

Erfahrungen mit dem multimedialen, didaktischen Modellierungswerkzeug DAVE

Jörg Pleumann*
Lehrstuhl für Software-Technologie
Universität Dortmund
<joerg.pleumann@uni-dortmund.de>

Abstract:

Professionelle Modellierungswerkzeuge sind für den Einsatz in der Lehre oft ungeeignet. Gründe sind die enorme Komplexität sowie die fehlende Unterstützung für didaktische Einsatzszenarien. Dieser Beitrag stellt als Alternative exemplarisch das didaktische Modellierungswerkzeug DAVE vor, das zur Modellierung und Simulation von UML-Zustandsdiagrammen dient und durch die Integration multimedialer Elemente eine Brücke zwischen abstraktem Formalismus und konkreten Problemen schlägt. DAVE ist Teil einer Familie von didaktischen Modellierungswerkzeugen, die sich ausschließlich an den Bedürfnissen der Lehre orientieren und bereits erfolgreich in Lehrveranstaltungen eingesetzt wurden.

1 Einleitung

Beim Lehren und Lernen graphischer Modellierungssprachen wie der Unified Modeling Language (UML) [BRJ99] ist eine Unterstützung durch Modellierungswerkzeuge sinnvoll und wünschenswert – nicht zuletzt, weil es die Lernenden frühzeitig an einen Umgang mit Werkzeugen gewöhnt, wie er im späteren beruflichen Umfeld üblich ist. Die meisten existierenden Modellierungswerkzeuge (z.B. Rational Rose oder Together) richten sich jedoch ausschließlich an die Zielgruppe der professionellen Software-Entwickler und lassen einen Einsatz in der Lehre völlig außer Acht. Das Ergebnis sind ausgesprochen „schwerewichtige“ Produkte (im Sinne von Funktionalitätsumfang, benötigtem Hauptspeicher und CPU-Leistung), deren reichhaltige Möglichkeiten zwar den Bedürfnissen eines professionellen Umfelds entgegenkommen, aber weit über das hinausgehen, was in einem Praktikum oder einer Übungsgruppe benötigt wird oder angemessen ist. Zu viele Funktionen lenken die Studierenden vom eigentlichen Lehrstoff ab und bergen die Gefahr, daß mehr Zeit in die Erlernung der Verwendung des Werkzeugs als in die eigentlich zu vermittelnde Modellierungssprache investiert wird. Kommen im Verlauf des Curriculums mehrere Modellierungssprachen – und damit mehrere Werkzeuge – zum Einsatz, multipliziert sich dieser Aufwand, da die einzelnen Werkzeuge einander meist nicht ähneln. Bei einer großen Anzahl von Studierenden können auch Lizenzkosten schnell zu einem Problem werden.

*Unterstützt durch das Bundesministerium für Bildung und Forschung (BMBF), Fördernummer 08NM098.

Um diesen Schwierigkeiten zu begegnen, wurde in Dortmund im Rahmen des BMBF-Projektes „MuSoft – Multimedia in der Software-Technik“ [DE02] eine Familie von graphischen Modellierungswerkzeugen ausschließlich für die Lehre entwickelt. Diese Familie umfaßt derzeit verschiedene Vertreter für strukturelle und dynamische Anteile der UML und für Prozeßmodellierung und -begleitung auf der Basis des Unified Process. Bei der Planung und Realisierung dieser Werkzeuge wurde bewußt Wert darauf gelegt, nicht mit professionellen Produkten zu konkurrieren, sondern stattdessen „leichtgewichtige“ Werkzeuge zu schaffen, die auf die Kernfunktionalität des Modellierens reduziert sind. Da alle Werkzeuge die gleiche technische Basis in Form eines Java-Frameworks besitzen, war es möglich, eine einheitliche Benutzeroberfläche zu etablieren, die sich auf notwendige Elemente konzentriert und damit den Einarbeitungsaufwand minimiert. Gleichzeitig wurde didaktisch motivierte Funktionalität in die Werkzeuge eingebracht, die in professionellen Produkten nicht zu finden ist. Das Framework für leichtgewichtige Modellierungswerkzeuge wurde bereits an anderer Stelle beschrieben [APS04]. In diesem Beitrag soll an einem konkreten Beispiel gezeigt werden, wie ein solches Werkzeug um didaktische Funktionalität bereichert werden kann.

Der Rest des Beitrags ist wie folgt gegliedert: Abschnitt 2 zeigt am Beispiel von UML-Zustandsdiagrammen einige typische Probleme, die bei der Lehre graphischer Modellierungssprachen auftreten können. Abschnitt 3 stellt das Modellierungswerkzeug DAVE vor, bei dessen Entwicklung diese Probleme explizit berücksichtigt wurden. Abschnitt 4 berichtet über einen umfangreichen, evaluierten Einsatz von DAVE an der Universität Dortmund. Abschnitt 5 wirft einen Blick auf verwandte Arbeiten. Der abschließende Abschnitt 6 fasst den Beitrag kurz zusammen und gibt einen Ausblick auf künftige Arbeiten.

