Arbeitsgebiet beschäftigt sich das ISST mit Methoden und Techniken zur gründlichen Überarbeitung von (Software-)Systemen. Re-Engineering ist immer dann notwendig, wenn existierende Systeme an veränderte Nutzeranforderungen (z.B. neue Geschäftsprozesse), erhöhte Qualitätsansprüche oder an moderne Hard- und Software-Architekturen (z.B. Client-Server-Architekturen) angepaßt werden müssen. Die Arbeiten am Institut gehen dabei von Re-Engineering-Szenarien aus, wie sie in konkreten Industrieprojekten auftreten.
- **CASE-Einführung**

Ziel der Arbeiten im Arbeitsgebiet CASE-Einführung ist die Entwicklung eines Bündels von Konzepten, Methoden und Werkzeugen, um Unternehmen bei der Einführung der werkzeuggestützten Softwareentwicklung zu unterstützen. Bei der Beratung werden neben den technischen Bedingungen besonders auch die organisatorischen und personellen Voraussetzungen der Unternehmen berücksichtigt.
- **CASE-Kompetenzzentrum**

Im CASE-Kompetenzzentrum des ISST werden Software-Entwicklungswerkzeuge und -Produktionsumgebungen verschiedener Hersteller bezüglich eines methodenspezifischen Anforderungskataloges bewertet. Dabei werden nicht nur funktionale, sondern auch nicht-funktionale Kriterien wie Benutzerfreundlichkeit und Integrationsfähigkeit berücksichtigt. Die Ergebnisse werden in Berichten zusammengefaßt.

- Software-Qualitätssicherung

Im Arbeitsgebiet Software-Qualitätssicherung werden Konzepte und Methoden zur Einführung einer systematischen Qualitätssicherung über den gesamten Software-Life-Cycle entwickelt. Im Vordergrund stehen dabei Arbeiten zur ISO 9000-Konformität der eingesetzten Qualitätssicherungssysteme.

2.2.1 Objektbasierte Softwareentwicklung

Aufgabenstellung und Ziele

Im Rahmen dieses Arbeitsgebietes wurde die objektbasierte Spezifikations- und Implementationssprache II entwickelt, um eine systematische Produktion großer, nebenläufiger und verteilter Software-Systeme zu ermöglichen. Dabei soll durch die Sprache und die entsprechenden Entwicklungswerkzeuge ein rigoroser softwaretechnischer Stil erzwungen werden, der sich durch eine strenge Modularisierungsdisziplin auszeichnet und eine gute Kontrolle von Seiteneffekten ermöglicht.

Der hohe Formalisierungsgrad der Schnittstellen von II erlaubt es, Module wiederzuverwenden und dabei ein großes Maß an Sicherheit in der Funktionalität der erstellten Programme zu erreichen. Dabei geht es sowohl um geplante als auch um ungeplante Wiederverwendung von Modulen und Systemen.

Zum Entwurf und zur Wiederverwendung von Software soll II prinzipiell unter Nutzung von Softwarewerkzeugen angewendet werden. Neben der Unterstützung der interaktiven Erstellung von Quelltexten sollen diese auch die Überprüfung von Konsistenz- und Kontextbedingungen sowie die Ausführung unvollständiger Spezifikationen gestatten und damit helfen, die Korrektheit der Spezifikationen und der aus ihnen abgeleiteten Programme zu sichern.

Die Sprache II und ihre Werkzeuge sollen eine wesentliche Rolle bei der Herausbildung einer weitgehend homogenen Methoden- und Werkzeug-Basis am ISST spielen. Sie ist als ein projektübergreifender Integrationsfaktor anzusehen und bildet einen Teil des Kerns fundamentaler Konzepte und Sprachen zur Entwicklung tragfähiger Methoden des Software Engineering am Institut.

Vorgehensweise

Das prinzipielle Vorgehen im Arbeitsgebiet „Objektbasierte Softwareentwicklung“ ist durch die Grundkonzepte der Sprache II und ihrer Methoden geprägt.

Es beruht sowohl bei der Strukturierung des Softwareentwicklungsprozesses unter Anwendung von II als auch beim Entwurf von Software-Architekturen auf den beiden fundamentalen Strukturierungsprinzipien „Teile und herrsche“ (*divide and conquer*) und „Separierung von Charakteristiken“ (*separation of concerns*).

Das erste Prinzip, das dem Abstraktionsprinzip „Verbergen von Information“ (*information hiding*, auch: Geheimnisprinzip) entspricht, wird in der Sprache II durch eine hierarchische Struktur der Softwarearchitektur realisiert: Jedes zu spezifizierende Software-System wird als Konfiguration von Komponenten dargestellt. Dabei können die Komponenten nebenläufig ausführbare Module (*concurrently executable modules*, abgekürzt CEM) oder selbst wieder Konfigurationen von Komponenten sein. Die CEMs sind nach einem einheitlichen Prinzip strukturiert. Nach Auflösung der beschriebenen rekursiven Beziehung bilden sie die Basiskomponenten eines jeden mit II spezifizierten Software-Systems.

Dem zweiten Prinzip der Softwareentwicklung entspricht das Abstraktionsprinzip „Weglassen von Information“ (*information neglection*). Es wird in der Sprache II durch verschiedene Sichten (*views*) realisiert, aus denen jedes Modul spezifiziert werden kann. Grundlegend sind dabei die Typ-Sicht (*type view*), die imperative Sicht (*imperative view*) und die Nebenläufigkeits-Sicht (*concurrency view*), in denen die Schnittstellen eines Moduls sowie seine interne Struktur jeweils durch algebraische Spezifikation, durch die Angabe imperativer Programme für die Operationen des Moduls sowie durch die Einschränkung ihrer möglichen Nebenläufigkeit durch Pfadausdrücke definiert werden. Die Offenheit des Sichtenkonzepts ermöglicht es, entsprechend dem Bedarf von Anwendungen und dem Stand der Forschung weitere Sichten, z.B. über zeitliche Einschränkungen des Programmverhaltens (*performance view*) oder über die Zuordnung von Softwarekomponenten zu Prozessoren (*distribution view*), zu integrieren. So können existierende Spezifikationen und Werkzeugkomponenten für die bestehenden Sichten gültig bleiben.

Zur Unterstützung des Software-Entwicklungsprozesses mit II wurde in Kooperation mit der Universität Dortmund eine Entwicklungsumgebung entworfen. Alle Komponenten der Umgebung arbeiten auf einem gemeinsamen internen Objektmodell. Durch die Realisierung eines abstrakten Fenstersystems für die Nutzeroberfläche ist die Umgebung unabhängig von einem konkreten Fenstersystem. Die Schnittstelle zur persistenten Datenhaltung wurde durch eine direkte Anbindung an das UNIX-Filesystem möglich, ist aber dank ihrer abstrakten Definition ebenfalls austauschbar.

Da jedes II-Modul gleichzeitig aus verschiedenen Sichten spezifiziert sein kann, erfordert die Korrektheitsanalyse ein spezifisches Herangehen. Die Werkzeuge zur Überprüfung der Konsistenz und der kontextsensitiven Korrektheit von Spezifikationen ermöglichen daher unter anderem auch, redundante Teile von Spezifikationen zu generieren. Die (beabsichtigte) Redundanz entsteht dabei durch die Überlappung der verschiedenen Sichten.

Die im Rahmen dieser Arbeiten erzielten Ergebnisse sollen in Projekten des ISST praktisch angewandt werden. Dazu wurde zunächst eine Analyse der gegenwärtigen Softwareentwicklungstechnik am Institut vorgenommen. Diese bildet die Basis für die Ausarbeitung und Erprobung einer eigenen ISST-Softwareentwicklungsmethode. Diese soll nicht nur „hausintern“, sondern auch zur Analyse unternehmensspezifischer Modelle der Softwareentwicklung und zur Beratung bei der Einführung neuer Softwareentwicklungsmethoden in Unternehmen und Verwaltungen angewendet werden.

Ein besonderer Schwerpunkt der II-Methode besteht in der Unterstützung der frühen Phasen des Software-Lebenszyklus, insbesondere des Entwurfs. Ihre

Erprobung erfolgte bisher in einem mehrwöchigen Testprojekt am ISST, das die wesentlichen Phasen einer Softwareentwicklung (Begriffsanalyse, Daten- und Geschäftsprozeßmodellierung, Architekturentwurf einschließlich Datenbankanbindung, Spezifikation mit II und Implementierung) beinhaltet. Die Erfahrungen wurden dokumentiert und hinsichtlich einer weiteren Vervollkommnung der Methodik ausgewertet.

Ergebnisse

- Syntax II 93

Nach ersten Erfahrungen bei der internen Anwendung von II am ISST und an der Universität Dortmund wurde nach geringfügigen Modifikationen der Syntaxversion II 92 die auf absehbare Zeit stabile Syntax II 93 festgelegt.

- Prototyp einer Software-Entwicklungsumgebung

Ein Prototyp einer Software-Entwicklungsumgebung für die objektbasierte Spezifikationssprache II wurde fertiggestellt und anhand einiger Anwendungsfälle erprobt. Die Umgebung unterstützt die inkrementelle Softwareentwicklung und wurde selbst entsprechend der II-Methodik objektbasiert entworfen. Wesentliche Komponenten sind ein syntaxgesteuerter Moduleditor, ein graphischer Konfigurationseditor, die Bibliotheksverwaltung, ein Kontextchecker zur semantischen Analyse und ein Compiler zur Generierung und Ausführung von C-Quelltexten aus II-Spezifikationen.

- Methodik für die Softwareentwicklung mit II

Verschiedene Ausprägungen der II-Softwareentwicklungsmethodik wurden in internen Projekten am ISST und an der TU Berlin erprobt. Ein wesentliches Merkmal dieser Methodik ist eine enge Kopplung von Analyse, Entwurf, Spezifikation und Implementierung mit dem Ziel einer durchgängig modularen Softwareentwicklung. Die weiteren Arbeiten im Kontext von II werden sich im Jahre 1994 vor allem auf dieses Arbeitsgebiet („ISST-Methode“) konzentrieren.

Veröffentlichungen

[2], [32], [50], [54], [55], [60], [61], [62], [76], [80], [86]

2.2.2 Lisp-Entwicklung

Aufgabenstellung und Ziele

Das Ziel des BMFT-Verbundprojektes APPLY bestand darin, eine Strategie zur Lisp-Implementation zu entwickeln, die Lisp-Anwenderprogrammen die Speicherplatz- und Laufzeitperformanz von äquivalenten C-Programmen verleiht. Dabei sollte ein Konzept entwickelt und überprüft werden, das eine natürliche Integration von Lisp-Programmen in die übrige Softwarewelt ermöglicht, ohne daß die Vorteile von Lisp für die Programmentwicklung verloren gehen.

Lisp soll dadurch die notwendige Akzeptanz bei Softwareproduzenten und Anwendern finden und in der Nutzung nicht nur auf die KI-Forschung beschränkt bleiben. Insbesondere soll die Lücke geschlossen werden, die sich zwischen dem Stand der KI-Forschung und der verfügbaren KI-Basis-Software aufgetan hat, da sie die rasche kommerzielle Nutzung der Forschungsergebnisse zunehmend behindert oder gar unmöglich macht.

Im Rahmen von APPLY erfolgte eine Zusammenführung bzw. Abstimmung deutscher Lisp-Initiativen an Forschungsinstituten und Hochschulen. Durch die Konzentration der Erfahrungen soll der Einfluß auf die internationale Standardisierung von Lisp verstärkt werden. Die Ergebnisse des Sprachdesigns und die Implementationserfahrungen sollen insbesondere in die europäischen Standardisierungsbemühungen (EuLisp) eingebracht werden.

Vorgehensweise

Das APPLY-Vorhaben ging von folgenden Arbeitshypothesen aus:

1. Für in Lisp geschriebene Anwendungen läßt sich die gleiche Performanz erzielen wie für äquivalente C-Programme, wenn bereits mit Beginn der Konzeption des Lisp-Systems folgende Punkte berücksichtigt werden:
 - Klare Trennung zwischen Programmentwicklung und Anwendung
 - Verbesserung der Möglichkeiten zur Gewinnung von statischen Programminformationen durch den Compiler für komplexere Optimierungen
 - Verbessertes Sprachdesign
 - Konfigurierbarkeit
2. Die Erzeugung von Objekt- und C-Quellfiles aus Lisp-Programmen ist eine wesentliche Grundlage für die harmonische Integration in die übrige Softwarewelt.

Die Implementationsstrategie existierender Lisp-Systeme² bietet keine grundsätzliche Lösung, um die oben genannten Ziele zu erreichen. Dies zeigt sich nicht zuletzt in der geringen Akzeptanz von Lisp außerhalb der KI. Lisp-Systeme benötigen für die Abarbeitung fertiger Applikationen dieselben leistungsfähigen Rechner wie zur Entwicklung und sind verhältnismäßig speicherplatzintensiv. Das gilt auch für Lisp-Systeme, die die Ablaufumgebungen für Applikationen verkleinern. Eine Lisp-Applikation kann bisher nicht unabhängig vom

Lisp-System abgearbeitet werden. Dies führt nicht nur dazu, daß Lisp-Applikationen Inselfösungen sind, sondern erhöht für Anwender auch die Kosten, da zusätzlich Lizenzgebühren für das Lisp-System anfallen.

Vor diesem Hintergrund hat sich die APPLY-Projektgruppe des ISST für eine radikal neue Strategie bei der Implementation von Lisp entschieden.

Ergebnisse

Schwerpunkte der Aktivitäten im Projekt APPLY waren

- die Zusammenführung der einzelnen Compilerteile,
- der Ausbau des Compilers, so daß dieser EuLisp-Level-0 Programme verarbeiten kann und
- der Anschluß eines Codegenerators zum Erzeugen von gut lesbarem C-Code.

Als Ergebnis wurde die erste Version des „EuLisp nach C - Compilers“ (Eu2C) auf dem ftp-Server, unter anonymous ftp von 'ftp.isst.fhg.de' im Verzeichnis '/pub/APPLY/Distribution', bereitgestellt.

Im einzelnen liegen Ergebnisse zu folgenden Arbeitspaketen vor:

- Typen und Typinferenz: Der neu erarbeitete Ansatz zur Typinferenz wurde in den Compiler integriert, die notwendigen Typschemata wurden definiert.
- Programmanalyse und Optimierung: Die Arbeiten an den wichtigsten Optimierungen, das „Inlining“ von Funktionen und die Auflösung der Rest-Rekursion wurden abgeschlossen.
- EuLisp-Oberteil: Der Transformator von EuLisp auf die Lisp-nahe Zwischensprache (LZS) wurde bis auf die Behandlung von Syntaxfehlern fertiggestellt. Dabei wurde ein Konzept für einen einfachen Übergang zum Metaobjektprotokoll erarbeitet und implementiert sowie ein Makrointerpreter realisiert.
- Speicherverwaltung und Objektrepräsentation: Die Arbeiten an der neuen, effizienteren und sicheren Speicherverwaltung wurden beendet und eine lauffähige Version auf dem ftp-Server des ISST zur Verfügung gestellt.
- Datentypen und Funktionsbibliotheken: Es wurde eine Beschreibungsform für die Standarddatentypen (Klassen) entwickelt und damit die Datentypen implementiert.

2. Die klassische Implementationsstrategie für Lisp ist durch folgende Merkmale gekennzeichnet:

- Keine Trennung zwischen Entwicklung(ssystem) und Anwendung(ssystem). Es ist immer das vollständige System aktiv, unabhängig davon, ob der gesamte Leistungsumfang benötigt wird oder nicht
- Die Integrierbarkeit in andere Softwareprodukte ist nicht gegeben; Lisp ist ein abgeschlossenes System
- Lisp-Systeme sind in zentralen Teilen nicht konfigurierbar
- Compiler können nicht genügend Informationen aus den Programmen gewinnen, um eine mit C vergleichbare Performanz zu erzielen

- Internationale EuLisp-Standardisierung: In Vorbereitung der Definition von EuLisp Version 1.0 wurde die vorliegende Version der Sprachdefinition unter dem Gesichtspunkt der Komplettkompilation durchgearbeitet und entsprechende Änderungsvorschläge eingebracht.
- Integration in GUI (*Graphic User Interface*): Die Arbeiten an den graphischen Nutzeroberflächen dienen dazu, die Integrierbarkeit der mit Eu2C erzeugten Compilats nachzuweisen und anschaulich darzustellen.

Veröffentlichungen

[7], [20], [21], [22], [67]

2.2.3 Re-Engineering

Aufgabenstellung und Ziele

Gegenwärtig sind in unserer Wirtschaft zahlreiche große Software-Systeme im Einsatz, die bereits vor mehr als zehn Jahren entwickelt wurden. Diese Langlebigkeit von Software-Systemen ist von den Anwendern erwünscht, geht es doch darum, teure und zum Teil maßgeschneiderte EDV-Ausstattungen weiterbetreiben zu können. Kontinuierliche Wartung, Anpassung und Weiterentwicklung sollen dabei die Funktionstüchtigkeit und Robustheit der Systeme gewährleisten. Auf der anderen Seite bedingt aber der lange Betrieb der Systeme, daß deren technische Basis (Hardware, Betriebssystem, Datenhaltungssystem etc.) sowie ihre ursprüngliche Konzeption (DV-Entwurf) und organisatorische Einbettung (Fachkonzept) modernen Anforderungen nicht mehr standhalten. Insbesondere können „betagte“ Systeme die Möglichkeiten fortschrittlicher Technologien, wie etwa graphische Oberflächen, relationale oder objektorientierte Datenhaltung, Verteilung und Offenheit, nicht ausreichend ausnutzen bzw. anbieten.

„Software-Re-Engineering“ - die gründliche Überarbeitung von „gealterten“ Software-Systemen - ermöglicht hier einen Ausweg. Es steht zu erwarten, daß sich dieser Bereich innerhalb der Software-Technologie in naher Zukunft zu einem bedeutenden Marktsegment mit hohen Wachstumsraten entwickeln wird. Denn die sinnvolle Sanierung von Software kann Unternehmen helfen, ihre Investitionen der letzten dreißig Jahre trotz der unvermeidbaren Alterung der Software zu schützen und ihre Erträge für die Zukunft zu sichern.

Das ISST will für anwendungsbezogene Problemstellungen individuelle Lösungsmöglichkeiten erarbeiten. Dazu wurden existierende Re-Engineering-Werkzeuge in Hinblick auf ihre Einsetzbarkeit evaluiert sowie eigene prototypische Werkzeuge entwickelt. Ziel ist ein Bündel von Methoden, Techniken und Werkzeugen (*Re-Engineering-Kit*), das Unternehmen Unterstützung bietet für:

- den Entwurf unternehmens- und anwendungsspezifischer Strategien zur Software-Sanierung (*System-Re-Engineering*)

- die Analyse von Schwachstellen in „gealterten“ System- und Organisationsstrukturen (*Reverse Engineering, System and Application Understanding*)
- die Anpassung von Systemen an moderne Hard- und Software-Architekturen (*Migration* oder *Software-Resozialisierung*)
- die methodische und technische Unterstützung bei der Durchführung von Re-Engineering-Aufgaben

Vorgehensweise

In der akademischen Forschung erhielt das Software-Re-Engineering bisher nur geringe Aufmerksamkeit. Das liegt nicht zuletzt an der Breite des Gebietes und der Verschiedenheit der einzelnen Problemstellungen. Von der Industrie werden in der Regel stark auf das konkrete Unternehmen bzw. die konkrete Anwendung zugeschnittene, pragmatische Lösungen gefordert. Das ISST trägt diesen Anforderungen der Praxis bei seinem Vorgehen zum Re-Engineering Rechnung: Bei Industriaufträgen wird der jeweils zugrundeliegende methodische Kern herausgearbeitet, die verwendeten Werkzeuge bewertet und Vorschläge für Werkzeuge erarbeitet, die Lücken in dem jeweiligen Anwendungsfall schließen könnten. Auf diese Weise werden die wesentlichen Szenarien für das Re-Engineering ermittelt und die dafür optimal geeigneten Methoden, Techniken und Werkzeuge zusammengestellt.

