

# Didaktische Modellierungswerkzeuge für die Präsenzlehre der Softwaretechnik\*

Jörg Pleumann

Jens Schröder

Lehrstuhl für Software-Technologie  
Universität Dortmund

{pleumann,schroeder}@ls10.cs.uni-dortmund.de

## Abstract:

Dieser Aufsatz stellt eine Familie von graphischen Modellierungswerkzeugen vor, die zur Unterstützung der Präsenzlehre in der Softwaretechnik entwickelt wurden. Die Werkzeuge zeichnen sich durch einen auf die Lehre zugeschnittenen Funktionsumfang und eine intuitive Benutzeroberfläche aus. Zur Unterstützung des Lehrens und Lernens des jeweiligen Formalismus wurden in einige Werkzeuge Simulationsfunktionen und multimediale Elemente integriert. Evaluierete Einsätze der Werkzeuge brachten vielversprechende Ergebnisse und bestätigten unseren Ansatz.

## 1 Einleitung

Im Rahmen des BMBF-Projekts „MuSoFT – Multimedia in der Softwaretechnik“ [DE02], an dem acht Partner von insgesamt sieben deutschen Hochschulen beteiligt waren, wurden multimediale Lehrmaterialien zur Unterstützung der Präsenzlehre in der Softwaretechnik erstellt. Jeder Projektpartner hat sich der Aufarbeitung eines oder mehrerer Themenkomplexe dieses praktischen Teilbereichs der Informatik gewidmet. Die Teilprojekte an der Universität Dortmund haben sich mit den Themen Zustandsdiagramme, Softwarearchitektur und Prozessmodellierung auf der Basis der Unified Modeling Language (UML) [BRJ99] und des Unified Process [JBR98] befasst. In allen drei Themen werden – speziell im Übungsbetrieb oder in Praktika – üblicherweise industrielle Modellierungswerkzeuge verwendet. Der Einsatz solcher Werkzeuge in der Lehre ist mit einer Reihe von Problemen behaftet und dementsprechend oft unbefriedigend. Deshalb haben wir uns entschlossen, die Lehre dieser Themen durch spezielle Werkzeuge zu unterstützen, die auf didaktischen Ideen basieren und durch die Integration multimedialer Elemente zu größerer Anschaulichkeit bei der Lehre der entsprechenden Formalismen beitragen.

Der weitere Aufsatz ist wie folgt strukturiert: Zunächst betrachten wir in Abschnitt 2 die Unzulänglichkeiten industrieller Modellierungswerkzeuge, bevor wir in Abschnitt 3 die in Dortmund entwickelten Werkzeuge vorstellen. Abschnitt 4 berichtet über Erfahrungen beim Einsatz der Werkzeuge. Abschnitt 5 fasst den Aufsatz knapp zusammen.

---

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

## 2 Probleme beim Einsatz von Modellierungswerkzeugen

Beim Lehren und Lernen graphischer Modellierungssprachen ist eine Unterstützung durch Werkzeuge prinzipiell sinnvoll und wünschenswert – nicht zuletzt, weil es die Lernenden frühzeitig an einen Umgang mit Werkzeugen gewöhnt, wie er im professionellen Umfeld Standard 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 schwergewichtige Produkte (im Sinne von Funktionalität, benötigtem Hauptspeicher und CPU-Leistung), deren reichhaltiger Funktionsumfang zwar den Bedürfnissen eines professionellen Umfelds entgegenkommt, aber weit über das hinausgeht, was in einem Praktikum oder einer Übungsgruppe benötigt wird oder angemessen ist. Zu viele Funktionen lenken die Studierenden vom eigentlichen Lehrstoff ab und führen dazu, dass mehr Zeit in die Erlernung der Verwendung des Werkzeugs als in die eigentlich zu vermittelnde Modellierungssprache investiert wird. Kommen 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.