2 Probleme bei der Lehre graphischer Modellierungssprachen

Als Beispiel für eine graphische Modellierungssprache, deren Lehre von geeigneten Werkzeugen profitieren kann, sollen in diesem Artikel Zustandsdiagramme [Ha87] dienen, eine Teilsprache der UML, die das diskrete Verhalten eines Systems in Form von Zuständen und Übergängen zwischen diesen Zuständen beschreibt. Zustandsdiagramme können zum Beispiel das Verhalten einer Klasse auf einem hohen Abstraktionsniveau oder die erlaubten Aufrufreihenfolgen der Methoden einer Schnittstelle spezifizieren. Im Bereich eingebetteter Systeme, wo Aktualisierungen oder Korrekturen der Software nach dem Ausliefern des Systems nur schwer möglich (weil sich die Software in einem nur lesbaren Speicher befindet) oder sehr teuer (weil damit eine Rückrufaktion verbunden ist) sind, dienen Zustandsdiagramme zur rigorosen Spezifikation und Verifikation des Systemverhaltens vor dessen Implementierung.

Durch ihre vielfältigen Einsatzmöglichkeiten sind Zustandsdiagramme Teil praktisch aller Vorlesungen, die sich mit Softwaretechnik im allgemeinen oder mit UML im besonderen beschäftigen. Ihre Lehre ist jedoch mit einer Reihe von Problemen behaftet, von denen hier stellvertretend zwei herausgegriffen werden sollen: Zustandsdiagramme besitzen eine komplexe Laufzeitsemantik und einen hohen Abstraktionsgrad.

2.1 Komplexe Laufzeitsemantik

Zustandsdiagramme besitzen eine komplexe Laufzeitsemantik, die nicht ohne Grund lange Gegenstand der wissenschaftlichen Diskussion war. Die Feinheiten dieser Semantik sind in Vorlesungen schwer zu vermitteln und werden von den Studierenden oft entsprechend schlecht verstanden. Im Rahmen von Übungsaufgaben konstruierte Modelle sollen helfen, den Formalismus einzuüben. Die semantische Korrektheit eines komplexeren Zustandsdiagramms in Bezug auf gegebene Anforderungen ist aber sowohl für die Studierenden als auch für die am Übungsbetrieb beteiligten Tutoren durch bloßes Hinschauen oder gedankliches Durchspielen einzelner Eingaben schwer zu entscheiden. Durch große Studierendenzahlen wird das Problem der Korrektur solcher Aufgaben noch verschärft. Als Ergebnis prüfen Übungsaufgaben für Zustandsdiagramme vielfach schwerpunktmäßig deren Syntax ab und betrachten die – ungleich wichtigere – Semantik nur oberflächlich.

2.2 Hoher Abstraktionsgrad

Zustandsdiagramme beschreiben Verhalten auf eine sehr abstrakte Weise. Anwendungen von Zustandsdiagrammen in konkreten Systemen können – gerade bei der klassischen Durchführung des Übungsbetriebs mit Papier und Bleistift – bestenfalls als Ausgangspunkt von Übungsaufgaben dienen. Anschließend bewegen sich die Studierenden ausschließlich auf der Ebene des abstrakten Modells. Ein Rückschritt in die Realität und damit eine Verbindung von abstraktem Modell und konkreter Realität ist nicht möglich. Hinzu kommt speziell bei Zustandsdiagrammen die Tatsache, daß diese sich nicht – wie etwa Klassendiagramme – direkt auf Quellcode in einer Programmiersprache abbilden lassen und damit für die Studierenden, abgesehen von einem Erkenntnisgewinn, der sich bei der Modellierung einstellt, keinen unmittelbaren Nutzen haben. Als Ergebnis sind Zustandsdiagramme bei den Studierenden tendenziell eher unbeliebt.

3 Ein didaktisches Modellierungswerkzeug für Zustandsdiagramme

Um Unterstützung bei der Lehre von Zustandsdiagrammen anzubieten, wurde der Dortmunder Automatenvisualisierer und -editor, kurz DAVE¹, entwickelt. DAVE ist ein Modellierungswerkzeug für Zustandsdiagramme, das primär für den Einsatz innerhalb des Übungsbetriebs gedacht ist, aber auch in Vorlesungen oder zum Selbststudium Verwendung finden kann. Abbildung 1 zeigt ein Bildschirmfoto von DAVE mit einem einfachen Beispielmodell einer elektronischen Eieruhr. Die folgenden Abschnitte beschreiben das Werkzeug näher und gehen dabei insbesondere auf jene Funktionalität ein, die sich aus den zuvor beschriebenen didaktischen Problemen ergeben hat.

¹Die namentliche Verwandtschaft von DAVE mit David Harel, dem Erfinder der „klassischen“ Zustandsdiagramme, ist dabei durchaus beabsichtigt.