1993 wurden vom ISST zwei Industrieprojekte durchgeführt und zwei weitere vorbereitet:

Projekt MAS90 hatte zum Ziel, den Re-Engineering-Bedarf für das gleichnamige System zu ermitteln und Vorschläge für kurz-, mittel- und langfristige Maßnahmen zur Verbesserung des Systems und seiner Akzeptanz zu unterbreiten. Als Grundlage dafür diente die methodische Befragung des Entwicklungsteams, der Wartungsfachleute, der Installationsexperten sowie der Systemnutzer.

Die Nachdokumentation des Systems für die Installationssituation war Gegenstand des Projektes MAS90-TIS. In Zusammenarbeit mit den Entwicklern und Installationsexperten wurden über 800 Parametertabellen und Hunderte von Dateien des Systems, die eine kundenspezifische Anpassung ermöglichen, in einem Glossar von ca. 100 betriebswirtschaftlichen Stichworten und 500 Suchbegriffen gruppiert. Mit dem erstellten Tabelleninformationssystem ist es möglich, die zu einem Stichwort relevanten Tabellen und Dateien von MAS90 zu ermitteln und zu verwalten. Damit läßt sich die Abbildung der Geschäftsvorfälle des Kunden auf die Standardabläufe im System unterstützen.

In einem weiteren Projekt mit der IBM Deutschland GmbH analysiert das ISST FORTRAN-Programmsysteme und deren organisatorisches Umfeld. Ziel ist, die für konkrete Re-Engineering-Aufgaben wesentlichen Informationen in einem Repository aufzunehmen und zu integrieren. Daraus sollen Projektionen abgeleitet werden, die - adäquat visualisiert - den Re-Ingenieur bei seiner Aufgabe unterstützen.

Außerdem erarbeitet das ISST eine Vorgehensweise zur Aufteilung monolithischer Mainframe-Systeme auf moderne Client-Server-Architekturen. Im Vordergrund steht dabei besonders die Verbesserung der Benutzerschnittstelle.

Die Ergebnisse aus diesen Industrie-Projekten werden durch begleitende Eigenforschung im Arbeitsgebiet Schritt für Schritt zu einem *Re-Engineering-Kit* aufgearbeitet.

Ergebnisse

- Die komplexen Re-Engineering-Aufgaben, die zur Zeit in Einrichtungen der Wirtschaft zu bewältigen sind, können weitgehend nicht automatisiert werden, sondern sind von Experten, d.h. Software-Re-Ingenieuren, zu bearbeiten.
- Die vorhandenen Re-Engineering-Werkzeuge, z.B. zur Programmanalyse und -restrukturierung oder zum Reverse Engineering, sind in diesen Szenarien hilfreich, reichen aber zur Bewältigung der Gesamtaufgaben nicht aus.
- Die wesentlichen Defizite im Re-Engineering liegen im Bereich der Methodik.
- Weitaus wichtiger als die Inspektion des Programm-Codes ist für das Re-Engineering auf Systemebene die Informationsgewinnung durch zielgerichtete Interviews mit Entwicklern, Wartungsexperten und Anwendern.

Die Ergebnisse der abgeschlossenen Projekte sind in Arbeits- und Ergebnisberichten zusammengefaßt.

Durch die 1993 initiierten strategischen Kooperationen des ISST mit der Industrie hat das Arbeitsgebiet Re-Engineering im Institut an Bedeutung gewonnen. Mit Wirkung zum 1.1.1994 wird es als eigenständiger Bereich weiter ausgebaut.

2.2.4 CASE-Einführung

Aufgabenstellung und Ziele

Die Erfahrung vieler Unternehmen zeigt, daß allein die Verwendung von Werkzeugen die Probleme der Softwareentwicklung nicht löst. Erst wenn sie in Verbindung mit veränderten Software-Entwicklungsverfahren und -methoden zum Einsatz kommen, sind wirkliche Verbesserungen zu erreichen.

Dabei ist zu berücksichtigen, daß es nicht ein für alle Unternehmen einheitliches Entwicklungsmodell geben kann. Jedes Unternehmen hat seine Besonderheiten, die die Verwendung bestimmter Verfahren und Methoden sinnvoller erscheinen lassen als andere. Demzufolge muß jedes Unternehmen seine spezifische Softwareentwicklungstechnik definieren.

Weiterhin gilt es zu beachten, daß Menschen diese Prozesse maßgeblich bestimmen. Sei es als Entscheidungsträger, die die Veränderungen beschließen oder als Entwickler, die sie in die Praxis umsetzen. Beide Seiten tragen letztendlich mit ihren Aktionen zum Erfolg oder Mißerfolg der unternehmensspezifischen Softwareentwicklung bei. Entscheidend ist daher, daß alle betroffenen Personen über den Entscheidungsprozeß informiert oder sogar daran beteiligt werden. Dabei gilt es nicht nur die Vorlieben und Abneigungen der Personen zu

erfassen, sondern auch ihr Zusammenwirken als Individuen, als Gruppe und als Abteilung zu beschreiben und bestmöglich zu unterstützen.

Vorgehensweise

Das ISST geht bei der Einführung von CASE in die industrielle Praxis von dem Ist-Zustand im jeweiligen Unternehmen aus. Mittels einer gezielten Analyse wird die bisher praktizierte Softwareentwicklung definiert. Das ISST verfolgt dabei nicht den revolutionären Ansatz, bei dem komplett neue Verfahren und Methoden oktroyiert werden, sondern versucht sukzessive, graduelle Verbesserungen zu erreichen. Denn jedes Unternehmen hat auch schon vor dem Einsatz von Werkzeugen Software mehr oder weniger erfolgreich produziert.

Zur Erhebung des Status quo der Softwareentwicklung werden die Informationen im Unternehmen über Interviews erfaßt und liegen nur informal vor. Bei einer informalen Darstellung ist es schwer, die verschiedenen Abhängigkeiten und Interaktionen zu analysieren. Daher ist es hilfreich, wenn zumindest eine semi-formale Darstellung der Informationen in Form eines Modells erfolgt. Durch die Definition von Regeln kann dieses dann analysiert werden, um die konkreten Stärken und Schwächen herauszufinden.

Wichtige Anforderungen an eine semi-formale Darstellung sind Verständlichkeit, Erweiterbarkeit und die Möglichkeit, verschiedene Aspekte zu erfassen. Das Modell muß aber auch umfassend sein, damit nicht nur die technischen, sondern auch die organisatorischen und personellen Aspekte definiert werden können. Zur Vereinfachung der Terminologie bezeichnen wir dieses Modell als DV-Organisationsmodell.

Es gibt verschiedene Modelle, die unter dem Namen *Referenzmodelle* für Software-Entwicklungsumgebungen bekannt sind. Ziel der meisten Referenzmodelle ist die Darstellung des computerisierten Teils, also der Werkzeuge und Komponenten. Dies reicht für das DV-Organisationsmodell nicht aus. In dem Projekt *Eureka Software Factory* (ESF) wurde ein Referenzmodell (CoRe) entwickelt, das sowohl die Werkzeuge als auch die beteiligten Personen berücksichtigt. Mit CoRe ist es möglich, sowohl die Komponenten einer sogenannten Software-Fabrik als auch deren Interaktionen zu modellieren. Auf diese Weise kann eine DV-Organisation beschrieben werden. Da für ein DV-Organisationsmodell, das als Basis für die CASE-Einführung in einem Unternehmen dienen soll, eine detaillierte Darstellung unabdingbar ist, werden die CoRe-Konzepte erweitert.

Alle relevanten Informationen können so in dem Modell definiert werden. Für eine gezielte CASE-Einführung wird der derzeitige Stand einer DV-Organisation erfaßt, analysiert und auf der Grundlage dieser Ergebnisse reorganisiert. Die Vorschläge zur Reorganisation werden dann im Unternehmen umgesetzt. Aus den Zielen der CASE-Einführung ergibt sich ein Ablauf, wie ihn Abb. 3 zeigt.

Die Aktivitäten sind durch die länglichen Kästen dargestellt. Die Pfeile kennzeichnen den Dokumentenfluß. Die Aktivitäten im einzelnen:

Bei der Akquisition werden Informationen über das Unternehmen auf der Basis von Interviews gesammelt und in Form einer DV-Organisation dargestellt. Die

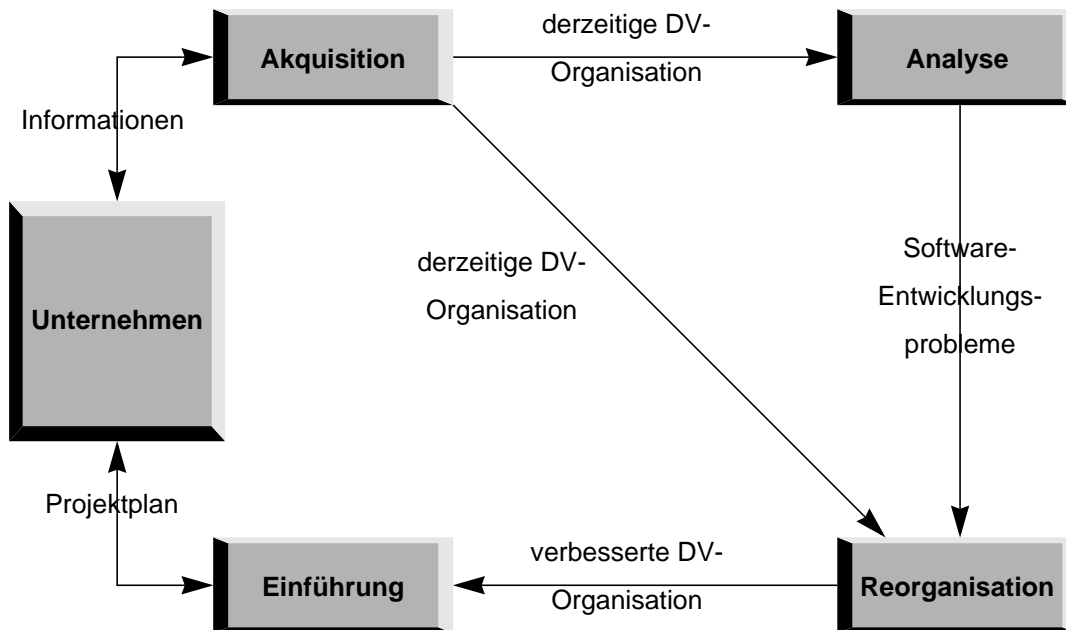


Abb. 3: Prozeß der CASE-Einführung

derzeitige DV-Organisation wird dann analysiert und in Hinblick auf bestimmte Kriterien wie Methoden- und Werkzeugabdeckung bewertet. Bei der Reorganisation wird anschließend versucht, die Probleme, die bei der Analyse entdeckt wurden, zu beheben. Dabei ist es wichtig, eventuelle Randbedingungen, die sich aus der derzeitigen DV-Organisation ergeben, bei einem Lösungsvorschlag mitzubedenken. In der letzten Aktivität „Einführung“ wird in einem Projektplan beschrieben, wie die verbesserte DV-Organisation in das Unternehmen zu integrieren ist.

CASE-Einführung in die industrielle Praxis

Neben dem Ausbau seines Dienstleistungsangebotes zur CASE-Einführung arbeitete das ISST 1993 an einem vom Land Nordrhein-Westfalen (Ministerium für Wirtschaft, Mittelstand und Technologie) geförderten Projekt, das den Technologietransfer in die industrielle Praxis zum Ziel hat. Bei der Konzeption dieses Projektes ging das ISST von der Erkenntnis aus, daß CASE-Werkzeuge nicht nur aus technischen Gründen in den Regalen der Unternehmen verstauben: Die am Markt verfügbaren CASE-Tools lassen sich zum einen nur schlecht integrieren; ein Manko, das Unternehmen bei der Entwicklung von singulären Werkzeugen zu integrierten Umgebungen und Software-Fabriken zu kompensieren versuchen. Andererseits - und dies ist der bedeutendere Aspekt - spielen auch organisatorische und personelle Faktoren bei der Einführung von Werkzeugen in die Softwareentwicklung eine Rolle. Diese wurden bislang allerdings nicht genügend beachtet.

Die organisatorische Dimension läßt sich an einem Beispiel verdeutlichen: Integrierte Software-Entwicklungsumgebungen erlauben im Idealfall, die Softwareentwicklung eines Unternehmens mittels eines Prozeßmodells zu unterstützen und weiterzuentwickeln. Weniger innovative Lösungen beinhalten das Prozeßmodell in einer "fest verdrahteten Form". Im allgemeinen beschreibt ein

Prozeßmodell die Abläufe bei der Softwareentwicklung und die Beziehungen zwischen den Entwicklern. Passen die beschafften Werkzeuge bzw. die Software-Entwicklungsumgebung nicht zur (Ablauf- und Aufbau-)Organisation der Software-Entwicklungsabteilung, kommt es zu Problemen. Softwareentwicklung muß daher immer als *Ganzes* betrachtet werden: Software-Entwicklungsabteilungen stehen in engen Beziehungen zu anderen Unternehmensbereichen. Sie entwickeln und pflegen Anwendungen und unterstützen Anwender bei der Benutzung der zur Verfügung gestellten Systeme. Veränderungen in einem Teilbereich der Softwareentwicklung, wie beispielsweise in der Anforderungsanalyse für die Weiterentwicklung von Anwendungen, wirken sich immer auch auf die anderen Bereiche aus. Nur wenn man diese Beziehungen angemessen berücksichtigt, können Veränderungen erfolgreich vorgenommen werden. Wie bei einem Puzzle müssen auch in der Softwareentwicklung die einzelnen Teile zueinander passen.

Werkzeuge unterstützen die Anwendung von Methoden zur Softwareentwicklung. Sind die Anwender der Werkzeuge jedoch mit den Methoden nicht vertraut, können weder Qualitäts- noch Effizienzsteigerungen erwartet werden. Das Qualifikationsprofil des Software-Entwicklers stellt die personelle Dimension bei der Einführung der werkzeuggestützten Softwareentwicklung dar.

Ergebnisse

An diesem Punkt - der Weiterbildung des Softwareentwicklers - setzte das ISST an und erarbeitete im Rahmen des beschriebenen Projektes eine fünftägige Workshop-Reihe. Diese richtet sich an Praktiker - an Leiter von Software-Entwicklungsprojekten und -abteilungen, aber auch an das für diesen Bereich verantwortliche Management.

Das Seminar des ISST zeigt die Komplexität des Problems und zugleich auch individuelle Lösungswege auf. Die Teilnehmer werden in die Lage versetzt, die erforderlichen Änderungen in der DV-Organisation zu erkennen, geeignete Verfahren und Methoden einzuführen und Werkzeuge zu deren Unterstützung auszuwählen. Der erste Tag befaßt sich mit der Akquisition, Analyse und Verbesserung der DV-Organisation. Der zweite und dritte Tag behandeln die Anforderungsanalyse von Software-Produktionsumgebungen. Die beiden letzten Tage vermitteln den Teilnehmern die Kenntnisse, um Software-Produktionsumgebungen bewerten zu können.

Veröffentlichungen

[12], [13]

2.2.5 CASE-Kompetenzzentrum

Aufgabenstellung und Ziele

Eine Vielzahl von CASE-Werkzeugen wird derzeit am Markt angeboten. Für den Software-Entwickler ist diese Fülle oft schwer überschaubar. Prospektmaterialien der Hersteller und auch Werkzeugübersichten verschiedener Anbieter eignen sich nur bedingt zur Auswahl. Die Beschreibungen sind häufig zu allgemein und die Bewertungskriterien treffen oft auf den Einzelfall nicht zu.

Das ISST bietet hier im Rahmen seines CASE-Kompetenzzentrums potentiellen CASE-Anwendern einen hilfreichen Überblick über vorhandene Werkzeuge und Umgebungen.

Vorgehensweise

Die Bewertung der CASE-Werkzeuge erfolgt in zwei Schritten: Die analytische Bewertung basiert ausschließlich auf existierenden Dokumenten wie Prospektmaterial und Handbüchern. Ziel ist es, mit minimalem Aufwand festzustellen, inwieweit beispielsweise die geforderten funktionalen Anforderungen überhaupt unterstützt werden.

Bei der experimentellen Bewertung wird das Werkzeug einem Experiment unterzogen. Dieses stellt einen Ausschnitt einer zu erstellenden Software-Applikation dar. Um präzise und vergleichbare Bewertungsergebnisse zu bekommen, wird das Experiment durch konkrete Handlungsanweisungen für den Bewerter beschrieben.

Die Ergebnisse der Bewertung werden in einem Bericht dokumentiert, der nach bestimmten Aspekten unterteilt wird. Beispiele hierfür sind funktionale Vollständigkeit, Benutzerfreundlichkeit und Integriertheit. Zusätzlich werden bei jedem Werkzeug Angaben zum Hersteller und zu den Hardware- und Software-Voraussetzungen gemacht.

Ergebnisse

Zwei Berichte zum Stand der Werkzeugunterstützung sind in Vorbereitung. Der erste Bericht faßt die Ergebnisse zu den Methoden *Strukturierte Analyse* und *Entity-Relationship* zusammen, der zweite die Ergebnisse zu den Methoden *Strukturierter Entwurf* und *Modular Design*.

2.2.6 Software-Qualitätssicherung

Aufgabenstellung und Ziele

Was in anderen Ingenieurdisziplinen seit Jahren praktiziert wird, beginnt sich nun auch in der Software-Industrie durchzusetzen. Der zunehmende Druck auf Hersteller von Software im Bereich der Haftung für einwandfreies Funktionieren und Schadensersatz bei Mängeln hat dazu geführt, formalisierte Qualitätssicherungsmaßnahmen schon im Produktionsprozeß zu berücksichtigen. Qualitätssicherung soll sich also nicht mehr nur auf das Testen des fertigen Produktes beschränken; vielmehr sollen die ständige Kontrolle der Zwischenergebnisse und konstruktive Maßnahmen wie der Einsatz von Methoden und Werkzeugen helfen, die Produktqualität zu erhöhen. Basis dieser Bemühungen ist die Norm ISO 9000. Diese international gültige Norm definiert Mindestanforderungen an Qualitätssicherungssysteme. Unternehmen, die ein solches System etabliert haben, können sich durch externe Prüfer zertifizieren lassen. Das Zertifikat bietet Wettbewerbsvorteile bei der Bewerbung um Aufträge. Von öffentlichen Auftraggebern wird es in der Regel heute schon als notwendige Voraussetzung gefordert.