Zusätzlich treten bei der Lehre graphischer Modellierungssprachen Probleme auf, mit denen professionelle Entwickler nicht konfrontiert sind und die dementsprechend von industriellen Werkzeugen nicht berücksichtigt werden: Manche Sprachen – zum Beispiel die erwähnten Zustandsdiagramme – besitzen eine komplexe Laufzeitsemantik, deren Feinheiten in Vorlesungen schwer zu vermitteln sind und von den Studierenden oft entsprechend schlecht verstanden werden. Im Rahmen von Übungsaufgaben konstruierte Modelle sollen helfen, den Formalismus einzuüben, sind aber auch für die am Übungsbetrieb beteiligten Tutoren durch bloßes Hinschauen oder gedankliches Durchspielen oft nur schwer zu korrigieren. Als Ergebnis prüfen Übungsaufgaben oft schwerpunktmäßig Syntax ab und betrachten die – ungleich wichtigere – Semantik nur oberflächlich. Ein weiteres Problem ist der naturgemäß hohe Abstraktionsgrad graphischer Modellierungssprachen. Der Bezug zwischen abstraktem Modell und realen Anwendungen ist für die Studierenden oft schwer zu erkennen, worunter neben dem Verständnis oft auch die Motivation leidet.

## 3 Didaktische Modellierungswerkzeuge

Um den genannten Schwierigkeiten zu begegnen, wurde an der Universität Dortmund eine Familie von graphischen Modellierungswerkzeugen ausschließlich für die Lehre entwickelt. Bei der Planung und Realisierung dieser Werkzeuge wurde 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 [APS04] besitzen, war es möglich, eine einheitliche Benutzeroberfläche zu etablieren, die sich auf wenige Elemente konzentriert und damit den Einarbeitungsaufwand minimiert. Zusätzlich wurde didaktisch motivierte Funktionalität in die einzelnen Werkzeuge eingebracht, die

in professionellen Produkten in dieser Form nicht zu finden ist. So enthalten zum Beispiel alle Werkzeuge einen Bereich, in welchem Hypertext angezeigt und mit den Elementen eines Modells verknüpft werden kann. Im Folgenden stellen wir einige der Werkzeuge vor.