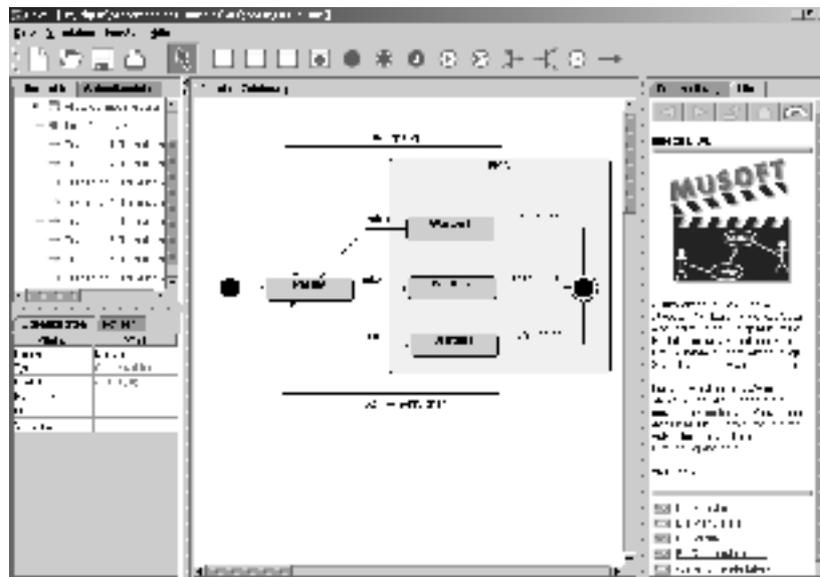


Abbildung 1: Bearbeiten eines Zustandsdiagramms mit DAVE

3.1 Benutzerschnittstelle

Mit Hinblick auf einen Einsatz im Rahmen der Lehre wurde Wert darauf gelegt, die Benutzerschnittstelle von DAVE einfach, intuitiv und möglichst wenig ablenkend zu gestalten. Um einen späteren Umstieg auf professionelle Werkzeuge zu erleichtern, lehnt sie sich strukturell an diese an, ist jedoch vom Umfang her deutlich reduziert. Der Löwenanteil des Bildschirms wird von der graphischen Darstellung des Zustandsdiagramms in Anspruch genommen, da dies der Fokus der Applikation ist. Sämtliche Aspekte, die unmittelbar die Arbeit am Modell betreffen – etwa das Verschieben, Vergrößern, Verkleinern oder Verbinden von Elementen – werden in diesem Bereich gehandhabt. Es ist in keinem Fall notwendig, Menüs zu bemühen, um häufig auftretende Funktionen wie Laden, Speichern oder Drucken aufzurufen oder dem Modell neue Elemente hinzuzufügen. Diese sind in einer Werkzeugleiste am oberen Rand des Fensters verfügbar, wobei Sorge dafür getragen wurde, jede mit einem großen und möglichst aussagekräftigen Symbol auszustatten.

Außer dem Modellierungsbereich und der Werkzeugleiste stellt die Benutzerschnittstelle der Applikation noch einige Komponenten bereit, die aus anderen Anwendungen bekannt sein dürften: Ein „Navigator“ zeigt sowohl eine baumartige Hierarchie des Modells als auch eine Übersicht. Ein „Inspektor“ stellt die Eigenschaften des aktuell ausgewählten Elementes dar und erlaubt ebenso deren Bearbeitung. Beide sind am linken Bildschirmrand angeordnet und können ausgeblendet werden, wenn sie nicht benötigt sind. Am rechten Bildschirmrand befindet sich ein Hypertextbereich, der zum Beispiel zur Integration von Vorlesungsstoff in das Werkzeug genutzt werden kann.

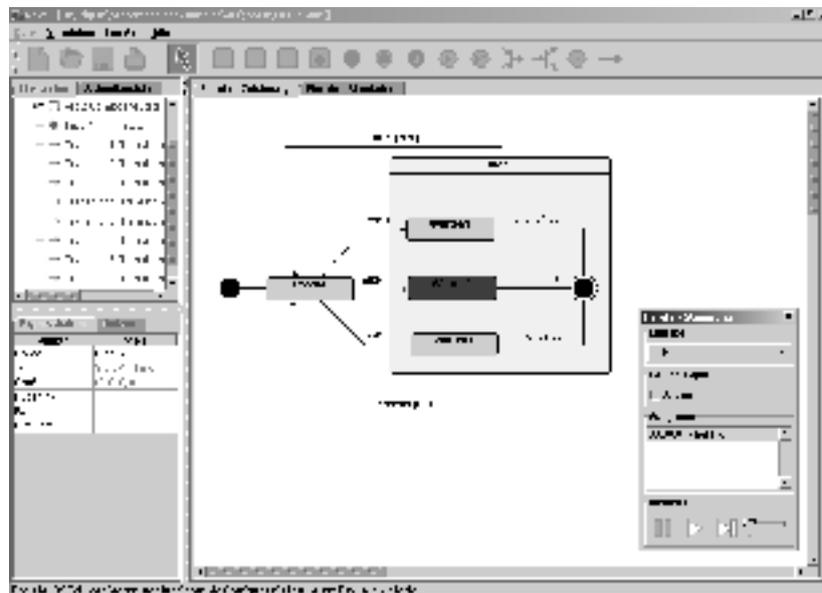


Abbildung 2: Simulieren eines Zustandsdiagramms mit DAVE

3.2 Simulieren von Zustandsdiagrammen

Um eine Brücke zwischen Syntax und Semantik von Zustandsdiagrammen zu schlagen und den Studierenden einen besseren Einblick in deren Laufzeitverhalten zu ermöglichen, unterstützt DAVE die Simulation des erstellten Modells. Basis für diese Simulation ist die in der UML-Spezifikation enthaltene Semantikbeschreibung für Zustandsdiagramme.