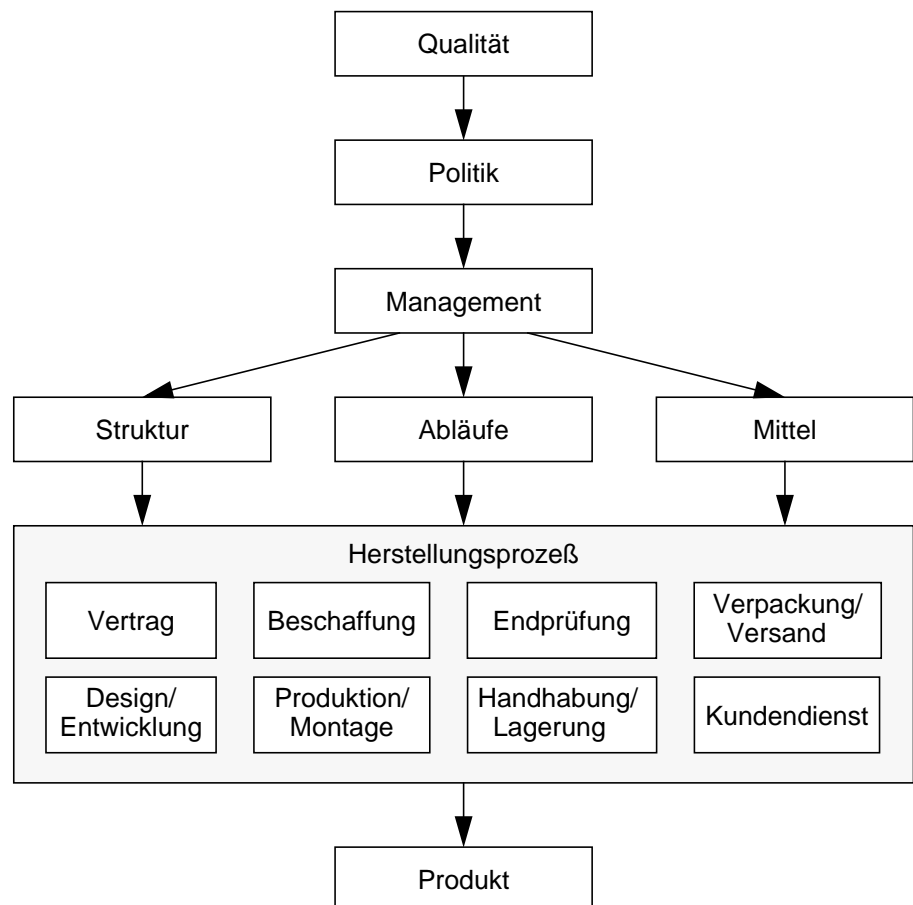


Abb. 4:
Prozeß der CASE-
Einführung

Vorgehensweise

Das ISST ist der Überzeugung, daß eine ISO 9000-konforme Qualitätssicherung nicht „von außen“ in ein Unternehmen eingebracht werden kann. Voraussetzung dafür ist, daß die Unternehmensleitung selbst die Notwendigkeit erkennt und die Ressourcen bereitstellt. In die Umsetzung der Maßnahmen sind dann alle Mitarbeiter eines Unternehmens eingebunden. Daher muß auch für eine möglichst umfassende Informationsverteilung gesorgt werden. Das ISST erarbeitet gegenwärtig Materialien, die die ISO 9000-Anforderungen allgemein zugänglich machen sollen. Ein erster Schritt in Richtung eines verbesserten Qualitätssicherungssystems ist dabei die Erfassung und Analyse des in einem Unternehmen praktizierten Vorgehens. Ist der initiale Zustand der ISO 9000-Konformität erreicht, muß durch ständige Kontrolle und Verbesserung das System optimiert werden.

Ergebnisse

Erste Informationsveranstaltungen zur ISO 9000-konformen Qualitätssicherung wurden durchgeführt.

2.3 Integrierte Software-Infrastrukturen

Zielsetzung des Bereichs

Eine wesentliche Grundlage für die Funktionsfähigkeit vieler Bereiche unserer Gesellschaft stellen „Infrastrukturen“ dar. Der Duden charakterisiert diese als den „wirtschaftlich-organisatorischen Unterbau einer hochentwickelten Wirtschaft“. Beispiele finden sich allenthalben: Verkehrs-Infrastrukturen wie Flughäfen, Bahnhöfe, Nahverkehrsnetz oder Straßen und Autobahnen, Kommunikations-Infrastrukturen wie Fernsehen, Telefon, Fax und Briefpost oder Gebäude-Infrastrukturen wie Frisch- und Abwassersysteme, Fahrstühle etc.

Infrastrukturen basieren auf Ingenieurkunst und Organisationsvermögen. Der Vorteil von Ingenieurtechniken ist, daß sie auf Erfahrungen zurückgreifen können, die über lange Zeiträume entstanden sind. Standardisierte Komponenten und Methoden zur Konstruktion von Infrastrukturen sind beispielhafte Ergebnisse langjähriger Arbeit.

Die Software-Technologie ist im Vergleich zu den klassischen Ingenieurwissenschaften (z.B. Architektur oder Maschinenbau) eine noch sehr junge Disziplin. Dennoch sind Software-Systeme inzwischen in nahezu allen Bereichen unseres Lebens vorhanden. Auch in kritischen Bereichen, in denen es um die Gesundheit und das Leben von Menschen oder deren Privatsphäre geht, spielen sie eine wichtige Rolle. Sicherheit und Zuverlässigkeit sind daher besonders bei Software-Systemen oberste Qualitätskriterien; gerade auch in Hinblick auf deren langfristigen Einsatz. Damit Software diesen Anforderungen standhalten kann, müssen in der Softwareentwicklung verstärkt auch ingenieurtechnische Prinzipien zur Anwendung kommen.

Moderne Software-Systeme zeichnen sich häufig durch einen hohen Komplexitätsgrad aus. Es ist absehbar, daß diese Komplexität in Zukunft noch zunehmen wird, da derzeit einzelne, isolierte Systeme mehr und mehr zu integrierten Gebilden verbunden werden. Diese werden - in Analogie zu den komplexen technischen Strukturen anderer Art - „informations- und kommunikationstechnische Infrastrukturen“ genannt.

Komplexe Software-Systeme lassen sich durch die folgenden Eigenschaften charakterisieren:

- Sie sind in der Regel verteilte Systeme.
- Sie sind aus älteren und neu entwickelten Komponenten zusammengesetzt.
- Sie unterstützen verschiedene Kategorien von Benutzern mit zum Teil völlig verschiedenen Anforderungen.
- Sie sind langlebige Wirtschaftsgüter.

Die Situation in der Software-Technologie ist gegenwärtig dadurch gekennzeichnet, daß eine beachtliche Zahl von Komponenten zur Konstruktion geeigneter Software-Infrastrukturen entwickelt worden ist (z.B. Datenbanken, Kommunikationsmechanismen, Funktionsbibliotheken). Die Integration dieser Komponenten zu problemadäquaten Infrastrukturen bereitet jedoch immer noch erhebliche Schwierigkeiten. Warum derzeit nur wenige integrierte Software-Infrastrukturen existieren, illustriert auch die folgende Abb. 5. Sie zeigt, welche

vielfältigen Aspekte bzw. Ebenen es bei der Integration von Software-Systemen zu berücksichtigen gilt:

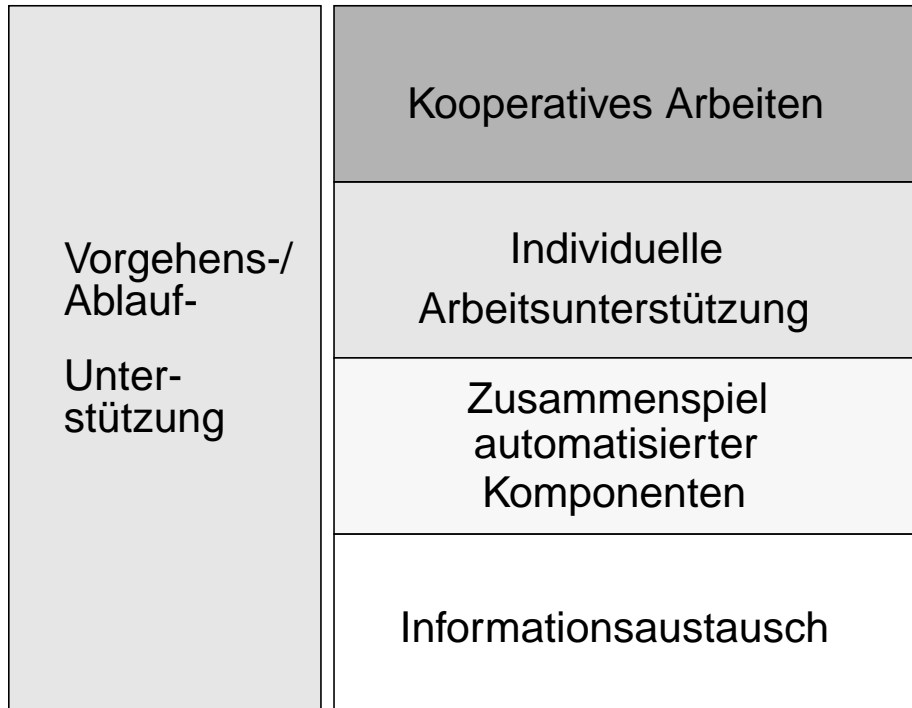


Abb. 5:
Integrationsebenen
in integrierten
Infrastrukturen

Arbeitsgebiete im Bereich Integrierte Software-Infrastrukturen

Im Fachbereich Integrierte Software-Infrastrukturen wurden 1993 Projekte in folgenden Arbeitsgebieten durchgeführt:

- Software Factory Engineering

Die Integration verschiedener Abteilungen und Arbeitsgruppen und der bereits vorhandenen informationstechnischen Systeme stellt für Unternehmen und Organisationen heute ein Hauptproblem dar. Dies gilt insbesondere auch für die IT-Abteilungen großer Institutionen. Deren Aufgabe ist nicht nur die Softwareentwicklung, sondern auch der Rechnerbetrieb, die Software-Wartung und Anwender-Unterstützung, d.h. Kundendienst, Hotline etc. Diese Tätigkeiten werden vom ISST unter der Metapher „Software Factory“ zusammengefaßt.

Das ISST entwickelt in diesem Arbeitsgebiet Konzepte zur Erstellung, Einführung und zum Betreiben von Software-Fabriken sowie entsprechender informations- und kommunikationstechnischer Infrastrukturen.

Sein Leistungsspektrum reicht von der Erhebung der fabrikspezifischen Anforderungen über die Integration von Werkzeugen zu einer Software-Fabrik bis hin zur Integration verschiedener Fabriken.

Die erstellten Konzepte und Lösungen werden dabei auch auf andere Anwendungsbereiche übertragen.

- Daten- und Objektmanagement

In modernen Unternehmen sind heute etwa drei bis vier unterschiedliche Datenbanksysteme neben verschiedenen Dateisystemen der eingesetzten Rechner anzutreffen. Die Bandbreite der benutzten Speichertechnologien reicht dabei von konventionellen Disketten und Festplatten bis hin zu CD-ROMs und optischen Wechselplatten. So unterschiedlich wie die eingesetzten Technologien sind auch die Anwendungen. Der Anwender muß dabei in der Regel wissen, in welchem Datenbanksystem sich die von ihm benötigten Informationen befinden, welcher Art (z.B. Tabellen, Text, Grafik, Bilder, usw.) und wie sie strukturiert sind. Ein wesentliches Ziel der Arbeiten auf dem Gebiet Datenhaltung besteht darin, den Benutzer von vielen der für ihn irrelevanten Details zu entlasten.

Basierend auf seiner langjährigen Erfahrung im Umgang mit konventionellen und objektorientierten Datenbanksystemen arbeitet das ISST in diesem Arbeitsgebiet an Lösungen zur Erweiterung, Migration und Koexistenz relationaler und eingeschränkt objektorientierter Systeme hin zu voll-objektorientierten Systemen. Die konzeptionellen Lösungen werden in Prototypen umgesetzt und in Anwendungsprojekten erprobt.

- Multimedia-Systeme

In diesem Arbeitsgebiet entwickelt das ISST Infrastruktur-Konzepte und -Lösungen für multimediale Arbeitsplätze. Ein Schwerpunkt liegt dabei auf verteilten multimedialen Systemen, die z.B. Konsumenten an verschiedensten Verkehrsorten mit aktuellen Informationen bedienen (z.B. Info-Terminals auf Bahnhöfen, Flughäfen usw.). Wichtigstes Beispiel hierfür sind sogenannte Point-of-Information (POI)/Point-of-Sale (POS)-Systeme. Die derzeit am Markt verfügbaren POI-/POS-Systeme befinden sich allerdings noch in einem Anfangsstadium, das durch einen geringen Grad der Integriertheit und hohe Betriebskosten gekennzeichnet ist.

- System- und Netzwerkmanagement

Die Komplexität der im Wirtschaftsleben bestehenden Rechnernetze nimmt beständig zu: Nicht nur die verwendeten Protokolle, auch die Netzwerk-Hardware-Plattformen sowie die verfügbare Software sind überaus heterogen. Das Betreiben von Anwendungen, die Installation und Konfigurierung von Systemen ist dadurch zu einer schwierigen Aufgabe geworden, die ohne automatische Unterstützung schwer oder gar nicht zu bewältigen ist. In diesem Arbeitsgebiet des ISST werden deshalb Konzepte zur Systemmodellierung, kooperativen Systemadministration und zum Konfigurationsmanagement erarbeitet und implementiert.

- Computergestützte Kooperation

Computergestützte Arbeitsplätze werden heute zunehmend miteinander verknüpft. Schon in naher Zukunft - davon kann ausgegangen werden - wird das gemeinsame Arbeiten am Bildschirm in den Arbeitsalltag Einzug halten.

Die dazu erforderlichen Hardware-Infrastrukturen sind entweder schon verfügbar oder kommen gegenwärtig auf den Markt. Im Software-Bereich befindet man sich dagegen noch im Experimentierstadium. Ziel dieses Arbeitsgebietes ist es, geeignete Infrastrukturen für die Zusammenarbeit von Menschen (*Computer Supported Cooperative Work - CSCW*), zu schaffen, zu erproben und in die Praxis einzuführen.

2.3.1 Software Factory Engineering

Aufgabenstellung und Ziele

Um die geforderte Qualität und Langlebigkeit von Software-Systemen sicherzustellen, muß deren Entwicklung, Weiterentwicklung, Pflege und Wartung auf der Basis ingenieurtechnischer Prinzipien erfolgen.

Als wesentliche Prinzipien sind die Strukturierung, die Standardisierung und die Evolutionsunterstützung zu nennen. Die Entwicklung von regulären, d.h. strukturell einfachen Infrastrukturen stellt einen wesentlichen Schritt in diese Richtung dar. Dies garantiert, daß ein System über mehrere Jahrzehnte hinweg mit vertretbarem Aufwand an veränderte Randbedingungen angepaßt werden kann.

Ein Konzept, das eine derart systematische Softwareentwicklung auf industrieller Basis ermöglicht, ist die „Software-Fabrik“. Darunter wird in der Regel der Unternehmensbereich verstanden, der für die Bereitstellung und Pflege aller Software-Systeme eines Unternehmens verantwortlich ist. Typische Aufgaben einer Software-Fabrik sind die Softwareentwicklung, die Auswahl und der Einsatz lizenzierter Software-Pakete, die Integration und Weiterentwicklung der eingesetzten Software-Systeme und die Betreuung der Anwender. Für alle diese Aktionen ist eine Fülle an interner und externer Kommunikation erforderlich, die im Sinne einer optimalen Ablauforganisation gezielt unterstützt werden muß.

Die Metapher der Software-Fabrik steht darüber hinaus auch stellvertretend für komplexe, verteilte Mehrbenutzer-Systeme, in denen Benutzer in verschiedenen Rollen agieren.

Das ISST befaßt sich in diesem Arbeitsgebiet mit der Entwicklung von Konzepten, Architekturen, Techniken und Methoden und deren Umsetzung in konkrete Systeme und Umgebungen zum systematischen Software Factory Engineering.

Vorgehensweise

Im Rahmen des EUREKA-Projekts ESF (*Eureka Software Factory*) ist unter Beteiligung des ISST ein Referenzmodell (*CoRe*) für Software-Fabriken entwickelt worden. Darin werden vier verschiedene Kommunikations- und Integrations-ebenen beschrieben. Diese sehen Integrationsrahmen vor, mit denen

- kooperatives Arbeiten (InterWorking),
- individuelle Arbeitsunterstützung (InterAction),
- das Zusammenspiel automatisierter Komponenten (InterOperation) und
- Informationsaustausch zwischen verschiedenen Subsystemen (InterConnection)

in integrierten Software-Infrastrukturen erreicht wird.

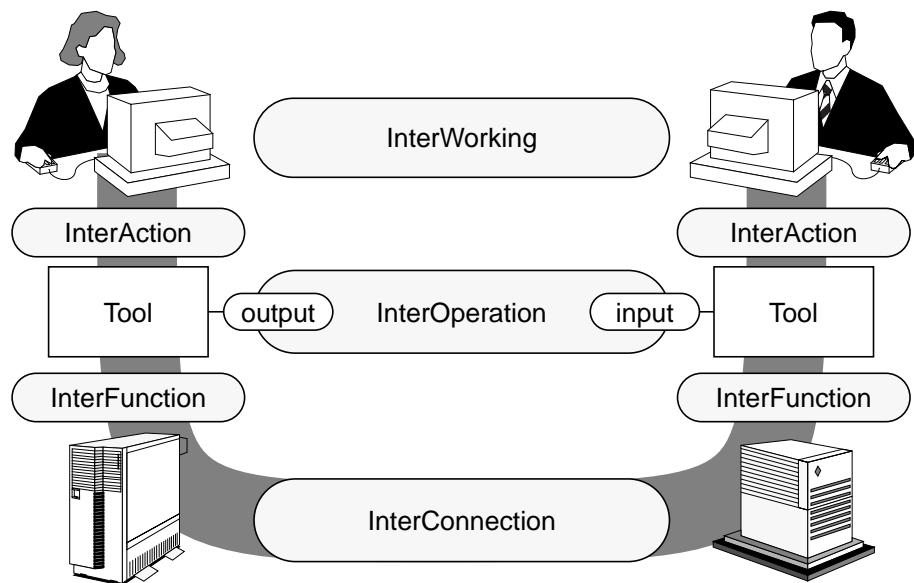


Abb. 6:
Integrationsebenen
und Schnittstellen
gemäß dem CoRe-Modell
für Software-Fabriken

Das ISST hat Konzepte und Lösungen für diese Integrationsrahmen entwickelt und erprobt.

Ergebnisse

Das ISST hat - in Kooperation mit der Universität Dortmund und der STZ GmbH - eine CoRe-konforme integrierte Infrastruktur entwickelt. Auf der Grundlage dieses „Kernel/2r“ können Software-Fabriken zusammengesetzt werden. Der Kernel/2r bietet Unterstützung bei der Integration von Fremdkomponenten und bei der Organisation ihrer Zusammenarbeit sowie bei der Definition und Überwachung von Vorgängen. Sowohl Gruppenarbeit als auch individuelles Arbeiten werden unterstützt. Die Integrationsplattform umfaßt eine Prozeßmanagement-Komponente (InterWorking) und verschiedene Benutzer-Interfaces (InterAction) für die beteiligten Benutzergruppen, die über einen Software Bus (InterOperation und InterConnection) integriert sind.

