

# Softwaretechnik – Praxiserfahrung im Studium

Alexander B. Schmidt  
Franz Huber  
Institut für Informatik  
Technische Universität München  
14. November 1996

**Im Sommersemester 1996 wurde am Institut für Informatik nunmehr zum dritten Mal ein großes Softwaretechnik-Praktikum durchgeführt, um Studierenden einen Einblick in industrielle Softwareentwicklung zu vermitteln. In drei Monaten entwarfen und implementierten 14 Studierende der Informatik AutoFocus, ein CASE-Tool für die Spezifikation verteilter Systeme. Erste Erfahrungen mit AutoFocus werden zusammen mit der Siemens AG bei der Spezifikation von Multimediadiensten gesammelt.**

Praktische Erfahrung mit großen Softwareprojekten zählt zu den immer wieder geforderten Inhalten der universitären Informatikausbildung. Die Lehrstühle von Prof. Broy und Prof. Endres veranstalten dazu in regelmäßigen Abständen ein Softwaretechnik-Praktikum (STP), das den Studierenden die Möglichkeit bietet, Erfahrungen mit der Durchführung großer Projekte zu sammeln und einen Einblick in industrielle Praxis und Methoden zu gewinnen. Während der Praktika wird der komplette Entwicklungsprozeß eines Software-Projektes vom Auftrag bis zur Auslieferung des Produktes durchlaufen.

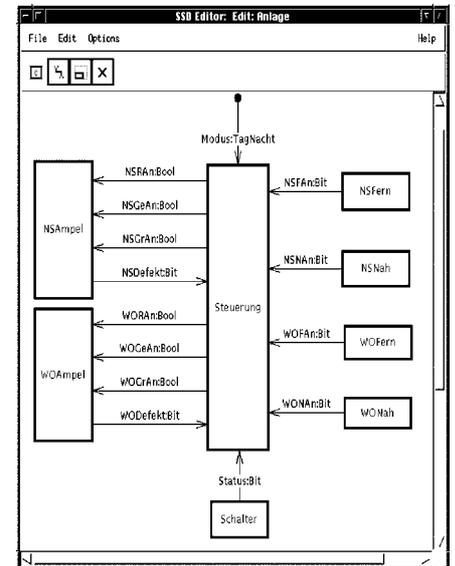
Die vorangegangenen Softwaretechnik-Praktika von 1994 und 1995 waren mit TIS, einem in das World Wide Web (WWW) integrierten Informationssystem mit Anbindung an eine relationale Datenbank und Feedback, einem System zur rechnerunterstützten Generierung und Durchführung von Umfragen, überaus erfolgreich. Denn auch beim STP ist das Ergebnis der Projekte genauso-

wenig vorhersehbar, wie dies bei innovativen Industrieprojekten der Fall ist.

Innovativ war das diesjährige STP auf jeden Fall: Schlagworte wie „Einsatz formaler Methoden“ und „CASE-Tool“ begleiten heute die Entwicklung großer Softwaresysteme. Bei formalen Methoden sind die Beschreibungstechniken und Schritte des Entwicklungsprozesses mathematisch fundiert. Erst damit ist es überhaupt möglich, die Korrektheit eines Entwurfs gegenüber einer Spezifikation oder die Korrektheit einzelner Entwicklungsschritte nachzuweisen. Dies ist besonders für sicherheitskritische Systeme von großer Bedeutung. Klar, daß formale Methoden deswegen auf zunehmendes Interesse seitens der Industrie stoßen. Häufig scheidet deren Einsatz bei industrieller Systementwicklung an oft unhandlichen mathematischen Formalismen. Andererseits werden von Entwicklerteams eine Vielzahl graphischer und intuitiver Beschreibungstechniken eingesetzt, deren Formalismen mathematisch jedoch nicht exakt sind und die somit auch keine beweisbar korrekte Systementwicklung erlauben. Das gleiche gilt, wenn Computer den Entwicklungsprozeß unterstützen: Viele der dazu industriell eingesetzten CASE-Tools (Computer Aided Software Engineering) besitzen ebenfalls keine exakte und eindeutige mathematische Deutung der Vorgehensweise.