Zur Kontrolle der Simulation erscheint ein zweites, kleineres Fenster, dessen unterer Teil der Steuerung einer Stereoanlage ähnelt (siehe Abbildung 2). Die einzelnen Schalter dienen zum Pausieren der Simulation, zum erneuten Starten oder zum Ausführen eines einzelnen Schrittes. Zudem kann die Geschwindigkeit der Simulation festgelegt werden. Für jeden Schritt der Simulation kann im oberen Teil des Fensters ein Ereignis ausgewählt werden, das vom Zustandsdiagramm verarbeitet werden soll. Die beiden Abschnitte in der Mitte des Fensters dienen zur Kontrolle der Überwachungsbedingungen und zur Anzeige von Ausgabeereignissen, die während der Simulation erzeugt werden. Aktive Zustände und schaltende Transitionen werden während der Simulation im Zustandsdiagramm farblich hervorgehoben. Das Diagramm selbst wird während der Simulation in einen nur lesbaren Modus versetzt, damit keine Inkonsistenzen entstehen können.

Während der Simulation wird deren gesamter Verlauf – inklusive aller aktiven Zustände sowie Ein- und Ausgabeereignisse – in einer Tabelle aufgezeichnet. Dies ermöglicht die Analyse und den Vergleich von kompletten Simulationsläufen *post mortem*.

3.3 Multimediale Visualisierung der Simulation

Zum Überwinden der Kluft zwischen abstraktem Modell und realen Anwendungen bietet DAVE die Möglichkeit, die Simulation eines Zustandsdiagramms multimedial zu visualisieren. Wird diese Funktion genutzt, dann erscheint zur Steuerung der Simulation nicht das zuvor erwähnte Fenster, sondern eines, das eine graphische Darstellung eines realen Gerätes enthält. Den Studierenden wird der Eindruck vermittelt, daß das von ihnen erstellte Zustandsdiagramm die (eingebettete) Steuerungssoftware für ein vorhandenes Stück Hardware realisiert. Entsprechend besteht ihre Aufgabe darin, das Verhalten des Gerätes auf der Basis natürlichsprachlicher Anforderungen möglichst gut zu modellieren.

Das im Arbeitsbereich erstellte Modell erhält seine Eingabeereignisse und die aktuellen Werte der Überwachungsbedingungen von diesem Gerät. Ebenso werden Ausgabeereignisse an das Gerät geleitet und wirken sich auf dessen Zustand aus. Zustandsänderungen des Gerätes werden durch Änderungen der graphischen Darstellung wiedergegeben. Gelangt das Gerät in einen fehlerhaften Zustand, dann kann dies ebenfalls entsprechend dargestellt werden, und die Simulation endet. Damit das Zusammenspiel zwischen Modell und Gerät überhaupt möglich ist, werden Ereignisse und Überwachungsbedingungen zuvor (z.B. in einem Aufgabentext) mit den Studierenden vereinbart. Zustände werden nicht vorgegeben, so daß bei der Strukturierung der Lösung völlige Freiheit besteht.

DAVE enthält zur Zeit zwei multimediale Visualisierungen solcher Geräte:

- Eine Waschmaschine, bei der Eigenschaften wie Motorgeschwindigkeit sowie Zu- und Ablauf von Wasser kontrolliert werden können. Das Zustandsdiagramm der Studierenden soll einen kompletten Waschvorgang realisieren. Die Visualisierung gibt Hinweise auf die Korrektheit der Lösung: Zu Beginn der Simulation wird schmutzige Wäsche in die Maschine gelegt. Am Ende eines korrekten Waschvorgangs sollte diese Wäsche sauber sein. Fatale Fehler im Zustandsdiagramm enden zum Beispiel in einer Überschwemmung. Abbildung 3 zeigt einige Bilder der Waschmaschine, die verschiedenen Zuständen der Simulation entsprechen.
- Eine Kaffeemaschine, bei der das Zustandsdiagramm die Heizvorrichtung kontrolliert und korrekt auf die Einstellungen des Benutzers reagieren soll. Zudem müssen bestimmte Vorgaben wie eine automatische Abschaltung nach einer festen Zeitspanne berücksichtigt werden. Auch hier gibt die Visualisierung Hinweise auf die Korrektheit der Lösung: Erscheint Kaffee in der Kanne und schaltet sich die Maschine nach der geforderten Zeit automatisch ab, dann ist die Lösung gutartig – geht die Maschine in Flammen auf, dann ist die Lösung verbesserungsbedürftig.