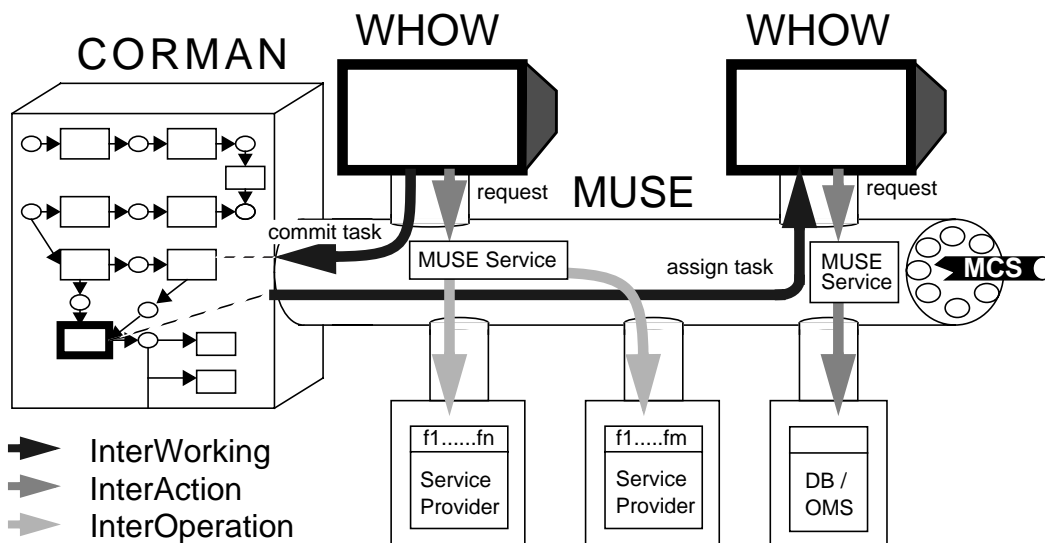


Abb. 7: Integrationsplattform Kernel/2r

Kernel/2r bildet die Basis für die IRIN-Software-Fabrik, die im Rahmen des Projekts COBRA-1 zur verteilten Softwareentwicklung erprobt wird (vgl. Arbeitsgebiet *System- und Netzwerkmanagement*). In IRIN (*ISST Research Instance*) evaluieren Arbeitsgruppen in Berlin und Dortmund die Möglichkeiten verteilter Softwareentwicklung unter Ausnutzung von Breitband-ISDN als Kommunikationsbasis.

Veröffentlichungen

[1], [19], [51], [52], [64]

2.3.2 Daten- und Objektmanagement

Aufgabenstellung und Ziele

In den letzten Jahren kommen in Unternehmen verstärkt relationale Datenbanken zum Einsatz. Diese bieten jedoch nur unzureichende Unterstützung bei komplexen Datenstrukturen, bei denen vielfältige Beziehungen zwischen den einzelnen Komponenten bestehen. Typische Anwendungsfälle, die von relationalen Systemen nur schlecht abgedeckt werden, sind Engineering-Tätigkeiten wie Software Engineering und CAD oder der Bereich der Multimedia-Anwendungen.

Objekt-orientierte Datenbanken unterstützen die Verwaltung der verschiedensten, auch komplexeren, Datentypen in einem System und bieten gleichzeitig Zugriffsoperationen auf die unterschiedlichen Daten an. Obwohl objekt-orientierte Datenbanken auf diese Weise eine Erleichterung für den Benutzer dar-

stellen und damit sicherlich auch einen Fortschritt repräsentieren, sind sie doch auch ein Glied in der Kette der Datenbanktechnologien. Bedingt durch die Evolution der Benutzer-Anforderungen koexistieren sie mit den bis dahin vorhandenen Datenbanken im jeweiligen Unternehmen. Das eigentliche Ziel, nämlich die Transparenz der eingesetzten Datenbank-Technologie, ist damit nicht erreicht.

Gegenwärtig ist der Trend zur Dezentralisierung bei gleichzeitiger Integration der vorhandenen Informationsinseln in ein unternehmensweites Informationsnetz zu beobachten. Entsprechend gilt es auch, unterschiedliche Datenbank-Technologien zu einem Gesamtsystem zu integrieren, das dem Benutzer in einheitlicher Form Datenhaltungsdienste für die verschiedensten Datentypen bereitstellt und dabei Aspekte wie Verteiltheit etc. verbirgt.

Vorgehensweise

Ausgehend von existierenden Standards wie relationalen Datenbanken und der zugehörigen Abfragesprache SQL oder dem in der Standardisierung befindlichen strukturell objekt-orientierten PCTE-System wurden neue Konzepte zur Erweiterung solcher Systeme hin zu voll objektorientierten entwickelt. Dabei wurden auf der einen Seite allgemeine Anforderungen berücksichtigt, wie sie im „Object-Oriented Database System Manifesto“ fixiert sind, auf der anderen Seite wurden auch spezifische Anforderungen betrachtet, die aus dem Anwendungsbereich an ein Datenbanksystem herangetragen werden. Selbstverständlich finden auch aktuelle Entwicklungen ihren Niederschlag. Entsprechende Aktivitäten in anderen Institutionen werden verfolgt.

Die Umsetzbarkeit der erarbeiteten Konzepte wurde, soweit dies möglich war, durch prototypische Realisierungen überprüft. Das gilt insbesondere auch für die Validierungsphase. Hier ist insbesondere die Entwicklung geeigneter Benchmarks zu nennen, da die Performanz von Datenhaltungssystemen zu den kritischen Kenngrößen zählt.

Um den Praxisbezug der Arbeiten sicherzustellen, wird in diesem Arbeitsgebiet generell versucht, so früh wie möglich mit Anwendern zu kooperieren. Nur dadurch ist gewährleistet, daß auch konkrete reale Anforderungen bei den Entwicklungen berücksichtigt werden.

Ergebnisse

Um den Anforderungen der neuen und ständig wachsenden Datenbank-Anwendungen, wie Softwareentwicklung oder Multimedia-Systemen, gerecht zu werden, wurde im ISST das objekt-orientierte Datenbanksystem InORM entwickelt. Die benötigten Operationen und Befehle werden in Form einer SQL-ähnlichen Sprache (EQL) zur Verfügung gestellt. Dies ist eine objekt-orientierte Erweiterung von SQL. EQL erweitert die Abfragesprache um die Definition und Manipulation von Typen, Instanzen und Komplex-Werten, zusammen mit erweiterbaren Funktionen und Operationen, welche den Typen zugeordnet sind. Als Typen sind sowohl Grundtypen, Komplex-Typen als auch abstrakte Datentypen möglich. Die Abfragesprache stellt die Operationen auf Objekten und relationale Operationen in einer integrierten Umgebung zur Verfügung.

Mit InORM wurde der Versuch unternommen, relationale und objekt-orientierte Fähigkeiten in einem System zu kombinieren. Dabei ging das ISST - statt von einem objekt-orientierten - von einem relationalen System aus.

Auf der Basis von InORM wurde das Multimedia-Datenbanksystem MediaBase entwickelt. Dieses unterstützt die Verwaltung komplexer Multimedia-Objekte, wie z.B. CDs inklusive Cover oder MPEG-komprimierter Videos.

Im Rahmen des EUREKA-Projekts ECMA PCTE hat das ISST ein Konzept für die Erweiterung des PCTE/OMS zu einem voll objektorientierten OOPCTE System entwickelt.

Der Vorschlag des ISST für die Definition und die Behandlung von Methoden lehnte sich eng an das Schema-Definition-Set (SDS)-Konzept des PCTE/OMS und versuchte, dessen Semantik auch auf die Ausführung von Methoden zu übertragen. Die Methoden und ihre Ausführung wurden in OOPCTE integriert, um ein homogenes System zu erhalten.

Die Resultate dieser Arbeiten werden in Standardisierungsarbeiten bezüglich der Objekt-Orientierung von PCTE bei den relevanten Gremien der ECMA (ECMA-TC33/TGOO) und der OMG (PCTE Special Interest Group) eingebracht.

Veröffentlichungen

[14], [26], [27], [28], [72], [72], [72], [72], [75]

2.3.3 Multimedia-Systeme

Aufgabenstellung und Ziele

Multimediale Systeme versprechen ein Massenmarkt der Zukunft zu werden. Gerade im Bereich der verteilten multimedialen Systeme zeichnet sich gegenwärtig eine rasante Entwicklung ab. Wichtigstes Beispiel dafür sind Point-of-Information (POI)- und Point-of-Sale (POS)-Systeme. Die dazu am Markt verfügbaren Systeme befinden sich allerdings noch in einem Anfangsstadium, das durch einen geringen Grad der Integriertheit und hohe Betriebskosten gekennzeichnet ist.

Ziel des ISST ist es, Infrastrukturen für verteilte (POI- und POS-)Systeme zu entwickeln, in denen Konsumenten- bzw. mobile Sachbearbeiter-Front-Ends mit einem Dienstleistungszentrum als Back-End online verbunden sind.

Dabei liegen die Schwerpunkte der Arbeiten auf der Identifizierung von Basis-komponenten und der Generalisierung von Dialogstrukturen für verschiedene Anwendungskategorien. Damit wird die Grundlage für die Vereinheitlichung der Kommunikation zwischen Front-Ends und dem Service-Center geschaffen. Auf dieser Basis ergibt sich die Möglichkeit, die Kooperation durch standardisierte Komponenten, Schnittstellen und Protokolle zu unterstützen.

Vorgehensweise

Ausgangspunkt der Arbeiten hinsichtlich eines Systemmodells für derartige Systeme ist das CoRe-Modell, das im Rahmen des Eureka Software Factory-Projektes als Referenzmodell für Software-Fabriken entwickelt worden ist. Dieses Modell charakterisiert zwar komplexe verteilte Umgebungen, Konsumenten-Front-Ends sind jedoch nicht berücksichtigt. An dieser Stelle setzten die Erweiterungen des ISST an.

Entsprechendes galt auch für die Architektur der Systeme. In ESF war zwar eine Referenzarchitektur für Software-Fabriken definiert worden, der Aspekt der relativ passiven Komponenten, wie sie Konsumenten-Front-Ends darstellen, mußte allerdings zusätzlich berücksichtigt werden.

In einem weiteren Schritt wird sich die Arbeitsgruppe der Identifizierung elementarer Komponenten und Dialogstrukturen widmen. Da dieses Problem kaum allgemeingültig zu lösen ist, werden wir uns auf Point-of-Information- bzw. Point-of-Sale-Lösungen konzentrieren. Da es sich hier nicht nur um eine rein software-technologische Fragestellung handelt, werden wir insbesondere den Punkt Dialogstrukturen in Kooperation mit kompetenten Institutionen angehen.

Die Umsetzbarkeit der Konzepte wurde anhand prototypischer Realisierungen überprüft.

Ergebnisse

Im Projekt TELIS wurden - auf der Basis des „Kernel/2r“ - Software-Infrastrukturen für integrierte Point-of-Information-/Point-of-Sale-Systeme prototypisch entwickelt. Diese Systeme bestehen aus multimedialen Front-Ends, die zur Präsentation von Informationen, Produkten und Dienstleistungen dienen und online mit einem Service Center verbunden sind. Das Dienstleistungszentrum stellt den Betrieb des Gesamtsystems sicher und sorgt für eine stete Aktualisierung des am Front-End dargestellten Angebots. Als Beispiel-Szenarien wurden ein System für Fahrkartenverkauf sowie ein integrierter Point-of-Information realisiert. Auch erste Erfahrungen hinsichtlich der Akzeptanz durch Benutzer und Betreiber konnten gesammelt werden.

Außerdem wurde mit dem Aufbau eines Multimedia-Labors begonnen, das die Umgebung zur Erstellung spezifischer Anwendungen bereitstellt. Dazu zählen sowohl Software-Komponenten, z.B. speziell für die multimediale Präsentation von Informationen, als auch spezielle Hardware wie Flachbettscanner oder Videokameras für die Realisierung von Videokonferenzen innerhalb einer Point-of-Information-Anwendung.

2.3.4 System- und Netzwerkmanagement

Aufgabenstellung und Ziele

In diesem Arbeitsgebiet lief zu Beginn des Berichtsjahres mit SINUS ein strategisches Rahmenprojekt an. Darin beschäftigte sich das ISST mit „Systemmanagement und -integration großer Netze und Systemumgebungen“. Im Rahmen von Teilprojekten werden konzeptionelle Grundlagen und prototypische Lösungen für die Systemadministration in mittleren und großen Organisationen erarbeitet. Die Schwerpunkte des Projektes liegen im Bereich Konfigurations- und Installationsmanagement. Weitere Arbeitsbereiche sind das Problemmanagement und die Verteilung und Bearbeitung von Systemnachrichten.

Ziel des Projektes SINUS ist es, eine Plattform sowie einen Satz von Werkzeugen und System-Infrastrukturen zur Verfügung zu stellen, die den Aufwand für die System- und Anwendungsadministration vereinfachen. In diesem Rahmen wird als globales Ziel ein System zur Systemmodellierung entstehen, das die einfache Modellierung eines komplexen Rechnernetzes werkzeuggestützt ermöglicht und dieses Modell automatisch in korrespondierende Konfigurationen und Installationen umsetzt.

Vorgehensweise

Das Projekt SINUS wird im Rahmen mehrerer Teilprojekte umgesetzt. Anforderungen dazu kamen zum Teil aus dem eigenen Hause, aber auch aus anderen Projekten (vgl. die folgenden Projekte COBRA-1, LIT-SAZ).

Die Problemlösung im Projekt SINUS erfolgt stufenweise. Die Prozesse im einzelnen: Der Konfigurationsentwurfs-Prozeß (siehe Abb. 8) beschreibt die Teilprojekte der zweiten Phase des Projektes: Hier können komplexe Systeme, bestehend aus tausenden Komponenten, modelliert und die notwendigen Konfigurationen und Installationen abgeleitet werden. Der Konfigurationsimplementierungs-Prozeß (siehe Abb. 9) beschreibt die Teilprojekte der ersten Phase. Er soll - soweit vorhanden - existierende Produkte verwenden.

Ergebnisse

Projekt HOTCON (abgeschlossen):

Im Rahmen des SINUS-Teilprojektes HOTCON (*Hotline und Consulting System*) wurde ein System (Prototyp) entwickelt, das Benutzer in Problem- und Fehlersituationen unterstützt. Es kann sowohl für einen Hotline-Betrieb (*User Help Desk*) mit einer Fehlerdatenbank als auch als Consulting-System eingesetzt werden. Die Anwendung ist nicht nur in lokalen Umgebungen, sondern auch bei geographisch verteilten Rechnerarbeitsplätzen möglich.

Zum Konzept: HOTCON ist ein datenbankbasiertes System. Es erlaubt einen Dialog zwischen einem Benutzer und einem Berater. Dialoge bestehen aus

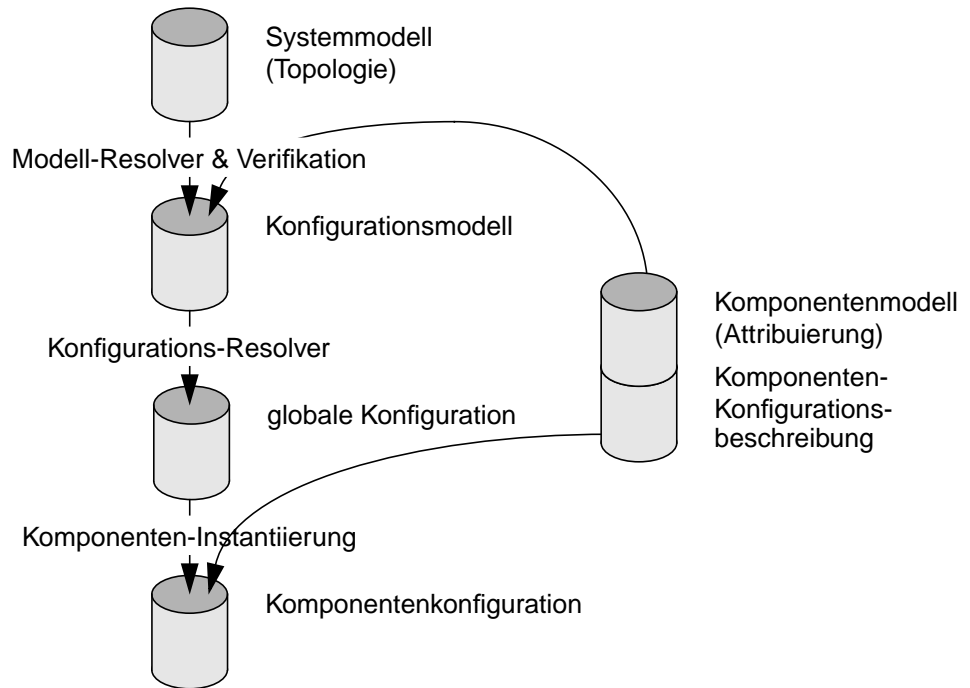


Abb. 8:
Konfigurations-
entwurfs-Prozeß

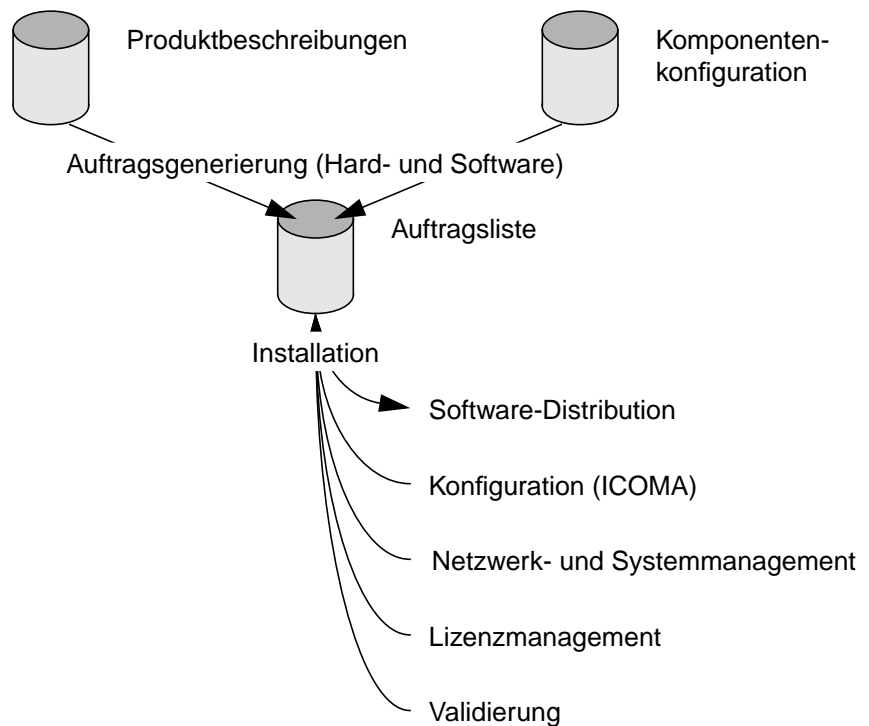


Abb. 9:
Konfigurations-
implementierungs-
Prozeß

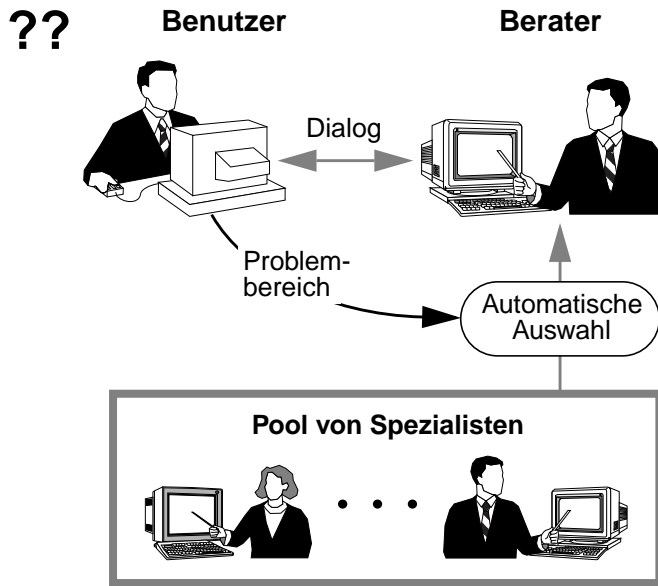


Abb. 10:
Problemlösungs-
prozeß

mehreren Elementen, die entsprechend attribuiert werden können (z.B. Frage, Antwort, Mitteilung etc.).