AutoFocus soll zwischen beiden Welten eine Brücke schlagen: Eine am Lehrstuhl von Prof. Broy entstandene formale Methode („Focus“) wird mit praxisnahen und industriell erprobten Beschreibungstechniken in einem Werkzeug kombiniert. Mit der projektierten ersten Version von AutoFocus können Anwender ein System aus verschiedenen Sichten modellieren: Mit Strukturdiagrammen wird zunächst der Aufbau eines zu entwickelnden Systems angegeben. Das Verhalten einzelner Systemkomponenten wird von Zustandsübergangsdiagrammen (Automaten) beschrieben und Beispielabläufe des Systems können mit sogenannten Extended Event Traces (EET) graphisch dargestellt werden. Das Besondere in AutoFocus ist: Die Beschreibungstechniken

sind ineinander integriert und hierarchisch. Beispielsweise kann ein Strukturdiagramm mehrere Substrukturen enthalten, deren Verhalten wieder durch Automaten beschrieben werden. Oder AutoFocus prüft, ob ein Automaten dokument mit einem Beispielablauf zusammenpaßt.



Strukturdiagramm einer Ampelsteuerung in AutoFocus

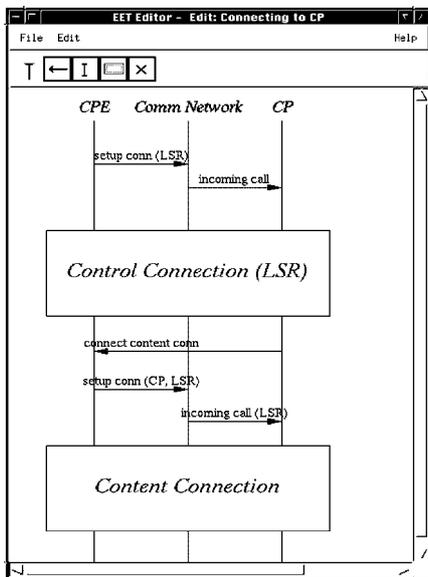
Der Arbeitsweise von Entwicklerteams entsprechend war AutoFocus als offenes System mit Client/Server-Architektur und Mehrbenutzerbetrieb vorgesehen. Als Zugeständnis an heterogene Rechnerwelten und an den Zeitgeist sollte das Projekt plattformunabhängig und objektorientiert mit der neuen Programmiersprache Java implementiert werden.

Soweit die Vorgaben. Zur Realisierung innerhalb des Softwaretechnik-Praktikums standen nach gezielter Werbung für das Projekt letztendlich 14 Studierende (überwiegend aus dem 6. und 8. Fachsemester) zur Verfügung. In Vorstellungsgesprächen wurden die unterschiedlichen Vorkenntnisse und Vorlieben erörtert und darauf basierend fünf Entwicklerteams mit abgeschlossenen Teilaufgaben gebildet. Zwei Mitarbeiter des Lehrstuhls von Prof. Broy fungierten in dem Projekt als Auftraggeber.

Gemäß industrieller Praxis – wenn auch ungewohnt für die Studierenden – muß am Anfang eines Projektes ein Systemspezifikationsdokument erarbei-

tet werden, das dann mit dem Pflichtenheft abgeglichen wird. Bereits hier wird die elementare Bedeutung der Kommunikation im Team deutlich. Die verschiedenen E-Mailverteiler und die von einer studentischen Hilfskraft verwalteten Seiten im World Wide Web, auf denen projektrelevante Dokumente und Informationen für die Beteiligten verfügbar gemacht werden, machen das Internet zu einem unverzichtbaren Informationsmedium.

In wöchentlichen Projektmeetings wurde der Fortschritt des Dokuments referiert und in professionell moderierten Review-Sitzungen der Feinschliff vorgenommen. So erstaunlich es ist, frühestens nach diesem Dokument kann man davon ausgehen, daß alle Beteiligten über das gleiche reden!



Aufbau einer Verbindung zu einem Multi-Mediadienst, dargestellt als EET.