Beide Beispiele wurden mit geringem Aufwand digital fotografiert und anschließend nachbearbeitet bzw. in „Kacheln“ unterteilt, die je nach aktuellem Zustand kombiniert werden können. Die Umsetzung innerhalb der Simulationssteuerung ist mit einem gewissen, von der Komplexität des gewünschten Verhaltens abhängigen Aufwand verbunden. Da jedoch auf der Basis jeder Visualisierung mehr als eine Übungsaufgabe gestellt werden kann, zahlt sich dieser Aufwand schnell aus. Neue Visualisierungen können dem Werkzeug über eine definierte Schnittstelle hinzugefügt werden.

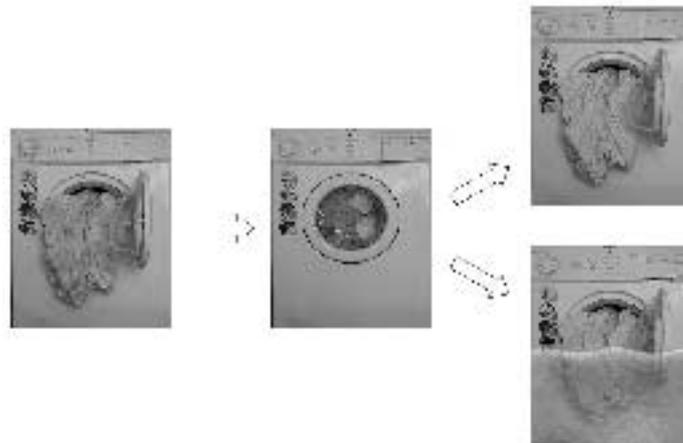


Abbildung 3: Multimediale Visualisierung der Simulation

4 Erfahrungen beim Einsatz in der Lehre

Im Sommersemester 2003 war die Entwicklung von DAVE so weit fortgeschritten, daß ein erster evaluierter Einsatz im Vorlesungsbetrieb möglich war. Die Evaluation wurde vom Hochschuldidaktischen Zentrum (HDZ) der Universität Dortmund unterstützt. Sie war in eine umfangreichere Erhebung eingebettet, innerhalb derer auch andere Fragen untersucht wurden [KMGT⁺04].

4.1 Vorgehen

Aufgrund der thematischen Nähe bot sich für die Evaluation von DAVE die Vorlesung „Software-Technologie“ an, die in Dortmund zu diesem Zeitpunkt im Hauptstudium angeboten wurde. Im Rahmen der Veranstaltung wurden neben anderen Spezifikationsmethoden auch UML-Zustandsdiagramme behandelt. Die Vorlesung wurde von etwa 100 Studierenden besucht, 80 davon haben an den Übungen teilgenommen. Einsatz und Evaluation von DAVE im Rahmen der Veranstaltung gestalteten sich wie folgt:

- Im Anschluß an die üblichen einführenden Vorlesungen zu Zustandsdiagrammen fand eine etwa einstündige Vorlesung zur Verwendung des Werkzeugs statt. Diese Vorlesung wurde von mehreren Mitarbeitern des HDZ beobachtet.
- Zur Bearbeitung der vorlesungsbegleitenden Übungszettel zu Zustandsdiagrammen (zwei Aufgabenzettel mit jeweils zwei umfangreicheren Aufgaben) haben die Studierenden das Werkzeug verwendet. Eine Gruppe von Studierenden wurde bei ihrem Erstkontakt mit DAVE durch das HDZ beobachtet und anschließend befragt.

- Bei der Besprechung des zweiten Aufgabenzettels in den Übungsgruppen wurde ein umfangreicher, mit Hilfe des HDZ erarbeiteter Fragebogen an die Studierenden ausgehändigt. Mit einer einzigen Ausnahme haben alle Studierenden diesen Fragebogen ausgefüllt, so daß die Rücklaufquote der Befragung bei nahezu 100% lag.
- Schließlich fanden noch Interviews mit den zwei Mitarbeitern statt, die für die Betreuung der Übungsgruppen verantwortlich waren.

4.2 Feedback der Studierenden

Die informalen und formalen Befragungen der Studierenden ergaben ein weitgehend positives Bild des Werkzeugs. Die folgenden Abschnitte geben das Feedback der Studierenden in verschiedenen Bereichen, die durch den Fragebogen berührt wurden, wieder.

4.2.1 Erwartungshaltung

Die Studierenden standen dem Werkzeugseinsatz nach der einführenden Vorlesung überwiegend positiv gegenüber (65% Zustimmung, 27% Ablehnung, 8% unentschieden) und sahen den Mehrwert, den das Werkzeug mit sich bringt (76% Zustimmung, 20% Ablehnung, 4% unentschieden). Eine Studierende oder ein Studierender äußerte auf dem Fragebogen, daß der Aufgabenzettel zu DAVE der erste sei, auf den sie oder er sich gefreut habe. Es ist also davon auszugehen, daß bei den Studierenden prinzipiell eine große Offenheit für den Einsatz neuer Medien als unterstützendes Element in einer Vorlesung bzw. im Übungsbetrieb vorhanden ist.