Hat ein Benutzer ein Problem, so wird ihm automatisch vom System ein Berater aus einem Pool von Spezialisten zugeordnet (siehe Abb. 10). Diese werden nach bestimmten Themengebieten und Prioritäten klassifiziert. Ist zur Zeit kein Spezialist im System angemeldet, wird das Problem so lange gespeichert, bis ein Spezialist zur Verfügung steht.

Nach der Beendigung eines Dialoges kann der Berater die Lösung des Problems in die Problemdatenbank übernehmen. Diese kann sowohl von den Benutzern als auch von den Spezialisten verwendet werden, um nach bereits vorhandenen Lösungsmöglichkeiten zu recherchieren (siehe Abb. 11).

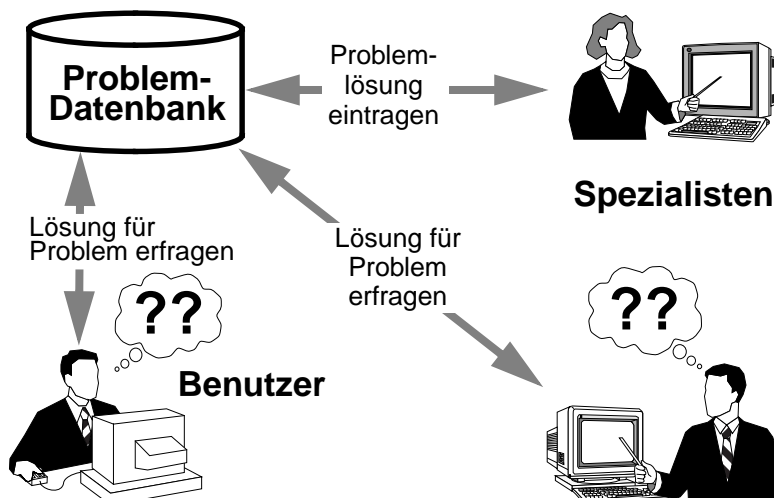


Abb. 11:
Unterstützung durch
die Problemdatenbank

Die Kernbausteine des HOTCON-Systems sind:

- eine grafische Benutzungsschnittstelle für das Führen von Dialogen und zum Suchen in der Problemdatenbank und
- eine Problemdatenbank, in der die Problemlösungen zur Wiederverwendung zur Verfügung stehen.

Projekt ICOMA (Laufzeit offen):

Das Projekt „Integrated Configuration Management“ (ICOMA) ist das bisher wichtigste Teilprojekt im Rahmen von SINUS. Das ISST erarbeitet darin Konzepte und prototypische Entwicklungen zur Konfiguration von verteilten Systemen. Ziel des Projektes ist es, die verschiedenen Standards wie IETF/SNMP, OSI/X.500, OSI/CMIP, DMTF/DMI, SUN/NIS+ und applikationsspezifische Methoden zu unterstützen und zu integrieren. Zielgruppen eines solchen Konfigurationssystems sind mittlere und große Organisationen, die mehrere Standorte bzw. selbständige Unternehmenseinheiten haben, d.h. räumlich „verteilt“ arbeiten.

Das ICOMA-System kann von einem zentralen Administrationssystem aus das gesamte verteilte System konfigurieren. Dabei ist für die Administratoren transparent, wie das jeweilige *Managed Object* (MO) implementiert ist. Die Konfigurationsaufgaben können auf mehrere Administratoren aufgeteilt werden. Ein Rollenkonzept definiert die Kompetenzen eines Administrators. Der Konfigurationsprozeß des ICOMA-Systems ist soweit anpaßbar, daß selbst komplexe Konfigurationen mit minimalem Administrationsaufwand realisierbar sind.

ICOMA bietet eine verteilte Konfigurationsmanagementplattform (siehe Abb. 12) in Form des Konfigurationsbusses. Dieser erlaubt, *Managed Objects* und Management-Anwendungen für das Konfigurationsmanagement zu integrieren. Der Konfigurationsbus ermöglicht dem Anwender, eigene und fremde

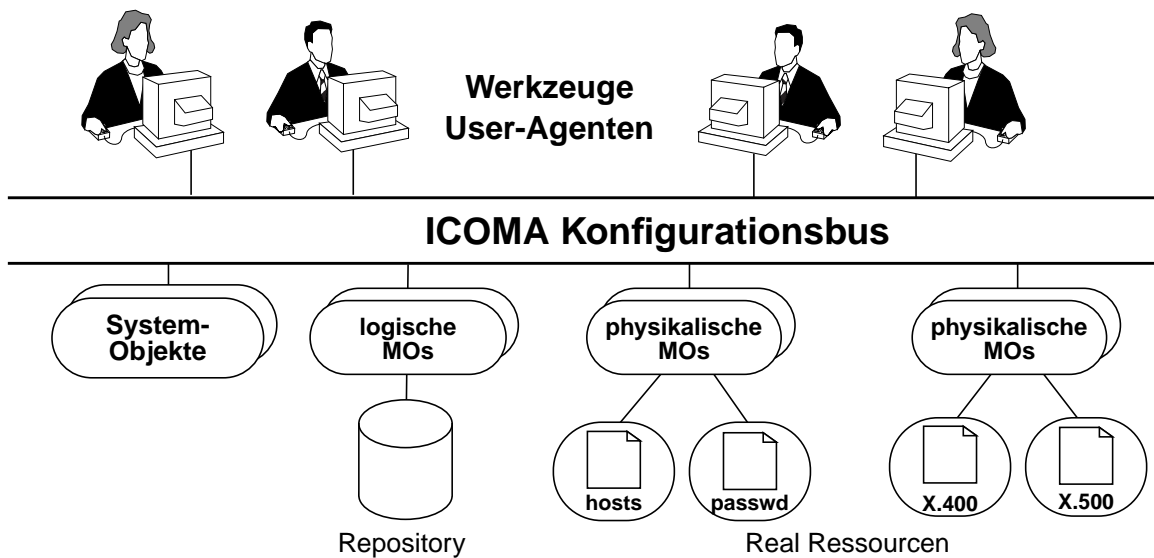


Abb. 12:
Architektur
des ICOMA-
Systems

Managed Objects für seine Systeme und Anwendungen zu entwickeln und zu verwenden. Hierbei sollen insbesondere die Agenten und Objekte von IETF/SNMP, OSI/CMIP, OSI/X.500, DMTF/DMI und SUN/NIS+ berücksichtigt werden.

Weiterhin ist es möglich, das System sowohl organisations- als auch administratorspezifisch mittels wiederverwendbarer Bausteine und einer eigenen Konfigurationssprache einfach anzupassen. Dieses unterstützt die Modellierung von Konfigurationssystemen auf der Basis von „Enterprise Policies“ und „Administrator Views“.

Bisherige Projektergebnisse:

Im Berichtsjahr wurden vom ISST ein Konfigurationsbus, ein MO-Editor, ein ICOLA-Compiler und einige *Managed Objects* implementiert. Der Konfigurationsbus erlaubt es, die *Managed Objects* in das ICOMA-System zu integrieren. Bisher steht dafür ein Adapter zur Verfügung, der die MO per Shellskripten verwaltet. Der Adapter verwendet Textdateien als Repository. Der Editor kann für beliebige MO verwendet werden, die in den Konfigurationsbus integriert wurden. Die Sicht eines Administrators auf seine zu administrierenden *Managed Objects* ist frei definierbar. Die Maske zur Dateneingabe für die MO kann ebenfalls durch den Administrator angepaßt werden. Die ICOLA-Sprache erlaubt die einfache Implementierung von *Managed Objects*, wobei bisher als Zielsprache die Shellskripte für den vorhandenen Adapter vorgesehen sind.

Veröffentlichungen zum Projekt: [5], [18]

Projekt LIT (abgeschlossen):

Das Land Berlin ist - in Hinblick auf seine Funktion als Hauptstadt und zukünftiger Regierungssitz - im Begriffe, seine Verwaltung zu modernisieren. Administrative Aufgaben sollen vor allem durch den Einsatz modernster Informationstechnologie besser und effizienter bearbeitet werden. Um die dafür erforderlichen IuK-Systeme auch langfristig optimal verwalten zu können, begann das Landesamt für Informationstechnik (LIT) Berlin 1993 mit dem Aufbau eines Service- und Administrationszentrums (SAZ). Mit diesem sollen sich die System- und Netzwerkmanagementaufgaben der Berliner Verwaltung kostengünstig realisieren lassen. Das ISST erhielt hierbei die Aufgabe, Lösungskonzepte für ein derartiges Administrationszentrum zu erarbeiten.

Das Institut hat zu diesem Zweck gemeinsam mit einer Projektgruppe des LIT ein Konzept für ein Systemadministrationszentrum (SaZ) entwickelt (MINZE-Studie). Dieses beschreibt, wie bis zum Jahr 2004 der Großteil der Rechner des Landes Berlin verwaltet werden kann und welche organisatorischen Voraussetzungen geschaffen werden sollen. Das Konzept sieht eine verteilte Administration der Rechner vor, d.h. ein Teil der Aufgaben wird - je nach Kosten - zentral im SaZ bzw. vor Ort beim Anwender in einem lokalen Administrationszentrum (LaZ) wahrgenommen. Das SaZ soll insbesondere Massenaufgaben, die für alle weitgehend identisch sind, sowie hochspezialisierte Aufgaben des Systemmanagementbereichs durchführen. Es wurden insbesondere die fünf Funktionsbereiche des ISO³/OSI⁴-Modells betrachtet: Konfigurations- (inkl. Installations-), Fehler-, Leistungs-, Sicherheits- und Abrechnungsmanagement.

In einem zweiten Schritt erarbeitete das ISST ein Pflichtenheft und einen Kriterienkatalog. Beides bildete die Grundlage für eine EG-weite Ausschreibung zur Realisierung des SaZ. Das Pflichtenheft enthält: einen allgemeinen Teil, der ergänzend zur MINZE-Studie die gewünschte Funktionalität des SaZ beschreibt, sowie einen umfangreichen Fragenkatalog, der die wesentlichen funktionalen Details erhebt und zur Bewertung eines Angebotes dient. Der Kriterienkatalog beinhaltet zu jeder Frage bzw. jedem Fragenkomplex einen Bewertungsvorschlag, der in Absprache mit der LIT-Projektgruppe ausgearbeitet wurde. Die endgültige Gewichtung der Fragen wurde durch den Auftraggeber selbst, das LIT, festgelegt.

Zusammen mit der Firma IBM, die den Zuschlag für den Auftrag erhielt, wurde dann ein Feinkonzept erstellt, das alle zu implementierenden Funktionen detailliert beschreibt. Das ISST wird dem Unternehmen auch über das Projekt hinaus beratend zur Seite stehen.

Ferner erstellte das ISST eine Wirtschaftlichkeitsbetrachtung, die der Berliner Senat als Entscheidungsgrundlage für Kostenkalkulationen heranzieht. Sie enthält eine Systematik, wie die personellen und investiven Aufwände für verschiedene Modelle der Systemadministration berechnet werden können und stellt für einige auserwählte Administrationsformen eine Kostenrechnung auf. Die Studie des ISST ergab, daß das SaZ/LaZ-Modell die wirtschaftlichste Form ist.

Im Rahmen dieses Projektes führte das ISST im Berichtsjahr Schulungen (UNIX und TCP/IP; Informationsdienste, SNMP-Management, Fehlermanagement, OSI-Management, Verteilte Systeme) für die Mitarbeiter des LIT durch.

Veröffentlichungen zum Projekt: [59], [63], [65]

Projekt COBRA-1 (Laufzeit bis März 1995):

Das Projekt Cobra-1 (*Co-Operation on Bureau, Research and Administration*) ist ein FhG-weiter Feldversuch, in dem moderne Kommunikationsinfrastrukturen aufgebaut und erprobt werden sollen, die insbesondere die Zusammenarbeit zwischen den Stamminstituten und ihren Außenstellen unterstützen. Das spezifische Interesse des ISST ist es, eine kooperative, über seine beiden Standorte verteilte Softwareentwicklung zu ermöglichen. Dieses Anwendungsszenario besteht aus drei Teilen:

Das erste Teilszenario vereinfacht die verteilte Netzwerk- und Systemadministration und ermöglicht ein kooperatives Management beider Standorte. Weiterhin wird eine erweiterte Systemumgebung zur Verfügung gestellt, um den Datenaustausch und die Arbeitsumgebung über beide Standorte transparenter zu gestalten. In diesem Rahmen sollen weitere Anwendungen, die zunächst nur an einem Standort existierten, an beiden Orten zugänglich werden. Darüber hinaus sollen - als Basis für die anderen Szenarien - multimediale Anwendungen und Dienste installiert werden.

Die Installation und der Betrieb verteilter Systeme bringt Probleme mit sich (Programm- und Systemfehler, Inkompatibilitäten usw.). Das zweite Szenario

3. *International Standard Organisation*

4. *Open Systems Interconnection*

unterstützt deshalb ein verteiltes Problemmanagement. Anwender können z.B. bei Problemen oder Fehlern diese an den Support für die entsprechende Anwendung schicken. Das eingesetzte HOTCON-System (vgl. Projekt HOTCON) erlaubt dem Spezialisten des Supports, einen Dialog mit dem Anwender zu führen. Dieser Dialog kann sowohl On- als auch Offline geführt werden. Die auszutauschenden Informationen können multimediale Daten wie Bilder, Sprachaufzeichnungen oder Videos enthalten. Alle ausgetauschten Informationen werden in Form eines Protokolls gespeichert und sind für beide Teilnehmer jederzeit verfügbar. Dieses Protokoll könnte vom Spezialisten weiter verarbeitet werden, um es z.B. in einer Datenbank dauerhaft zu speichern. Das HOTCON-System kann auch in der Entwicklungsphase eingesetzt werden, um Absprachen zu treffen.

Die skizzierten Probleme lassen sich im Dialog zwar genauer eingrenzen, sind dadurch aber in der Regel noch nicht gelöst. Ein Fehler muß in diesem Fall in Form eines *problem report* archiviert werden und durchläuft anschließend einen komplexen Bearbeitungsprozeß, in den Mitarbeiter beider Standorte involviert sein können. Zur Realisierung dieses Szenarios wird die vom ISST entwickelte Umgebung „Kernel/2r“ eingesetzt, eine komplette Infrastruktur zum Aufbau verteilter Systeme (vgl. Arbeitsgebiet *Software Factory Engineering*). Die automatische Steuerung eines formal definierten Prozesses koordiniert die Arbeit der involvierten Personen und den damit verbundenen Objektfluß. Jedem Mitarbeiter steht eine einheitliche, jedoch individuell anpaßbare Interaktionsumgebung für die Erledigung seiner Aufgaben zur Verfügung. In die Arbeitsplatzumgebung sind Konferenzwerkzeuge integriert, die zusätzlich die spontane Online-Kommunikation zwischen den Beteiligten ermöglichen.

Die aufgebaute Infrastruktur und die Szenarien sollen innerhalb des Feldversuches getestet und bewertet werden. Ziel des Projektes ist es, diese Umgebung auch nach dem Projekt weiter zu betreiben. Dazu müssen nicht nur die Ziele erfüllt, die Umgebung muß auch unter Kostengesichtspunkten rentabel betreibbar sein.

Die Ergebnisse und Erfahrungen von Cobra-1 sollen in zukünftige Projekte mit Industriepartnern - insbesondere kleineren und mittleren Unternehmen - einfließen. Sie sollen helfen, Kosten zu sparen - z.B. für Dienstreisen - und die Zusammenarbeit im Rahmen verteilter Projekte qualitativ zu verbessern.

Das Projekt wird in enger Kooperation mit dem Vertragspartner Siemens Nixdorf Informationssysteme (SNI AG) durchgeführt. Die Firmen SNI und Sie-tec installieren ein Wide Area Netzwerk auf der Basis einer 2 Mbit ISDN-Verbindung, die die Institute direkt miteinander verbindet. Weiterhin werden eine Dienstplattform für elektronische Post (X.400), Direktory-Services (X.500) und Tele-Konferenzen aufgebaut. Das ISST installiert darauf aufbauend ein kooperatives Systemmanagement für beide Institutsteile inklusive einer verbesserten Systemumgebung. „Kernel/2r“ wird vom ISST Dortmund auf die COBRA-Rahmenbedingungen angepaßt und zur Koordination des verteilten Software-Entwicklungsprozesses eingesetzt. Das ISST Berlin realisiert ein elektronisches Problem-Management auf der Basis seines HOTCON-Systems.

Während des Feldversuches werden statistische Daten erhoben, die die Bewertung der technischen Infrastruktur ermöglichen. Darüber hinaus werden mit Hilfe von Fragebögen die Erfahrungen der Teilnehmer des Feldversuches erfaßt.

Die Ergebnisse von Cobra-1 werden im März 1995 vorliegen. Sie sollen Aufschluß darüber liefern, welche Verbesserungen die Cobra-1-Infrastruktur für das ISST gebracht hat. Anhand der Schwachstellenanalyse soll die Infrastruktur soweit verbessert werden, daß sie in den täglichen Betrieb dauerhaft integriert werden kann.

2.3.5 Computergestützte Kooperation

Aufgabenstellung und Ziele

Die computergestützte Zusammenarbeit in örtlich verteilten Gruppen wird schon in naher Zukunft eine große Bedeutung einnehmen, da sich damit wichtige Wettbewerbsvorteile geltend machen lassen. Voraussetzung dafür ist, daß die in Betrieben und öffentlichen Organisationen vorhandenen EDV-Systeme angemessen vernetzt werden. Dabei ist auch darauf zu achten, daß Computer die zwischenmenschlichen Arbeitsprozesse flexibel unterstützen. Dies ist der Hintergrund des im Oktober 1993 angelaufenen Großprojektes WAM (*Wide Area Multimedia Group Interaction*). Dessen Ziel ist die Realisierung eines Software-Systems, das die örtlich verteilte Zusammenarbeit in öffentlichen Verwaltungen über Breitbandkommunikation unterstützt.