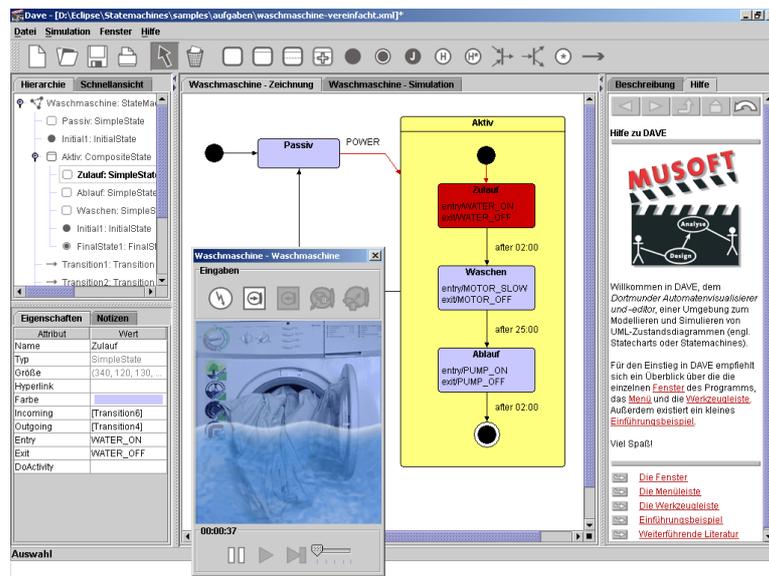


Abbildung 1: Modellierungswerkzeug DAVE mit multimedialer Simulation

Der Dortmunder Automatenvisualisierer und -Editor, kurz **DAVE**, ist eine graphische Umgebung zum Lehren und Lernen von UML-Zustandsdiagrammen. Er verzichtet auf fortgeschrittene, aber im Kontext der Lehre unnötige Funktionalität professioneller CASE-Tools. Stattdessen wurde eine Simulationsmaschine integriert, die den Studierenden das Durchspielen des erstellten Modells auf Basis der in der UML-Spezifikation angegebenen Laufzeitsemantik ermöglicht und damit diese Semantik verständlicher macht. Gleichzeitig besteht die Möglichkeit, die Simulation mit der multimedialen Visualisierung eines Alltagsgerätes – etwa einer Wasch- oder einer Kaffeemaschine – zu verbinden. Den Studierenden wird damit der Eindruck vermittelt, daß ihr Modell die (eingebettete) Steuerungssoftware für das entsprechende Gerät realisiert. Die Auswirkungen von fehlerhaften Modellierungen werden ebenfalls sehr plastisch dargestellt (siehe Abbildung 1). Die multimediale Visualisierung erhöht somit für die Studierenden gleichermaßen Anschaulichkeit wie Motivation. Für die Lehrenden bieten Simulation und Visualisierung Unterstützung bei der Präsentation in Vorlesungen und der Korrektur von Übungsaufgaben.

Ähnliche Ideen wie DAVE verfolgt auch der Softwarearchitektur-Modellierer **SAM**. Er berücksichtigt jedoch zusätzlich strukturelle Aspekte eines Softwaresystems, so daß eine komplexe, aus Komponenten und Konnektoren bestehende Architektur modelliert und simuliert werden kann. Der Ereignisfluss zwischen den Komponenten ist unter anderem in einem Sequenzdiagramm beobachtbar.

Das Prozessmodellierungswerkzeug **ProModUP** ermöglicht das Maßschneidern des ge-

nerischen Unified Process auf die Bedürfnisse eines konkreten Softwareprojekts. Es bietet den Lernenden syntaktische Unterstützung innerhalb dieses mehrschichtigen Modells und erlaubt den Lehrenden eine Anpassung der zur Verfügung stehenden Elemente und Rollen an die besonderen Bedürfnisse von Praktika in Grund- und Hauptstudium. Zur Unterstützung der Studierenden in einem solchen Praktikum dient das ergänzende Tutorwerkzeug **ProTut**. Basierend auf einem erstellten Prozessmodell leitet ProTut die Studierenden bei der Durchführung des modellierten Prozesses an, erlaubt die Verfolgung und Visualisierung des Projektfortschritts, die Generierung von ToDo-Listen und die übersichtliche schematische Darstellung des Projektablaufs in einem Gantt-Diagramm.

## 4 Erfahrungen

Im Sommersemester 2003 war die Entwicklung von DAVE so weit fortgeschritten, dass ein erster evaluierter Einsatz im Rahmen der Hauptstudiumsveranstaltung „Softwaretechnologie“ möglich war. Die Veranstaltung wurde von etwa 80 Studierenden besucht. Zustandsdiagramme wurden, wie üblich, zunächst im Rahmen der Vorlesung eingeführt, ehe das Werkzeug vorgestellt und dann für die Dauer von zwei Wochen zum Lösen der vorlesungsbegleitenden Übungszettel genutzt wurde. Hierbei fanden unter anderem die multimedialen Visualisierungen von Wasch- und Kaffeemaschine Verwendung. Der gesamte Einsatz wurde durch das Hochschuldidaktische Zentrum (HDZ) der Universität Dortmund begleitet [KMGT<sup>+</sup>04] und neben Interviews durch einen umfangreichen Fragebogen – mit einer Rücklaufquote von fast 100% – evaluiert.

Die Evaluation ergab ein weitgehend positives Bild des Werkzeugs. Die generelle Idee eines leichtgewichtigen Modellierungswerkzeugs, das spezielle Akzente auf einfache Benutzerführung und einen an die Lehre angepassten Funktionsumfang setzt, war offenbar erfolgreich. Die Oberfläche wurde als ansprechend empfunden (89% Zustimmung, 10% Ablehnung, 1% unentschlossen). Die Ausführungsgeschwindigkeit war weitgehend angenehm (79% Zustimmung, 21% Ablehnung), lässt aber Spielraum für Optimierungen.