4.2.2 Installation

Die Installation von DAVE bestand im wesentlichen darin, eine ZIP-Datei von der Webseite des Lehrstuhls zu laden, die Datei zu entpacken und anschließend das Programm zu starten. Dies stellte für die meisten Studierenden keine größere Hürde dar (86% hatten keine Probleme, 13% hatten lösbare Probleme, 1% schaffte die Installation gar nicht). Die Probleme bestanden fast ausschließlich darin, daß zunächst eine aktuelle Java-Laufzeitumgebung eingerichtet werden mußte (34% mußten Java erstmalig installieren oder aktualisieren, 63% besaßen bereits ein aktuelles Java, 3% machten keine Angabe).

4.2.3 Verwendung

Die Verwendung des Werkzeugs wurde von den Studierenden positiv beurteilt. Die Oberflächengestaltung wurde als ansprechend empfunden (89% Zustimmung, 10% Ablehnung, 1% unentschieden), was sicherlich auch der professionellen externen Beratung in Gestaltungsfragen zu verdanken ist, die MuSoFT im Laufe des Jahres 2002 erfahren hat. Die Ausführungsgeschwindigkeit war offenbar angenehm (79% Zustimmung, 21% Ablehnung), was bei der Verwendung von Java nicht selbstverständlich ist. Der Ressourcen-

bedarf scheint also im Rahmen dessen zu liegen, was die Rechner der Studierenden (fast alle nutzten das Werkzeug zuhause) bereitstellen. Es wäre interessant, an dieser Stelle eine vergleichende Untersuchung mit professionellen Werkzeugen anzustellen. Dies kann jedoch schlecht im Rahmen einer Vorlesung bzw. des Übungsbetriebes geschehen, da dann möglicherweise ein Teil der Studierenden bewußt benachteiligt würde. Hier wäre ein kontrolliertes Experiment mit Kleingruppen sinnvoller.

4.2.4 Simulation

Ein wesentliches Alleinstellungsmerkmal des Werkzeugs ist die Fähigkeit, Modelle zu simulieren. Diese Funktion wurde von allen Studierenden genutzt (75% ständig, 25% ein- oder mehrmals). Die Studierenden schätzten das frühzeitige Feedback, das sie durch diese Simulationsfunktion erhielten (90% Zustimmung, 8% Ablehnung, 2% Unentschlossen) und sahen die Simulation als Hilfe beim Lösen der Aufgaben (94% Zustimmung, 6% Ablehnung). Die Mehrheit der Studierenden glaubt durch die Simulation ein besseres Verständnis für den zugrunde liegenden Formalismus der UML-Zustandsdiagramme erhalten zu haben (81% Zustimmung, 18% Ablehnung, 1% keine Angabe), was ja eines der erklärten didaktischen Ziele war. Kein einziger Studierender möchte auf die Simulationsfunktion zugunsten eines ausschließlichen späteren Feedbacks durch den Übungsgruppenleiter verzichten (99% Ablehnung, 1% unentschlossen).

4.2.5 Visualisierung

Von den insgesamt vier Aufgaben waren die zwei, die durch multimediale Darstellungen echter Geräte visualisiert und unterstützt wurden, am beliebtesten. Die Möglichkeit zur Visualisierung wurde von den meisten Studierenden angenommen (38% bei jeder Aufgabe, 49% ein- oder mehrmals, 10% niemals)². Mit dem für sie neuen Konzept, ein Modell nicht „auf der grünen Wiese“, sondern zur Steuerung eines gedachten Gerätes zu erstellen, hatten die Studierenden keine Schwierigkeiten. Nur wenigen Studierenden war unklar, wie ihr Modell technisch mit der Visualisierung der Waschmaschine bzw. Kaffeemaschine zusammenspielt (13% Zustimmung, 85% Ablehnung, 2% unentschlossen).

Die meisten Studierenden fühlten sich durch die anschaulichen Beispiele angesprochen (75% Zustimmung, 22% Ablehnung, 3% unentschlossen) und sind der Meinung, daß diese dazu beitragen, die abstrakten Probleme deutlicher zu machen (63% Zustimmung, 30% Ablehnung, 7% unentschlossen). Dies kann als Bestätigung für die didaktische Idee der Brücke zwischen abstraktem Formalismus und Realität gewertet werden.

4.2.6 Gesamteindruck

Insgesamt wurde das Werkzeug von den Studierenden positiv beurteilt. Die Mehrzahl würde es anderen Studierenden weiterempfehlen (75% Zustimmung, 25% Ablehnung).

²Die Frage war ungünstig formuliert. Da nur die Hälfte der Aufgaben überhaupt die multimediale Visualisierung unterstützte, war es eigentlich nicht möglich, sie bei jeder Aufgabe zu nutzen

Ein sehr großer Teil der Studierenden wünscht sich ähnliche Werkzeuge für andere Themenbereiche der Vorlesung (86% Zustimmung, 14% Ablehnung). Hierfür bestehen bereits konkrete Pläne.