Vorgehensweise

Zur Steuerung der elektronisch unterstützten, räumlich getrennten Vorgangsbearbeitung entwickelt das ISST ein anwendungsspezifisches Workflow-Management-System. Ausgehend von einem typischen Szenario der öffentlichen Verwaltung werden wichtige Bearbeitungsabläufe analysiert und in einzelnen Prozeßmodulen spezifiziert („Prozeßmodellierung“). Diese unterstützen als eigenständige Komponenten individuelle Arbeitsvorgänge. Durch das Zusammenstecken von Prozeßmodulen wird analog zu Software-Modulen eine komplexe, verteilte Vorgangsbearbeitung realisiert. Dieses Modulkonzept hat den Vorteil, daß sich Änderungen in den Verwaltungsprozeduren nur auf einzelne Module auswirken. Das Zusammenstecken erfolgt dabei gemäß den Entscheidungen der betroffenen Mitarbeiter und in Übereinstimmung mit der aktuellen Arbeitssituation, d.h. es wird kein Modell im voraus für den gesamten Arbeitsprozeß erstellt. Ist das System bedarfsgerecht konzipiert, sorgen dann in der Durchführungsphase eigene Interpreten dafür, daß anhand des Prozeßmodells die einzelnen Arbeitsschritte den beteiligten Personen am EDV-Arbeitsplatz vorgeschlagen werden.

Da die Prozeßmodule lediglich Arbeitsaufgaben definieren, kann der einzelne Mitarbeiter frei bestimmen, welche Lösung in der jeweiligen Situation die beste ist. Dazu hat er zwei Möglichkeiten: Er entscheidet sich selbständig für ein vorhandenes Prozeßmodul, das dadurch mit der Aufgabe assoziiert wird; oder er bearbeitet die Aufgabe gemeinsam mit anderen Mitarbeitern. Auf diese Weise entstehen Beziehungen zwischen verschiedenen Beteiligten, die sich über Sprechakte zur Bearbeitung von Aufgaben verpflichten. Die Mitarbeiter können sich dann ihrerseits zur Lösung der Aufgabe für ein Prozeßmodul oder für eine

Zusammenarbeit mit anderen Personen entscheiden. Durch diese kooperative Bearbeitung von Aufgaben werden die Arbeitsabläufe innerhalb einer Organisation vernetzt. Sprechakte werden als wesentliche Basis dieser zwischenmenschlichen Interaktionen angemessen berücksichtigt.

Jeder Aufgabe werden zudem die benötigten Dokumente, z.B. Anträge, Verträge etc., aber auch alle Durchführungsanleitungen zugeordnet. In Form einer „elektronischen Mappe“ stehen diese Dokumente den Mitarbeitern zu jeder Zeit an ihrem Arbeitsplatz zur Verfügung. Darüber hinaus wird die „Bearbeitungsgeschichte“ der einzelnen Dokumente festgehalten. So kann rückverfolgt werden, wie Entscheidungen zustande kamen.

Das ISST-Konzept zur räumlich verteilten Vorgangsbearbeitung stellt einen Rahmen für eine einheitliche Arbeitsumgebung dar; dieser kann entsprechend den individuellen Bedürfnissen der Organisation ausgefüllt werden. Für die öffentliche Verwaltung sind das z. B. in Geschäftsordnungen festgelegte Prozeduren. Das Workflow-Management-System des ISST erlaubt außerdem, Videokonferenzen zu integrieren. Dies ermöglicht Arbeitsbesprechungen auch ohne aufwendige Dienstreisen. Da die Übertragungskosten für Breitbandkommunikation noch sehr hoch sind, müssen Videokonferenzen effizient und zügig durchgeführt werden. Auch hier stellen Prozeßmodelle sicher, daß alle erforderlichen Prozeduren samt Unterlagen verfügbar sind, um keine Wartezeiten während der Videokonferenz entstehen zu lassen.

Ergebnisse und Ausblick

Das WAM-Projekt, die Realisierung einer verteilten multimedialen Vorgangsbearbeitung mit integrierten Videokonferenzen, befindet sich noch in der Entwurfsphase. Mit einer „Teststrecke“ zwischen den ISST-Institutsteilen in Berlin und Dortmund sollen die Praktikabilität des Konzepts und die Vorteile des modularen Prozeßmodellansatzes unter realen Bedingungen demonstriert werden. Anschließend werden aufgrund der Erfahrungen die einzelnen Prozeßmodule weiterentwickelt und das Konzept verfeinert.

Konzept und Realisierung sind so allgemein ausgelegt, daß die Ergebnisse des WAM-Forschungsvorhabens auf fast jedes Gebiet, bei dem mehrere Personen zusammenarbeiten, angewendet werden können.

2.4 Prozeß-Engineering

Zielsetzung des Bereichs

Der Markt fordert von den Unternehmen, daß sie sich flexibel an aktuelle Kundenwünsche anpassen können. Wettbewerbsvorteile hat dabei vor allem, wer schnell auf Nachfrageveränderungen reagieren kann. Darüber hinaus wird von Unternehmen verlangt, problemspezifische Lösungen anbieten zu können. Ein vielfältiges, diversifiziertes Produkt- und Dienstleistungsangebot ist gefragt.

Wie die unternehmensspezifische Angebotspalette, so sollten auch die Geschäftsprozesse eines Betriebes bzw. einer Organisation maximale Anpassungsfähigkeit erlauben. Sie sind das Herzstück unternehmerischer Tätigkeit und verdienen im Sinne eines Total Quality Managements besondere Beachtung. Dabei steht im Vordergrund, die Fülle oft schwer überschaubarer Geschäftsaktivitäten in einem Unternehmen zu koordinieren.

Eine softwaretechnische Methode zur gezielten Gestaltung von Arbeitsvorgängen stellt das systematische Prozeßmanagement dar. Das ISST widmet sich dieser Aufgabe in einem eigenen Fachbereich.

Eine systematische Behandlung von Geschäftsprozessen ist im wirtschaftlichen Leben und bei der Entwicklung komplexer Produkte von erheblicher Bedeutung. Geschäftsprozesse wie beispielsweise in Betrieben, Verwaltungen, Banken und Versicherungen laufen in der Regel arbeitsteilig ab; d.h. viele Sachbearbeiter arbeiten gemeinsam an komplexen Aufgaben. Wird diese Kooperation nicht hinreichend unterstützt, kann es leicht zu Problemen in der Ablaufor-

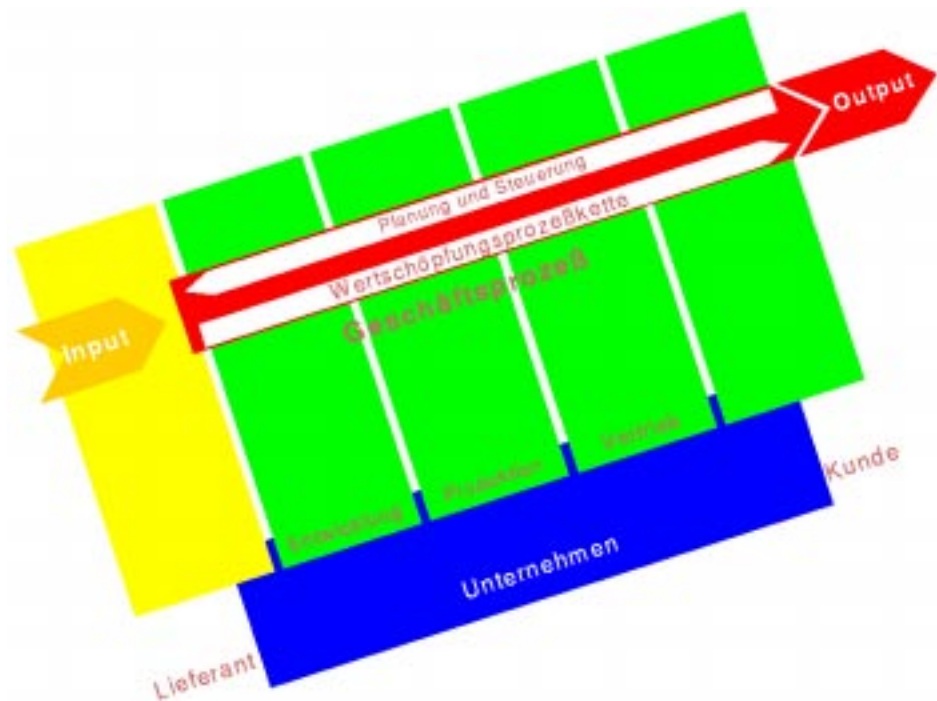


Abb. 13:
Geschäftsprozesse:
Kernfunktionen eines
Unternehmens

ganisation kommen. Hier setzt das systematische Management von Geschäftsprozessen an: Es bietet die Möglichkeit, zwischenmenschliche Arbeitsbeziehungen zu koordinieren und damit Reibungsverluste zu minimieren.

Im heutigen Arbeitsalltag ist zu beobachten, daß viele Geschäftsprozesse zwar initial geplant werden, diese ursprüngliche Planung jedoch schnell obsolet ist. So verlaufen Prozesse häufig anders als beabsichtigt, überschreiten Zeit- und Kostenlimits, Beteiligte halten sich nicht an die Planungen und die zwischenmenschlichen Interaktionen sind oft durch Kommunikationsschwierigkeiten geprägt.

Am Bereich Prozeß-Engineering des ISST wurden Techniken entwickelt, die dazu beitragen können, diese Nachteile zu überwinden. Die Vorgehensweise ist folgende: In einem ersten Schritt werden die Abläufe im Unternehmen explizit gemacht, d.h. beschrieben. Dies geschieht mittels „Prozeß-“ bzw. „Vorgangsmodellen“, die geeignet sind, Geschäftsaktivitäten in ihrer gesamten Komplexität abzubilden. Im einzelnen lassen sich definieren: die Inhalte von Aktivitäten (z.B. was ist zu tun, welche Werkzeuge werden eingesetzt), deren zeitliche Abfolge (erfolgen Aktivitäten nacheinander in einer bestimmten Reihenfolge oder parallel zueinander), Verantwortlichkeiten (wer ist wofür zuständig) sowie anvisierte Geschäftsziele. Der so modellierte Vorgang - in der Regel werden die Modellinformationen in mehrmaligen Durchgängen und enger Abstimmung mit den Prozeßbeteiligten erhoben - kann anschließend auf verschiedenste Kriterien überprüft werden: Plausibilität, Konsistenz, spezielle Prozeßeigenschaften wie Engpässe, kritische Pfade, Personal-/Zeitbedarf usw.

Über die reine Modellierung hinaus sieht das systematische Prozeßmanagement auch vor, daß Vorgänge gesteuert, d.h. rechnergestützt, ablaufen. Das garantiert, daß sich Geschäftsvorgänge nur entsprechend dem ihnen zugrundeliegenden Modell und damit gemäß den zuvor aufgestellten Richtlinien und Dienstanweisungen verhalten; daß automatische Aktivitäten ohne Verzögerungen durchgeführt werden; und daß die am Prozeß beteiligten Personen jederzeit wissen, welche Aktivitäten zu bestimmten Zeitpunkten sinnvoll bzw. unangebracht sind.

Der Vorteil der am ISST entwickelten Methode ist, daß sie zu jedem Zeitpunkt flexible Änderungen erlaubt. Treten also gewisse Schwachstellen des Vorgangsmodells erst bei der Ausführung des kooperativen Prozesses zutage, kann unverzüglich mit einer entsprechenden Modelländerung auf die Anforderungen der Praxis reagiert werden.

Prozeßmodelle machen Geschäftsvorgänge den beteiligten Personen verständlich und transparent und zeigen vorhandene Probleme auf. Außerdem stellt das systematische Management von Geschäftsprozessen eine wichtige Maßnahme zur Qualitätssicherung dar: Nicht erst das fertige Produkt, d.h. das Ergebnis eines Prozesses, wird geprüft, sondern bereits der komplette Entwicklungsprozeß. Damit wird die Voraussetzung geschaffen, daß auch das Endprodukt qualitativ hochwertig ist. Überdies lassen sich Fehlentwicklungen und Ressourcenverschwendung frühzeitig vermeiden.

Arbeitsgebiete im Bereich Prozeß-Engineering

Die Aktivitäten am Bereich Prozeß-Engineering erstreckten sich 1993 auf die beiden Arbeitsgebiete „Prozeßmanagement-Werkzeuge“ und „Management kundenspezifischer Geschäftsprozesse“.

2.4.1 Prozeßmanagement-Werkzeuge

Aufgabenstellung und Ziele

In diesem Arbeitsgebiet konzentrierte sich das ISST auf die Entwicklung eines „Werkzeugkastens“ zum systematischen Management von Geschäftsprozessen (*Prozeßmanagement-Kit*). Hierunter fielen konzeptionelle Arbeiten, Arbeiten zur prototypischen Umsetzung, die Entwicklung eines Bausteinsatzes standardisierter Prozeßmodell(fragment)e sowie Schulungsangebote.

Vorgehensweise

Im Rahmen der konzeptionellen Arbeiten wurde der ISST-spezifische Ansatz zum Management von Geschäftsprozessen weiterentwickelt und verfeinert. Die Anforderungen für diese Weiterentwicklungen ergaben sich aus den jeweiligen Auftragsprojekten. Im einzelnen wurden Konzepte, Methoden und Techniken erarbeitet: zur Bestandsaufnahme (Erhebung) von Geschäftsprozessen in Unternehmen (Ist-Prozeßsituation), zur verständlichen Beschreibung und Dokumentation dieser Prozesse, zur Analyse und Entwicklung eines Soll-Konzeptes für die optimale Gestaltung von Geschäftsprozessen, zur computergestützten Ausführung dieser Prozesse, zum verteilten Management von Prozessen sowie zur Erweiterung von Prozeßmodellanalysen in Hinblick auf betriebswirtschaftliche Fragestellungen.

Ergebnisse

Die Konzepte und Methoden wurden in der Regel prototypisch in der eigens entwickelten Prozeßmanagement-Umgebung CORMAN (*Coordination Manager*) umgesetzt. Dadurch konnte zum einen die Tragfähigkeit der entwickelten Konzepte demonstriert werden, zum anderen war eine Anwendung in praktischen Modellierungsprojekten möglich.

Darüber hinaus wurde mit der Entwicklung standardisierter Modellbausteine begonnen (Beispiel: V-Modell als standardisierte Vorgehensweise zur Abwicklung von Datenverarbeitungsprojekten in öffentlichen Verwaltungen). Für solche Bausteine sollen Anpassungs- und Einführungsstrategien erarbeitet werden, die „maßgeschneiderte“ Modelle ermöglichen und so allgemeingültige Richtlinien um firmenspezifische Anforderungen ergänzen. Auf diese Weise können zukünftig bei der Prozeßmodellierung „Schablonen“ angewandt werden, die nachweisbar über bestimmte Qualitätseigenschaften verfügen. Dies

reduziert den Erstellungsaufwand individueller Modelle und gewährleistet, daß Prozesse bewährten und allgemein angewendeten Richtlinien folgen.

2.4.2 Management kundenspezifischer Geschäftsprozesse

Aufgabenstellung und Ziele

In diesem Arbeitsgebiet wurde der Ansatz zum systematischen Management von Geschäftsprozessen im Rahmen von Projekten konkret angewandt. Mit den im ISST entwickelten Methoden und Techniken sollte es Auftraggebern aus Industrie und öffentlicher Verwaltung möglich sein, ihre Geschäftsprozesse besser zu organisieren und dv-technisch zu unterstützen.

Vorgehensweise

Projekte zum Management bzw. zur Verbesserung der DV-Unterstützung von Geschäftsprozessen stellen in der Regel zeit- und kostenintensive Vorhaben dar. Unternehmen und Organisationen wollen solche Vorhaben daher häufig schrittweise durchführen. Die erforderlichen Aufwendungen lassen sich zudem oft nur schwer und ungenau abschätzen. Um Prozeßmanagement-Projekte überschaubarer zu gestalten, erstreckt sich das Dienstleistungsangebot des ISST im Bereich Prozeßmanagement auf verschiedene Phasen:

- Consulting

Ziel dieser Phase ist es, Inhalte eines systematischen Managements von Geschäftsprozessen zu vermitteln. Interessierte Unternehmen sollen die Notwendigkeit bzw. die Möglichkeit der (Re-)Organisation ihrer Geschäftsprozesse erkennen und einschätzen lernen. Auf dieser Basis können dann die konkreten Inhalte und Ziele eines Folgeprojektes definiert werden.

- Prozeß-Engineering

In dieser Projektphase werden die Geschäftsprozesse einer Unternehmung erfaßt und formal beschrieben (Ist-Zustand). Anhand der Modelle können die Prozesse analysiert und Verbesserungen vorgeschlagen werden.

- Technisches Engineering

Auf der Basis der Modellierung können für das luK-System, das die Geschäftsprozesse DV-technisch unterstützen soll, Anforderungsanalyse und Systementwurf vorgenommen werden.

- **Betreuung**

Zur Umsetzung von Prozeßveränderungen bzw. zur Einführung von IuK-Systemen bedarf es geeigneter Einführungsstrategien. Diese berücksichtigen z.B. mitbestimmungswirksame Fragestellungen, Schulungen, die Planung von Einführungszeitpunkten u.a. Ferner sind die vorgenommenen Veränderungen kritisch zu evaluieren, um dauerhaft Prozeßverbesserungen zu erreichen.

Ergebnisse

Nach obigem Schema wurden im Jahr 1993 drei Projekte bearbeitet, von denen zwei im Berichtsjahr abgeschlossen werden konnten (FISCUS und KSG). Die Projekte im einzelnen:

Vorgangsmodellierung für das Steuerwesen:

Das Projekt FISCUS ist ein Gemeinschaftsprojekt des Bundes- und der Länderfinanzministerien zur Neukonzeption der EDV-technischen Abwicklung des Steuerwesens. Bisher existierten in den Landesrechenzentren Programmpakete zur Steuerabwicklung, die die einzelnen Bundesländer zum Teil eigenständig entwickelten. Da die bisher verwendeten Systeme veraltet waren, sollten sie durch eine neue Lösung ersetzt werden. Das neue System sollte dabei gemeinsam konzipiert werden.

Das ISST führte dazu im Berichtsjahr eine Prozeßmodellierung durch. Damit wurden die Organisations- und Ablaufstrukturen definiert, die die Zusammenarbeit der verschiedenen Gremien (Koordinierungsausschuß, Assistenzgruppe, Referentengruppe, ausführende Gremien, systemtechnische Zentrale, Migrationsgruppe etc.) innerhalb der Behörden festlegen. Neben den Kommunikationsbeziehungen der Gremien wurden auch einzelne Vorgehensweisen detaillierter modelliert. Vorgaben in Form expliziter Richtlinien garantierten die Qualitätssicherung der Arbeiten innerhalb des Projektes.

Vorgangsmodellierung für die Kraftwerks-Simulator-Gesellschaft:

Die Kraftwerks-Simulator-Gesellschaft mbH (KSG) betreibt Simulatoren für alle Kernkraftwerkstypen, die in der Bundesrepublik Deutschland am Netz sind. Ein Simulator verhält sich in allen relevanten Details wie sein reales Pendant und dient der Schulung des Kraftwerkspersonals. Für diesen Hochsicherheitsbereich ist es unverzichtbar, daß Veränderungen an den Kraftwerken umgehend an den Simulatoren, die teils durch Bauteile, teils durch Software realisiert sind, nachvollzogen werden. Ebenso unverzüglich müssen Fehler in den Simulatoren behoben werden.

In der Regel werden die erforderlichen Simulatoränderungen der KSG über Nachrüstungs- und Änderungsaufträge (*Discrepancy Reports*) mitgeteilt. Auch Schulungen des Kraftwerkspersonals durch die KSG fördern Diskrepanzen zwischen Modell und Wirklichkeit zutage.