Obwohl das auf die Spezifikation folgende Systementwurfsskizzen aus Zeitmangel nicht in der erforderlichen Detailgenauigkeit erstellt werden konnte, verblieb nach dessen Fertigstellung nur etwa 40% der im Sommersemester verfügbaren Zeit für die Implementierung. Zwar gibt es für die frühen Phasen des Systementwurfs eine Reihe von Entwurfs- und Beschreibungsmethoden, die aber wegen ihres Einarbeitungsaufwandes und des engen Zeitrahmens des STP in den praktikumsbegleitenden Veranstaltungen nur schwerpunktmäßig

angesprochen und damit auch nur ausschnittsweise eingesetzt werden können. Bedenkt man, daß neue Technologien wie Java neben verlockenden Eigenschaften auch unvorhersehbare Schwierigkeiten beinhalten können, ist die für die Implementierung verfügbare Zeit äußerst knapp.

Hilfsmittel wie Javas Klassenbibliotheken, mit denen Bedienoberflächen, Netzwerkverbindungen oder pseudoparallele Prozesse leicht realisiert werden können, beschleunigen den Entwicklungsprozeß. Darüber hinaus erleichtert die syntaktische Ähnlichkeit zum weit verbreiteten C++ vielen Teilnehmern die Einarbeitung. Sollten die kleinen, aber ärgerlichen Unzulänglichkeiten rund um Java in absehbarer Zeit beseitigt werden, ist Java eine gerade für Hochschulprojekte sehr geeignete Sprache.

Während der Implementierung helfen sogenannte Entwurfsmuster („Design Patterns“) den Entwicklern bei der Realisierung von ganzen Programmteilen, wie etwa Undo/Redo-Mechanismen, effiziente Datenhaltung oder Zeichenoperationen. Die Konzepte und Codefragmente der Entwurfsmuster für solche Standardprobleme sind in der Literatur zu finden.

Werden im weiteren Verlauf der Realisierung die ersten Änderungen von Programmfragmenten nötig, wird deutlich, wie wesentlich die Dokumentation des Entwicklers ist. Mit Hilfe von JavaDoc kann aus speziellen Kommentarseiten im Programmquelltext automatisch eine Hypertext-Dokumentation des Quellcodes im HTML-Format erzeugt werden.

Gegen Ende des Praktikums nimmt neben den Tests besonders bei großen Projekten die Integration der teamweise erarbeiteten Komponenten viel Zeit in Anspruch. Ein zu diesem Zweck berufenes Release-Team überwacht und koordiniert die Integration und stellt aus den hunderten von Programmdateien (eine Java-Eigenart) ein installierbares Softwareprodukt zusammen. Das Dokumentationsteam erstellt die Dokumentation sowohl auf Papier als auch im HTML-Format für die Online-Hilfe.

Neben den Testläufen der Entwickler trägt vor allem die enge Zusammenarbeit mit künftigen Benutzern zum Erfolg eines Softwareprojekts bei – bekanntlich ist oft der Wald vor lauter Bäumen nicht zu sehen. In Kooperation mit der Siemens AG soll AutoFocus in einem Pilotprojekt bei der Entwicklung von Multimediadiensten für intelligente Breitbandnetze eingesetzt werden.

Besonders motivierend war – nach erfolgreichem Abschluß des Projekts – damit auf der Systems '96 präsent zu sein. Unter dem Motto „InnovationsMARKT Bayern“ präsentierten die STP-Teilnehmer AutoFocus der Öffentlichkeit.

Klar ist, daß in einem dreimonatigen Praktikum natürlich nur ein Grundstock für ein Systementwicklungswerkzeug wie AutoFocus gelegt werden kann. Die am Lehrstuhl von Prof. Broy geplanten Weiterentwicklungen zielen auf die rechnergestützte Überprüfung von Systemeigenschaften ab, etwa mit automatischen Theorembeweisern oder Programmen zur Modellprüfung („Model Checking“). Einige der Praktikumssteilnehmer werden ihr AutoFocus-spezifisches Know-How in Form von Semester- oder Diplomarbeiten nutzen können.

Unter dem Strich ist das Softwaretechnik-Praktikum gerade wegen der engen Zeitvorgaben, knapp gehaltenen Spezifikationen, den Risiken neuer Technologien und alltäglichen Rechnerproblemen ein besonders reales Abbild der Wirklichkeit. Schrieb doch ein Student zur Frage, was denn sein tiefster Eindruck aus dem STP war: „Error: NFS server sunbroy14 not responding – still trying...“

Alexander B. Schmidt  
Franz Huber