Kritisch angemerkt wurde von vielen Studierenden, daß sie sich während der Arbeit mit dem Werkzeug beeinträchtigt gefühlt haben (40% fühlten sich beeinträchtigt, 47% nicht, 13% waren unentschlossen oder haben keine Angabe gemacht). Als Grund wurde hier fast ausschließlich die Unausgereiftheit des Programms angegeben, was angesichts der Tatsache, daß dies der erste Einsatz war, nicht verwunderlich ist. Die Studierenden zeigten eine Reihe von Schwachstellen und Fehlern auf, die im Anschluß an den Einsatz korrigiert werden konnten. Außerdem wurden einige Funktionen gewünscht, die in der ersten Fassung von DAVE nicht vorhanden waren, aber inzwischen teilweise integriert werden konnten.

4.3 Feedback der Betreuer

Das Feedback der beiden Übungsgruppenbetreuer war ebenfalls positiv. Hervorgehoben wurde die Erleichterung beim Korrigieren der Aufgaben, da hier ebenfalls die Simulationsfunktion des Werkzeugs zum Einsatz kommen konnte. Außerdem ergab sich in der Diskussion, daß mit dem Werkzeug neue Aufgabentypen möglich sind, die sich bei einer Bearbeitung mit Papier und Bleistift nicht anbieten: Die Aufgaben können, da sie durch die Simulation unterstützt werden, insgesamt komplexer und damit realistischer werden. Wo traditionelle Übungszettel zu Zustandsdiagrammen oft im wesentlichen die Syntax abfragen, erlaubt die Simulation die gewünschten Einblicke in deren Laufzeitsemantik.

4.4 Ergebnisse

Die Ergebnisse beziehen sich zunächst nur auf das beschriebene Werkzeug DAVE. Ein großer Teil der Ergebnisse ist jedoch unmittelbar auf die weiteren Mitglieder der Dortmunder Werkzeugfamilie übertragbar. Da insbesondere die Benutzerschnittstelle Teil des zugrunde liegenden Frameworks ist, gelten Aussagen zur Usability unmittelbar für die gesamte Werkzeugfamilie. Nicht zuletzt profitierten die weiteren Werkzeuge auch von den im Anschluß an die Evaluation durchgeführten Korrekturen bzw. Erweiterungen, so daß sich das Framework aus Wartungsgesichtspunkten bereits bewährt hat.

DAVE wurde im Wintersemester 2003/2004 in Dortmund erneut eingesetzt, diesmal in einer Grundstudiumsveranstaltung zur UML-Modellierung. Hier nutzten etwa 350 Studierende das Werkzeug zum Lösen der Übungsaufgaben. Auch dieser Einsatz wurde durch einen Fragebogen evaluiert. Die Ergebnisse aus dem ersten Einsatz wurden dabei bestätigt und teilweise sogar übertroffen.

5 Verwandte Arbeiten

Werkzeuge die neben der Modellierung von Zustandsdiagrammen auch deren Ausführung in Form von generiertem Code unterstützen, sind Rational Rose RT und iLogix StateMate. Beides sind jedoch industrielle Werkzeuge, die unter dem Komplexitätsaspekt für die Lehre eher ungeeignet sind. Das im akademischen Umfeld entstandene FUJABA [NNZ00] arbeitet ebenfalls mit Code-Generierung, ist jedoch weniger komplex als die beiden erstgenannten und wird dementsprechend häufiger im Lehrbetrieb eingesetzt. Es enthält mit der Visualisierungskomponente Mr. Dobs eine Möglichkeit zur Veranschaulichung von Objektstrukturen, die ähnlich motiviert ist wie die in DAVE enthaltenen Alltagsgeräte. Die in der Einleitung erwähnten Werkzeuge Together und Rational Rose unterstützen keine Ausführung des Modells.

Während Werkzeuge mit explizit didaktischer Ausrichtung für den Bereich der Modellierung insgesamt noch relativ rar sind, existiert für die Lehre objektorientierter Programmierung bereits eine ganze Reihe solcher Werkzeuge: BlueJ [KQPR03] ist eine integrierte graphische Java-Entwicklungsumgebung für pädagogische Einsatzzwecke. Neben einer generellen Reduktion der Komplexität im Vergleich zu professionellen Werkzeugen ist ein wesentliches Merkmal von BlueJ die technische Unterstützung des „Objects First“-Ansatzes: Die Studierenden können unmittelbar Klassen instanziiieren und mit den Objekten interagieren, ohne daß dafür ein komplettes Programm notwendig wäre. Sie erhalten so – ähnlich zu DAVE – frühzeitiges Feedback und Einblicke in die Semantik von Klassen und Objekten. Ähnliche Ziele verfolgt auch die Umgebung jGrasp [HCB04], bei der zur Objektinteraktion unter anderem noch die Möglichkeit der graphischen Visualisierung von Datenstrukturen hinzukommt. Das Werkzeug DrJava [ACS02] unterstützt die Objektinteraktion auf textuelle Weise, indem in einer speziellen Eingabezeile Java-Anweisungen eingegeben und anschließend interpretiert werden. Alle Werkzeuge betonen den leichtgewichtigen Ansatz, der sie von professionellen Werkzeugen unterscheidet.