Für die Bearbeitung der *Discrepancy Reports* sind in der KSG die Abteilungen T-SW und T-HW verantwortlich. T-SW führt Software-Änderungen an den Simulatorprogrammen durch. T-HW nimmt Hardware-Änderungen an den Leitwarten der Simulatoren vor. Die Abläufe zur Bearbeitung von *Discrepancy Reports* sind je nach Kraftwerkstyp unterschiedlich.

Das ISST hatte die Aufgabe, die Abläufe innerhalb der Abteilung T-SW zu modellieren. Neben der Dokumentation diente das erstellte Vorgangsmodell auch zur Analyse der bisherigen Bearbeitung von Änderungsaufträgen.

Im ersten Teil des Projektes wurde die aktuelle Vorgehensweise im Rahmen eines Workshops mit Mitarbeitern des Bereichs T-SW erhoben. Diese Informationen bildeten die Grundlage für eine graphische Darstellung der Abläufe mittels der CORMAN-eigenen Modellierungssprache FUNSOFT-Netz.

Im zweiten Teil des Projektes wurde das FUNSOFT-Netz analysiert. Dazu wurde es um notwendige Informationen erweitert.

Die Analyse umfaßte Untersuchungen der Netztopologie sowie Simulationen des FUNSOFT-Netzes. Damit wurden die Prozesse auf Schwachstellen, wie z.B. Deadlocks, permanent aktivierbare Instanzen oder Beschränktheit von Kanälen, untersucht. Bei der anschließenden Simulation wurde die Bearbeitung der *Discrepancy Reports* durchgespielt. So ließen sich die einzelnen Vorgänge auf mögliche Auswirkungen untersuchen. Auch Aussagen u.a. über Zeitverhalten, Personalbedarf eines Vorganges waren möglich.

Die Ergebnisse der Untersuchung hat das ISST in einem Bericht zusammengefaßt und der KSG mit Verbesserungsvorschlägen präsentiert. Das FUNSOFT-Netz wurde in das KSG-Handbuch zur Pflege von Kraftwerkssimulatoren aufgenommen.

Vorgangsmodellierung für das NRW-Wirtschaftsministerium:

Im Mai 1993 startete das ISST ein Projekt, das die Entwicklung eines „Dateninformationssystems Wirtschaft (DIS)“ für das Ministerium für Wirtschaft, Mittelstand und Technologie des Landes Nordrhein-Westfalen zum Ziel hat. Mit diesem System sollen die unterschiedlichsten Daten, die das Ministerium benötigt, elektronisch gespeichert, zusammengeführt und bearbeitet werden. Dabei handelt es sich beispielsweise um Informationen über Wirtschaftsförderdaten, laufende Projekte, regionale Entwicklungskonzepte, Qualifizierungsmaßnahmen oder Außenwirtschaftsaktivitäten. Die Daten sind für bestimmte Zwecke - beispielsweise für Analysen, Bewertungen und Darstellungen - und zielgruppenspezifisch - etwa als schriftliche Informationen, Graphiken oder Redeunterlagen - aufzubereiten. Diese Informationen sind wiederum als Dokumente zu speichern. Darüber hinaus mußte sich das zu entwickelnde System in die bestehende DV-Infrastruktur einfügen. Eine Vielzahl von Abteilungen und Referaten mit unterschiedlichen Arbeitsabläufen und -bezügen war zu integrieren: Alle Beteiligten sollten Informationen eingeben, abrufen und gruppenübergreifend austauschen können; Arbeitsabläufe sollten durch das System möglichst transparent gestaltet und beschleunigt werden.

Die Erstellung eines solchen Dateninformationssystems ist eine komplexe Aufgabe. Daher führte das ISST zunächst mit seinem Prozeßmanagementsystem CORMAN (auch: *Workflow Management System*) eine Anforderungsanalyse durch. Um eine zielgerichtete Verwendung des Informationssystems sicherzustellen, wurden die Arbeitsabläufe in den einzelnen Gruppen, Referaten und Abteilungen des Ministeriums erfaßt. Auch die Daten und Funktionen des Systems wurden konkretisiert.

In der zweiten Projektphase wird das System entworfen und die konkrete Umsetzung im Ministerium koordiniert. Aktivitäten dieser Projektphase sind: Daten- und Funktionsmodellierung zur Präzisierung aller Informationen, die im Rahmen der Bearbeitung und Auswertung von Förderprojekten benötigt werden, technische und organisatorische Maßnahmen zur effizienteren Gestaltung von Verwaltungsvorgängen, insbesondere zur kooperativen Bearbeitung von Dokumenten.

Das Projekt wird Frühjahr 1994 abgeschlossen sein.

Veröffentlichungen

[1], [3], [4], [11], [33], [34], [35], [36], [52], [56]

2.5 Aus- und Weiterbildung

Das ISST ist sich bewußt, daß die Einführung moderner Technologien in den dargestellten Bereichen der Software- und Systemtechnik das Fachpersonal vor neue Herausforderungen stellt. Der derzeitige Ausbildungs- und Wissensstand der Software-Spezialisten wird dem zukünftigen Anforderungsprofil häufig nicht gerecht. Daher sieht es das ISST als seine Aufgabe, Softwareentwicklern, Systemintegratoren und EDV-Managern mit einem breitgefächerten Aus- und Weiterbildungsprogramm die Möglichkeit zu bieten, sich das erforderliche Know-how praxisgerecht anzueignen.

Der Bereich AuW des ISST hat zum Ziel, Aus- und Weiterbildungskonzepte zu entwickeln und diese in Seminaren und Workshops umzusetzen. Im Rahmen des deutschen Kursangebotes zum *European Professional Software Engineering Program* (EPSE) wurde 1993 eine Reihe von Seminaren durchgeführt. Mit diesen und anderen Weiterbildungsveranstaltungen, die speziell auf die Bedürfnisse und Anforderungen von Software-Praktikern zugeschnitten sind, konnte das ISST einen wesentlichen Beitrag zum Wissens- und Technologietransfer in die Wirtschaft leisten.

European Professional Software Engineering Program

Das EPSE-Programm richtet sich an den praktisch tätigen Softwareentwickler, der sich in den Bereichen Software Engineering und werkzeuggestützte Softwareentwicklung weiterqualifizieren möchte. Das Kursprogramm besteht aus einer Reihe von Einzelkursen, die der Kursteilnehmer individuell, d.h. seinen aktuellen beruflichen Bedürfnissen entsprechend, zusammenstellen kann. Ziel der Teilnahme ist es, das anerkannte Zertifikat zum *European Professional Software Engineer* zu erwerben.

Das EPSE-Rahmenprogramm wurde von einem europäischen Konsortium großer Software-Häuser und führender Forschungsinstitutionen aus dem Bereich des Software Engineering entwickelt. Das deutsche Programm wurde unter der Regie des ISST im Auftrag des Informatik Centrum Dortmund erarbeitet.

Als Dozenten konnte das ISST renommierte Spezialisten aus Wirtschaft und Wissenschaft gewinnen. Einige Kurse wurden auch von den Experten des ISST abgehalten:

- Systematisches Management von Vorgängen
- Grundlagen der Informationsmodellierung
- Integrierte Software-Infrastrukturen
- Industrialisierung von Software Engineering und Computer Aided Software Engineering
- Software-Prozeß-Management mit FUNSOFT-Netzen
- Anforderungsanalyse und Bewertung von Software-Produktionsumgebungen
- Software Re-Engineering

Um das EPSE-Kursangebot des ISST noch stärker auf die Bedürfnisse des Adressatenkreises ausrichten zu können, wurde zudem eine Marktforschungs-

studie durchgeführt. Damit konnten die Weiterbildungswünsche von Softwareentwicklern identifiziert und zur Optimierung des Kursprogrammes herangezogen werden.

Informationen zum aktuellen Kursangebot können beim ISST angefordert werden.

Durchführung von Workshops zu aktuellen Themen

Das ISST veranstaltete im Berichtsjahr Workshops zu aktuellen Themen aus den Bereichen Software Engineering und Prozeß-Management. Diese meist eintägigen Workshops behandelten ein Thema in kompakter Form und vermittelten den Teilnehmern einen Überblick über den aktuellen Stand der Diskussion (u.a. Workshop „Workflow-Management für Geschäfts- und Verwaltungsprozesse“).

Die ISST-Workshop-Reihe soll auch in 1994 fortgeführt werden.

Aus- und Weiterbildung im universitären Rahmen

Der Bereich Aus- und Weiterbildung hat für den Fachbereich Informatik der Universität Dortmund eine Vorlesung zum Thema „Software-Prozeßmanagement mit FUNSOFT-Netzen“ ausgearbeitet und im Sommersemester 1993 gehalten.

Technologie-Transfer

Der Bereich AuW beschäftigte sich ferner mit Fragen des Technologie-Transfers von der angewandten Forschung des ISST zum industriellen Einsatz. In diesem Zusammenhang wurde ein Projekt in einem mittelständischen Softwarehaus bearbeitet. Ziel war die Erstellung einer Software-Entwicklungsumgebung auf der Basis der im ISST entwickelten Prozeßmanagement-Technologie. Das Projekt konnte zum Jahresende abgeschlossen werden.

3

Öffentlichkeitsarbeit

3.1 ISST-Eröffnungsveranstaltung

Am 17. Mai 1993 präsentierte sich das ISST im Rahmen einer feierlichen Eröffnungsveranstaltung einer Öffentlichkeit aus Wirtschaft, Wissenschaft, Politik und Medien. 150 geladene Gäste aus dem In- und Ausland informierten sich über Forschungsaufgaben und Kompetenzbereiche des jungen Fraunhofer-Instituts.

Repräsentanten des Berliner Senats und führende Wirtschafts- und Wissenschaftsvertreter würdigten in ihren Grußworten die Bedeutung des Instituts für die Weiterentwicklung der Softwaretechnologie. Fachbeiträge international renommierter Softwareexperten und Leistungsschauen aktueller ISST-Arbeiten machten die Veranstaltung zu einem anregenden Diskussionsforum zum Thema „Informationstechnik für das Jahr 2000“.



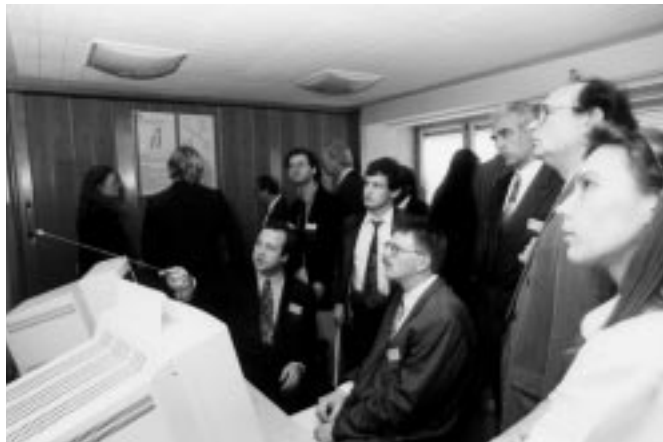
Prof. Dr. Herbert Weber, Leiter des ISST, begrüßt die Gäste der Eröffnungsfeier. Programmatischer Titel seines Festvortrages: „Informationstechnik mit Software-Infrastrukturen - entwickelt in Software-Fabriken“

Gäste der Eröffnungsveranstaltung (von links): Graham Samuel, ESF Projektdirektor, Prof. Leon Osterweil, University of California at Irvine, Hubert Tardieu, SEMA Group, Prof. Dieter Schumann, Präsident der Technischen Universität Berlin, Prof. Herbert Weber und ISST-Kurator Heinz Paul Bonn

Demos aktueller Entwicklungen:
Das multimediale Informationssystem TIC



Den Fragen der Journalisten stellen sich (v. l.): Fraunhofer-Altpäsident Prof. Dr. Max Syrbe, Prof. Weber, IBM-Geschäftsführer Bernhard Dorn und Staatssekretär Dr. Hans Kremendahl; Moderator Franz Miller



Fachgespräche: Wissenschaftler des ISST präsentieren ihre Arbeiten

3.2 Veröffentlichungen

3.2.1 Beiträge in (Fach-) Zeitschriften und Büchern

- [1] R. Adomeit,
W. Deiters,
B. Holtkamp,
B. Rockwell **Work Coordination Support in Kernel/2r**
In: Encyclopedia of Software Engineering, John Wiley & Sons, New York, 1993
- [2] L. Budach **Homogeneity versus Finiteness**
In: The Earth and the Universe, Interdivisional Commission on History of the International Association of Geomagnetism and Aeronomy (IAGA), 112-121, 1993
- [3] W. Deiters,
V. Gruhn,
H. Weber **Software Process Evolution in MELMAC**
In: Cooke, D. (ed.): The Impact of CASE on the Software Development Life Cycle, 1993
- [4] W. Deiters,
V. Gruhn **The FUNSOFT Net Approach to Software Process Management**
Beitrag 1993 eingereicht, erscheint 1994 in: International Journal on Software Engineering and Knowledge Engineering, Vol. 4, No. 2, Juni 1994
- [5] J. Gottschick,
L. Nentwig,
M. Timmermann **ICOMA - Ein Werkzeug für ein integriertes Konfigurationsmanagement**
In: Löw, H.-P.; Partosch, G. (Hrsg.): Verteilte Systeme: Organisation und Betrieb, Dt. Universitäts-Verlag, Wiesbaden, 1993
- [6] M. Heiner,
G. Ventre,
D. Wikarski **A Petri Net Based Toolkit to Integrate Quantitative and Qualitative Analysis**
In: Proceedings of the International Workshop on Software Engineering for Parallel Systems, World Transputer Congress '93, Aachen, September 1993
- [7] A. Kind,
H. Friedrich **A Practical Approach to Type Inference for EuLisp**
In: Lisp and Symbolic Computation: An International Journal, 6, 159-176, 1993
- [8] M. Löwe **Algebraic Approach to single-pushout graph transformation**
In: Theoretical Computer Science, 109, 181-224, 1993
- [9] M. Löwe,
M. Kerff,
W. Wagner **An algebraic framework for the transformation of attributed graphs**
In: Term Graph Rewriting: Theory and Practice, 185-199, Wiley, 1993
- [10] D. Wikarski **Locally Markovian Object Nets - The Conceptual Model**
In: Proceedings of the International Workshop on Concurrency, Specification and Programming, Informatik-Preprint 22, FB Informatik der Humboldt-Universität zu Berlin, 182-189, 1993

3.2.2 Vorträge auf Konferenzen und Tagungen

- [11] W. Deiters,
V. Gruhn **Software Process Technology Transfer - A Case Study**
In: Proceedings of the 8th International Software Process Workshop,
Dagstuhl, 1993
- [12] S. Dewal **CASE Deployment: Improving Software Development by
Transfer of Technology**
In: Proceedings of the International Conference on Software Engineering
Environments, Reading, Großbritannien, Juli 1993
- [13] S. Dewal,
B. Prather **CASE Implementation Life Cycle and Critical Success Factors**
In: Proceedings of the International Workshop on CASE, Singapur, Juli 1993
- [14] G. Dinkhoff, X. Wu **Graphical Interface for a Fully Object-Oriented Database System**
In: Proceedings of ACM SAC '93, 318-325, Indianapolis, Februar 1993
- [15] P. Gabriel **Practical Experiences from using the specification language II**
COMPASS/COMPUGRAPH subgroup meeting on Specification of Concurrent
and Object Oriented Systems, L'Aquila, Italien, Juni 1993
- [16] R. Gastner **FORWARD: A workbench for automating domain-specific
software design**
In: Proceedings of the Sixth Florida Artificial Intelligence Research Symposium,
166-170, 1993
- [17] J. Gottschick **ICOMA - Ein Werkzeug für ein integriertes Konfigurationsmanagement**
GI-Fachgespräch über Rechenzentren: Verteilte Systeme: Organisation und
Betrieb, Gießen, 16./17. 9. 1993
- [18] J. Gottschick **ICOMA - Integriertes Konfigurationsmanagement**
DFN-Betriebstagung, Berlin, 17. 3. 1993
- [19] B. Holtkamp,
H. Weber **Kernel/2r - A Software Infrastructure for Building Distributed Applications**
In: Proceedings of FTDCS, Lissabon, 22.9.-24.9.1993
- [20] A. Kind,
E.U. Kriegel,
H. Friedrich,
I. Mohr **Compiling EuLisp to C - New Approaches to Type Inference and
Memory Management**
Beitrag 1993 eingereicht zur Konferenz über Lisp und funktionale
Programmierung, Sommer 1994/Orlando/Florida
- [21] A. Kind **Typinferenz: Voraussetzung zur Erstellung effizienter Lisp-Applikationen**
In: Tagungsband zum BMFT-Statusseminar „Künstliche Intelligenz“, Berlin,
27./28.4.1993

- [22] E.U. Kriegel **A conservative garbage Collector for an EuLisp to ASM/C Compiler**
In: Collected Preprints of Position Papers OOPSL '93, Workshop on Memory Management and Garbage Collection, Washington/DC, 27.9.1993
- [23] M. Löwe, M. Beyer **AGG: An implementation of algebraic graph rewriting**
In: Proceedings of Rewrite Techniques and Application RTA '93, 451-456, 1993
- [24] M. Löwe **Algebraische Graphgrammatiken - Theorie und Anwendungen**
DFKI-Kolloquium Kaiserslautern, Januar 1993
- [25] M. Löwe **Initial Semantics for Specification Developments**
Dagstuhl-Seminar zum Thema "Spezifikation und Semantik", Mai 1993
- [26] J. Neuhaus, X. Wu **A Smooth Migration Path Towards a Fully Object-Oriented PCTE**
PCTE '93, Paris, November 1993
- [27] J. Neuhaus, X. Wu **Data integration for a software development environment:
Using an object-oriented OMS**
In: Welland, R. (ed.): Proceedings of the Software Engineering Environments Conference, 120-129, Reading, Großbritannien, 7.6.-9.6.1993
- [28] J. Neuhaus, X. Wu **Extending PCTE with object-oriented capabilities**
In: Mavrik, V., Lavzansky, J. und Wagner, R.R. (eds.): Proceedings of the 4th International Conference on Database and Expert Systems Applications, number 720 in Lecture Notes in Computer Science, 681-684, Prag, 6.9.-8.9.1993
- [29] H. Weber **Industrial Application of Software Engineering as
the Victim of Programming**
GI-Fachgespräch, GI-Jahrestagung Dresden, 27.9.-1.10.1993
- [30] H. Weber **Software Factory Engineering**
8th German-Japanese Forum on Information Technology, Weimar, 3.5.-6.5.1993
- [31] D. Wikarski **Further Remarks on the Relationship between C-E Structures and
Petri Nets**
Internationaler Workshop "Concurrency Specification and Programming", Nieborow, Polen, Oktober 1993