Die Simulationsfunktion liefert didaktisch wichtiges, frühzeitiges Feedback. Dies sahen auch die meisten Studierenden so (90% Zustimmung, 8% Ablehnung, 2% unentschlossen). Außerdem trägt sie zu einem besseren Verständnis des Formalismus bei (81% Zustimmung, 18% Ablehnung, 1% keine Angabe). Entsprechend wurde sie von allen Studierenden genutzt (75% ständig, 25% ein- oder mehrmals). Die multimediale Visualisierung der Simulation macht nach Aussage der Mehrheit der Studierenden abstrakte Probleme anschaulicher (63% Zustimmung, 30% Ablehnung, 7% unentschlossen). Die gewählten Beispiele waren ansprechend (75% Zustimmung, 22% Ablehnung, 3% unentschlossen). Die Möglichkeit zur Visualisierung wurde von den meisten Studierenden angenommen (38% bei jeder Aufgabe, 49% ein- oder mehrmals, 10% niemals).

Insgesamt wurde das Werkzeug von den Studierenden positiv beurteilt. Die meisten würden es anderen Studierenden weiterempfehlen (75% Zustimmung, 25% Ablehnung) und wünschen sich ähnliche Werkzeuge für weitere Themenbereiche der Vorlesung (86% Zustimmung, 14% Ablehnung). Diese Ergebnisse, die durch die gemeinsame technische Ba-

sis in Teilen auf die anderen Vertreter der Werkzeugfamilie übertragbar sind, wurden durch einen zweiten Einsatz im Wintersemester 2003/2004 bestätigt. Hier wurde auch erstmalig das Werkzeug ProModUP eingeführt und – in Verbindung mit ProTut – in einem anschließenden Praktikum als Mittel zur Projektplanung verwendet und evaluiert. Die Ergebnisse dieser Evaluation liegen allerdings noch nicht vor.

## 5 Zusammenfassung

In der Lehre der Softwaretechnik an Schulen und Hochschulen kommen vielfach industrielle Modellierungswerkzeuge zum Einsatz, deren Funktionsumfang sich an den Bedürfnissen professioneller Softwareentwickler orientiert. Der Einsatz solcher Werkzeuge in der Lehre ist oft unbefriedigend.

In diesem Aufsatz haben wir als Alternative eine Familie von graphischen Modellierungswerkzeugen vorgestellt, die primär für den Einsatz in der Lehre entwickelt wurden. Sie zeichnen sich durch einen reduzierten Funktionsumfang, eine überschaubare und intuitive Benutzeroberfläche und zusätzliche didaktische Funktionalität aus. Speziell letztere bietet für Lehrende wie Lernende einen echten Mehrwert gegenüber professionellen Produkten und bereichert somit die Präsenzlehre der Softwaretechnik. Die Werkzeuge basieren auf einem gemeinsamen Java-Framework, das auch die Basis für weitere Werkzeuge dieser Art bildet. So entsteht derzeit im Rahmen einer Diplomarbeit ein Editor und Simulator für Petrinetze, der die Konzepte von DAVE auf diesen Bereich zu übertragen versucht. Die Erfahrungen beim Einsatz der Werkzeuge waren vielversprechend. Alle Werkzeuge stehen im Internet unter <http://www.softwaretechnik.de> zur Verfügung.

**Danksagung:** Wir danken dem Team des Hochschuldidaktischen Zentrums der Universität Dortmund für die Unterstützung bei der Evaluation der Werkzeuge.

## Literatur

- [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*. S. 72–77. 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.
- [JBR98] Jacobson, I., Booch, G., und Rumbaugh, J.: *The Unified Software Development Process*. Addison Wesley. 1998.
- [KMGT<sup>+</sup>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.