6 Zusammenfassung und Ausblick

Beim Lehren und Lernen graphischer Modellierungssprachen wie der UML ist die Unterstützung durch Werkzeuge wünschenswert. Vielfach sind Schulen und Hochschulen aber in Ermangelung von Alternativen gezwungen auf Werkzeuge zurückzugreifen, deren eigentliche Zielgruppe professionelle Entwickler sind. Diese Werkzeuge sind für den Einsatz in der Lehre schon aufgrund ihrer Komplexität eher ungeeignet. Zudem besitzen sie üblicherweise keinerlei didaktisch motivierte Funktionalität, die bei Problemen der Vermittlung einer konkreten Modellierungssprache Unterstützung leisten könnte.

Als eine Alternative zu professionellen Werkzeugen wurde exemplarisch das Werkzeug DAVE vorgestellt, das die Lehre von UML-Zustandsdiagrammen unterstützt. Die bewußt einfach gehaltene Benutzeroberfläche macht das Werkzeug für nichtprofessionelle Benutzer wie Schüler oder Studierende sehr leicht zugänglich, so daß der Einarbeitungsaufwand gering ausfällt. Durch die Simulationsfunktion wird sowohl die generelle Semantik von

Zustandsdiagrammen als auch das Verhalten eines konkreten Modells ersichtlich. Gleichzeitig erhalten die Studierenden das für den Lernerfolg wichtige frühzeitige Feedback zu ihrer Lösung. Die zusätzliche multimediale Visualisierung der Simulation schlägt eine Brücke zwischen abstraktem Formalismus und realen Problemen und trägt so insbesondere zur Motivation der Studierenden bei. Damit wird DAVE zu einer echten Bereicherung gegenüber der klassischen Lehre von Zustandsdiagrammen – sei es auf der Basis von Papier und Bleistift oder unterstützt durch reine Modellierungswerkzeuge. Die Evaluation des Einsatzes von DAVE im Sommersemester 2003 bestätigt dies.

DAVE ist Teil einer Familie von didaktischen Werkzeugen, die auf einem gemeinsamen Java-Framework basieren, das in Dortmund entstanden ist. Zu den weiteren Vertretern dieser Familie gehören Werkzeuge für Softwarearchitekturen, Petrinetze und den Unified Process. Im Laufe des Jahres 2004 soll auf Basis der Erfahrungen mit DAVE ein weiteres UML-Werkzeug entstehen, das verschiedene strukturelle und dynamische Diagrammtypen vereint und simulierbar macht. Alle Werkzeuge stehen im Internet unter der Adresse <http://www.softwaretechnik.de> zur Verfügung.

Literatur

- [ACS02] Allen, E., Cartwright, R., und Stoler, B.: DrJava: A Lightweight Pedagogic Environment for Java. In: *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education*. 2002.
- [APS04] Alfert, K., Pleumann, J., und Schröder, J.: Software Engineering Education needs Adequate Modeling Tools. In: *Proceedings of the 17th Conference on Software Engineering Education and Training*. 2004.
- [BRJ99] Booch, G., Rumbaugh, J., und Jacobson, I.: *The Unified Modeling Language User Guide*. Addison Wesley Longman. 1999.
- [DE02] Doberkat, E.-E. und Engels, G.: MuSoft – Multimedia in der SoftwareTechnik. *Informatik Forschung und Entwicklung*. 17(1):41–44. 2002.
- [Ha87] Harel, D.: Statecharts: A visual formalism for complex systems. *Science of Computer Programming*. 8(3):231–274. June 1987.
- [HCB04] Hendrix, D., Cross, J., und Borowski, L.: An Extensible Framework for Providing Dynamic Data Structure Visualizations in a Lightweight IDE. In: *Proc. of the 35th SIGCSE Technical Symposium on Computer Science Education*. ACM Press. 2004.
- [KMGT⁺04] Kamphans, M., Metz-Göckel, S., Tigges, A., Drag, A., und Schröder, E. Evaluation des Editors DAVE in der informatischen Hochschullehre. Technischer Bericht des Lehrstuhls für Software-Technologie der Universität Dortmund. 2004. MuSoft-Bericht Nr. 6.
- [KQPR03] Kölling, M., Quig, B., Patterson, A., und Rosenberg, J.: The BlueJ System and its Pedagogy. *Journal of Computer Science Education, Special Issue on Learning and Teaching Object Technology*. 13(4). December 2003.
- [NNZ00] Nickel, U., Niere, J., und Zündorf, A.: Tool demonstration: The FUJABA environment. In: *Proc. of the 22nd International Conference on Software Engineering (ICSE), Limerick, Ireland*. S. 742–745. ACM Press. 2000.