3.2.3 Sonstige Vorträge

- [32] L. Budach **Objektbasierte Spezifikationstechniken in II**
Eröffnungsveranstaltung des ISST, Berlin, 17.5.1993
- [33] W. Deiters,
R. Fehling,
V. Gruhn,
W. Schäfer **Schlüsselkomponenten der Forschungsinstanz der EUREKA Software
Factory**
ONLINE '93, Hamburg, Februar 1993
- [34] W. Deiters,
V. Gruhn **Software Process Management with FUNSOFT Nets**
European Space Agency (ESA), Katwijk, Niederlande, 18.5.1993
- [35] W. Deiters **Systematisches Management von Geschäftsprozessen**
Eröffnungsveranstaltung ISST, Berlin, 17.5.1993
- [36] W. Deiters **Systematisches Management von Geschäftsprozessen mit CORMAN**
Präsentation für das ESPRIT Projekt HICOS, München, Januar 1993
- [37] S. Dewal **Wesentliche Kriterien der objektorientierten Software-Entwicklung im
Vergleich zu konventionellen Methoden**
Anwenderverband der Informationsverarbeiter (ADI), Neuss, September 1993
- [38] P. Gabriel **Praktische Erfahrungen mit der Spezifikationssprache II**
Forschungskolloquium der Einheit TFS (Theorie von formalen Sprachen und
Spezifikationen) des Fachbereichs Informatik der Technische Universität Berlin,
Mai 1993
- [39] J. Gottschick,
L. Nentwig **Service- und Administrationszentrum im
Landesamt für Informationstechnik**
Vorstellung der Ergebnisse der Minze-Studie im Landesamt für
Informationstechnik (LIT), Berlin, 24.2.1993
- [40] J. Gottschick **Organisation von Administrationszentren innerhalb einer IT-Infrastruktur
im Land Berlin**
Vorstellung der Ergebnisse der Wirtschaftlichkeitsbetrachtung im Landesamt
für Informationstechnik (LIT), Berlin, 3.9.1993
- [41] M. Löwe **Formale Methoden in der Software-Entwicklung am Beispiel eines
medizinischen Informationssystems**
Vorstellung des Projekts "Korrekte Software KORSO" für Industrie und
Universitäten, März 1993
- [42] H. Schween **S-Transaktionen**
Vortrag bei PSI, Berlin, 14.7.1993
- [43] M. Timmermann **ICOMA - Integriertes Konfigurationsmanagement**
Vorstellung des ICOMA-Projektes, ISST-Kolloquium, Berlin, 29.6.1993

- [44] H. Weber **Das Projekt Eureka Software Factory: Status, Ausblick und Marktwirkung**
IBM Zürich, 9./10.11.1993
- [45] H. Weber **Informationstechnik für das Jahr 2000**
Versicherungsausschuß, IHK Köln, 3.2.1993
- [46] H. Weber **IT-Perspektiven bis zum Jahre 2000**
Kölner-Software-Tage, Köln, 11.6.1993
- [47] H. Weber **IT-Perspektiven bis zum Jahre 2000**
GUS-Kundentreff, Königswinter, 14.9.-16.9.1993
- [48] H. Weber **Qualitätsnormen und Qualitätssicherungsverfahren bei Software**
Gemeinschaftsveranstaltung „Qualitätsnormen bei Software“ der
Wirtschaftsförderung Berlin GmbH, ISST Berlin, 3.11.1993
- [49] B. Windolph,
R. Witschurke **Erfahrungen mit den RE-SPEC Werkzeugen von EPOS**
EPOS Benutzertagung, Oktober 1993

3.2.4 Technische Berichte und Studien

- [50] H. Adametz **Konzept eines Kontextcheckers für die P-Umgebung**
Interner Bericht, Berlin, März 1993
- [51] R. Adomeit,
W. Deiters,
B. Holtkamp **Work Coordination in Kernel/2r**
ISST-Bericht 14, Juli 1993
- [52] R. Adomeit,
W. Deiters,
B. Holtkamp **Work Coordination Support in Kernel/2r**
ISST Technical Report No. 14, 1993
- [53] B. Barthel **Tutorial für die Einberufung von Videokonferenzen**
Interner Bericht, Berlin, Juni 1993
- [54] L. Budach **Programming in II -Primer and Method Guide**
Interner Bericht, Berlin, Juni 1993
- [55] L. Budach **Software Design in II - ein Vorschlag**
Interner Bericht, Berlin, Juli 1993
- [56] W. Deiters **A View Based Approach to Software Process Management**
Technical Report No. 467, Universität Dortmund, Fachbereich Informatik, 1993

- [57] G. Faustmann **Technikunterstützte Informationsverarbeitung - Einführung in die Problematik**
Interner Bericht, Berlin, Februar 1993
- [58] G. Faustmann **Einführung von Informationstechnik in Verwaltungsbetriebe - Methodiken und Werkzeuge**
Interner Bericht, Berlin, Juni 1993
- [59] A. Fellien,
J. Gottschick,
L. Nentwig,
M. Timmermann **Wirtschaftlichkeitsbetrachtung über die Organisation von Administrationszentren innerhalb einer IT-Infrastruktur im Land Berlin**
Berlin, Oktober 1993
- [60] U. Freitag,
M. Richter,
D. Wikarski **A Coarse State Transition Model for P-Components in PISA**
Interner Bericht, Berlin, April 1993
- [61] U. Freitag,
S. Lange **Die Realisierung der Komponente FILESYSTEM unter Unix**
Interner Bericht, Berlin, April 1993
- [62] U. Freitag,
P. Gabriel **Erfahrungsbericht zum Projekt MFZ**
Interner Bericht, Berlin, Oktober 1993
- [63] J. Gottschick,
L. Nentwig,
H. Noch,
P. Schlicht,
M. Timmermann,
G. Weiß **Minze - Studie für ein Administrationszentrum im Landesamt für Informationstechnik**
Studie zum Projekt SAZ des Landesamtes für Informationstechnik (LIT), Berlin, Februar 1993
- [64] B. Holtkamp,
N. Weissenberg **Kernel/2r - Eine Übersicht**
ISST-Bericht 16, Juli 1993
- [65] ISST Berlin, LIT **Pflichtenheft für ein Service- und Administrationszentrum im Landesamt für Informationstechnik Berlin**
Juni 1993
- [66] A. Kind,
H. Friedrich **A Practical Approach to Typeinference for EuLisp**
ISST-Bericht 8, April 1993
- [67] A. Kind,
H. Friedrich **A Practical Approach to Typeinference for EuLisp**
APPLY-Arbeitspapier APPLY/ISST/II.1/2, Berlin, Juni 1993
- [68] E.U. Kriegel **A conservative garbage Collector for an EuLisp to ASM/C Compiler**
Contribution to the Garbage Collection Workshop, Washington/DC, 27.9.1993;
APPLY-Arbeitspapier APPLY/ISST/VI.1/4, Berlin, 14.10.1993

- [69] E.U. Kriegel,
I. Mohr **Evaluierung des Static-Fileservers**
Technischer Bericht für Symbolics Systemhaus GmbH, Berlin 18.1.1993
- [70] M. Löwe,
J. Müller **Algebraische Graphersetzung: Mathematische Modellierung und Konfluenz**
Technische Universität Berlin, Bericht Nr. 93/37, 1993
- [71] J. Neuhaus, X. Wu **An overview of the InORM system**
ISST-Bericht 13, Juli 1993
- [72] J. Neuhaus, X. Wu **Data Integration for a Software Development Environment: Using an Object-Oriented OMS**
ISST-Bericht 6, Februar 1993
- [73] J. Neuhaus, X. Wu **Future directions for the PCTE/OMS: Achieving a fully object-oriented data model**
ISST-Bericht 11, Juni 1993
- [74] J. Neuhaus, X. Wu **Sprachbeschreibung EQL - Release 1.0**
ISST-Bericht 12, Juli 1993
- [75] J. Neuhaus,
H. Schween,
G. Wehrsdorfer,
X. Wu **Towards OOPCTE**
ISST-Bericht 7, März 1993
- [76] D. Wikarski **Netzdarstellung von Ausführungsfolgen von Operationen**
Interner Bericht, Berlin, August 1993

3.3 Wissenschaftliche Arbeiten

3.3.1 Studien- und Diplomarbeiten

- [77] S. Alker **Ein hybrider, grafischer Syntaxeditor für Konfigurationsspezifikationen in der Π -Sprache**
Diplomarbeit an der Universität Dortmund, 1993
- [78] F. Hendle **Entwicklung einer formalen Beschreibungsmethode für die Prozeß- und Datenvisualisierungskomponenten in Benutzerschnittstellen logischer Systeme**
Diplomarbeit an der Universität Dortmund, 1993
- [79] S. Kohaupt **Integration von Π und HIT**
Diplomarbeit an der Universität Dortmund, 1993
- [80] S. Lange **Konzept für die Einbindung von PISA in Kernel/2r**
Jahresarbeit an der Humboldt-Universität zu Berlin, 1993
- [81] A. Liman **Verteilte Modellierung und Ausführung von Software-Prozeßmodellen**
Diplomarbeit an der Universität Dortmund, 1993
- [82] J. Minneker,
A. Völlinger **Entwicklung und Realisierung eines Werkzeugs zur Ausführung von Π -Spezifikationen**
Diplomarbeit an der Universität Dortmund, 1993
- [83] D. Noss **DDT: Entwicklung eines Datenbank Design Tools zur Spezifikation wertabhängiger Integritätsbedingungen**
Diplomarbeit an der Universität Dortmund, 1993
- [84] J. Seekamp **Ein Prozeßmodell für die Entwicklung von Π -Spezifikationen**
Diplomarbeit an der Universität Dortmund, 1993
- [85] M. Stemmer **Kompatibilität von Π -Spezifikationen**
Diplomarbeit an der Universität Dortmund, 1993
- [86] S. Stieghahn **RAISE und PISA - ein Vergleich**
Diplomarbeit an der Technische Universität Dresden, 1993/94
- [87] M. Wilhelm **Untersuchung der Eignung objektorientierten Designs für die Entwicklung von Produktionsplanungs- und -steuerungssystemen für Kaltwalzwerke**
Diplomarbeit an der Universität Dortmund, 1993

3.3.2 Dissertationen

- [88] R. Gastner **Automatisches Programmieren von Werkzeugen für Wissensbasen**
Dissertation an der Universität Erlangen-Nürnberg, April 1993
- [89] T. Luckenbach **Netzwerkopplung (Interworking) über Breitband-ISDN - Entwurf und Realisierung einer modularen Systemarchitektur**
Dissertation an der Technischen Universität Berlin, September 1993
- [90] B. Messer **Fachsprachliche Entwicklung einer Theorie der Programme**
Dissertation an der Technischen Universität Berlin, Fachbereich Informatik, April 1993
- [91] V. Tschammer **Architektur und Dienstplattform für offene verteilte Anwendungen**
Dissertation an der Technischen Universität Berlin, September 1993

3.4 Veranstaltungen des ISST

3.4.1 Seminare und Workshops

W. Deiters,
H. Weber

**Workflow Management für Geschäfts-
und Verwaltungsprozesse**

Workshop, Technologiezentrum Dortmund, 28.4.1993

W. Deiters,
H. Weber

**Workflow Management für Geschäfts-
und Verwaltungsprozesse,**

Workshop, ISST Berlin, 26.5.1993

S. Dewal

Einführung werkzeuggestützter Software-Entwicklung

Seminar, ISST Dortmund, Dezember 1993

3.4.2 Kolloquia

G. Faustmann

Einführung von Informationstechnik in Verwaltungsbetriebe

ISST Berlin, Juni 1993

A. Kind

**Typinferenz - Zur Effizienz und Flexibilität dynamisch getypter
Programmiersprachen**

ISST Berlin, September 1993

B. Messer

**Mecca - Multimedia Communication Coordination - Ein Vorschlag für ein
Forschungsprojekt**

ISST Berlin, Januar 1993

B. Messer

Typen von Hochleistungsnetzen - B-ISDN, DQDB und FDDI

ISST Berlin, Februar 1993

L. Nentwig

**Konzepte für ein Administrationszentrum im Berliner Landesamt für
Informationstechnik**

Vorstellung der Minze-Studie, ISST Berlin, 23.3.1993

3.4.3 Präsentationen

Apply-Gruppe

EuLisp -> ASM-Compiler

Präsentation anlässlich der Tagung des Apply-Beirats, Kiel, 7.6.1993

Apply-Gruppe

EuLisp -> C-Compiler

Präsentation anlässlich der Tagung des Apply-Beirats, Berlin, 20.12.1993

- W. Deiters
V. Gruhn **Software Process Management with FUNSOFT Nets**
Präsentation CORMAN für die European Space Agency (ESA), Katwijk,
Niederlande, 18.5.1993
- G. Faustmann **Bürgerterminal - Unterstützung des Bürgers bei der Antragstellung**
ISST-Eröffnungsveranstaltung, Berlin, 17.5.1993
- J. Gottschick
M. Timmermann **ICOMA**
ISST-Eröffnungsveranstaltung, Berlin, 17.5.1993
- L. Nentwig **Der HotCon-Prototyp**
ISST-Eröffnungsveranstaltung, Berlin, 17.5.1993
- L. Nentwig **HotCon**
ISST-Eröffnungsveranstaltung, Berlin, 17.5.1993
- H. Weber **Das Weiterbildungsangebot des ISST**
Präsentation für Fa. Novell/England, ISST Dortmund, 12.7.1993
- H. Weber **Das Dienstleistungsangebot des ISST**
Präsentation für das Verteidigungs- und Verkehrsministerium, ISST Dortmund,
14.7.1993
- H. Weber **Das Kompetenz- und Dienstleistungsangebot des ISST**
Präsentation für Fa. SEL, ISST Dortmund, 19.7.1993
- H. Weber **Software Factory Engineering**
Präsentation für ICL/England, ISST Dortmund, 31.8.1993

3.4.4 Sonstige Veranstaltungen

Eröffnungsveranstaltung des ISST

ISST Berlin, 17.5.1993

Qualitätsnormen bei Software

Informationsveranstaltung in Kooperation mit der Wirtschaftsförderung Berlin GmbH am 3.11.1993 im ISST Berlin (gemeinsam mit PSI und Condat DVBeratung Organisation Software GmbH)

3.5 Mitarbeit in Gremien, Ausschüssen, Konferenz- und Programmkomitees

- APPLY-Gruppe **Beteiligung an der Überarbeitung des internationalen Standardisierungsvorschlags für EuLisp in der ISO-Arbeitsgruppe 16**
- S. Dewal **Programmkomitee-Mitglied bei IWCASE 93 in Singapur**
Mitveranstalter des Workshops „CASE Implementation Life Cycle and Critical Success Factors“
- H. Friedrich **Teilnahme am Arbeitstreffen der EuLisp-Standardisierungsgruppe**
Brüssel, 19./20.11.1993
- B. Holtkamp **Programmkomitee-Mitglied des GI-Fachgesprächs „Softwaretechnik ‘93“**
Mitglied im Control Board des EUREKA ECMA PCTE-Projekts
- C. Peitscher **Beteiligung an der Organisation der Tools-Demo bei der GI-Fachtagung Softwaretechnik 93**
Dortmund, November 1993
- H. Weber **Landesbeauftragter für Informationstechnik des Landes Nordrhein-Westfalen, Mitveranstalter Petersberg-Seminare/Bonn**
Control Board Speaker, EUREKA Software Factory
Programmkomitee, 9th International Conference on Data Engineering
Wien, April 1993
General Chairman, 2nd International Conference on the Software Process
Berlin, 25./26.2.1993
Co-Chairman, Fourth Workshop on Future Trends of Distributed Computing Systems
Lissabon, 22.9.-24.9.1993
Keynote-Speaker, 2nd International Workshop on Software Reusability
Lucca, Italien, 24.5.-26.5.1993
Teilnahme am ISDN-Workshop
Königswinter, 6./7.5.1993

X. Wu

Object Management Group (OMG)

PCTE Special Interest Group

**Task Group Object-Oriented des ECMA Technical Committee
(TC33/TGOO)**

3.6 Vorlesungstätigkeit an Hochschulen

- H. Adametz **Entwicklung modularer Software-Systeme zur Informationsverwaltung**
Projektbetreuung, Technische Universität Berlin, Sommersemester 1993
- Entwicklung modularer Software-Systeme zur Informationsverwaltung 2**
Projektbetreuung, Technische Universität Berlin, Wintersemester 1993/94
- L. Budach **Theoretische Informatik**
Vorlesung , Universität Potsdam, Wintersemester 1993/94
- Informatik II**
Vorlesung, Universität Potsdam, Sommersemester 1993
- Informatik III**
Vorlesung, Universität Potsdam, Wintersemester 1993/94
- V. Gruhn **Software-Prozeßmanagement mit FUNSOFT-Netzen**
Vorlesung, Universität Dortmund, Sommersemester 1993
- M. Löwe **Einführung in die Theoretische Informatik**
Vorlesung, Technische Universität Berlin, Wintersemester 1992/93
- Konstruktive Spezifikation: Initiale Semantik**
Vorlesung, Ludwig-Maximilians-Universität München, Wintersemester 1992/93
- Repository**
Seminar, Technische Universität Berlin, Wintersemester 1993/94
- H. Weber **Software-Design**
Vorlesung, Technische Universität Berlin, Wintersemester 1992/93
- Entwicklung modularer Software-Systeme zur Informationsverwaltung**
Projektbetreuung, Technische Universität Berlin, Sommersemester 1993
- Grundlagen der Informationsmodellierung**
Vorlesung, Technische Universität Berlin, Wintersemester 1993/94

3.7 Teilnahme an Messen und Kongressen

CeBIT '93

Fraunhofer-Gemeinschaftsstand, Hannover, März 1993

GI-Fachgespräch „Softwaretechnik '93“

Dortmund, November 1993

ONLINE '93

Hamburg, Februar 1993

Softwaretechnik-Tage 93

Dortmund, November 1993

SYSTEMS '93

Softech NRW-Stand, München, Oktober 1993

2nd International Conference on the Software Process

Berlin, Februar 1993

8th International Software Process Workshop

Dagstuhl, März 1993

3.8 Internationale Beziehungen

3.8.1 Internationale Industrie- und Wissenschaftskontakte

Jim Babcock

Georgia Institute of Technology, Atlanta, USA

Bryan Fugate

Georgia Institute of Technology, Atlanta, USA

Prof. Dr. Chris Marlin

The Flinders University of South Australia, Australia

Prof. Dr. Leon J. Osterweil

University of California, Irvine, USA

Julian Padget

University of Bath, England

Hubert Tardieu

SEMA Group, Paris, Frankreich

3.8.2 Internationale Besucher im ISST

Jonathan Dart

Abt. Wissenschaft und Technologie, British Embassy, Bonn

Department of Employment, Education and Training

Australische Regierungsdelegation, Canberra

Besuch im Rahmen des Gästeprogramms der BRD
zum Thema „Berufsausbildung in der BRD“

Dr. Gut

Entwicklungsleiter, Ascom Autelca, Schweiz

Dr. Matsumoto

Entwicklungsleiter, NEC, Japan

Dr. Ries

Botschaftsrat der USA in Bonn, Abteilung für Wissenschaft,
Technologie und Umwelt

Graham Samuel

ESF Projektdirektor, SEMA Group, England

Dr. Saunier

Journalist, New York, USA

Maurice Schlumberger

CAP Sesa Innovation, Paris

Dr. Y. Wada und Hayashi

Hitachi, Japan

Impressum

*Herausgeber: Fraunhofer-Institut für Software- und Systemtechnik
Berlin/Dortmund*

Redaktion: Eva Weber

*Fotos: ISST
Jons-Michael Voss*

*Satz und
Layout: Jens-Helge Dahmen*

Titelgrafik: Werbeberatung Herbert Theilig

*Druck und
Verarbeitung: Druckhaus Berlin-Mitte GmbH, Berlin*

