

Konzept-Tastaturen zur Algorithmenanimation

N. Baloian, H. Breuer, W. Luther, Chr. Middleton, Th. Pilz

Institut für Informatik und Interaktive Systeme (IIS)

Universität Duisburg-Essen

Lotharstr. 65

D-47048 Duisburg

{nbaloian,hbreuer,cmiddlet}@dcc.uchile.cl

{luther,pilz}@informatik.uni-duisburg.de

Abstract: Softwarevisualisierung und Algorithmenanimation sind wichtige Hilfsmittel der Informatikausbildung. Daher schlagen wir eine Interaktionsunterstützung zur aktiven Steuerung der Algorithmen mittels adaptiver Concept Keyboards vor. Aus dem Quelltext der Standardalgorithmen können Studierende Tastaturen frei erzeugen und konfigurieren, Methoden ausführen und so die Datenstrukturen selbst gesteuert explorieren. Zum Vergleich wird ein einfaches Schritttinterface angeboten. Nach der Evaluation ist ein vorlesungsbegleitender Einsatz geplant.

1 Projektumfeld und Anwendungskontext

An der Universität Duisburg-Essen sind am Campus Duisburg in den letzten Jahren Hörsäle und Seminarräume sowie die Servicebereiche Rechenzentrum, Bibliothek und das Audio-Visuelle Medienzentrum mit hochschulweit vernetzter multimedialer Präsentations- und Arbeitsplatztechnik ausgestattet worden. Diese unterstützt neben Video-Conferencing auch die Planung, interaktive Annotation und Anreicherung des Materials aus der Diskussion der Lehrenden mit den Studierenden heraus an einer elektronischen Tafel bzw. Wireless-Notebooks oder an Plätzen mit Standrechnern [Ba04]. So können sich Studierende campusweit in ein drahtloses Netz einloggen, von Experten unterstützt Arbeitsmaterialien auswählen, erfassen, gemeinsam bearbeiten und verteilen [BL03].

Diese Infrastruktur soll in Vorlesungen der Informatik genutzt werden, um dynamische Inhalte zu präsentieren aber auch Übungsbeispiele individuell zu entwerfen und mithilfe adäquater Schnittstellen zu bearbeiten. Dazu wurde das Java-Tool CONKAV zur Erzeugung und Konfiguration spezieller Tastaturlayouts, so genannter Concept Keyboards (CK) zur Algorithmenanimation und -visualisierung entwickelt. Es erlaubt den Studierenden, Datenstrukturen zu explorieren und grundlegende Prozeduren und Methoden auf selbst definierte Anwendungsbeispiele anzuwenden. CKs verwenden anstelle der Standardtastatur für spezielle Anwendungen gestaltete Tastenfelder. Jede Taste repräsentiert dabei ein Konzept oder aktiviert eine Funktion. Die Idee zu CONKAV wurde in [BL01] vorgestellt, das System in der Entwicklungszeit mehrfach evaluiert. Es wird demnächst in Vorlesungen zu Algorithmen und Datenstrukturen unterrichtsbegleitend eingesetzt.

2 Algorithmenanimation und ihre Visualisierung

Seit mehreren Jahrzehnten nutzen Lehrer und Lerner Algorithmenanimationen, um dynamische Inhalte besser zu erklären oder zu verstehen. Visualisierungen zeigen relevante Parameter oder Variablen, den aktuellen Zustand der Datenstrukturen in einer visuellen Repräsentation, oder eine animierte formale Beschreibung des Algorithmus. Komplexe Strukturen werden vereinfacht, indem nur die wichtigen Teile der Datenstruktur oder Speicherbelegung gezeigt, unwichtige zusammengefasst oder ausgeblendet, die Datenmenge reduziert, die Ablaufgeschwindigkeit verlangsamt und Übergänge zwischen Zuständen verstetigt werden. In vielen Fällen kann der Lerner den Algorithmus Schritt für Schritt ausführen, eigene Konfigurationen oder Eingangsdaten testen oder gestufte Hilfestellungen erhalten. Algorithmenanimationen (AA) zu allen Bereichen nicht nur der Informatik und Animationssysteme, die bei der Ausführung der Algorithmen eine semi-automatische Erzeugung der Illustrationen ermöglichen, wurden so unter verschiedenen Fragestellungen in aktuellen Arbeiten untersucht [Na03] und in umfangreichen Sammlungen [Rö02; Cr02] bereitgestellt. Andere Systeme unterstützen die Lerner bei der Konstruktion eigener Visualisierungen [Hu99] oder liefern ein Rahmenwerk zur Erstellung von Tools zum Lernen von Algorithmen mit Animationen, Übungen und Simulationen [Fa02]. AA-Systeme werden in [Rö03] klassifiziert und Methoden zur Erzeugung der Animationen beschrieben. Für den Bereich der Computergrafik und Bildverarbeitung haben die Autoren mehrere Lehr- und Lernsysteme entwickelt, in Vorlesungen und Übungen eingesetzt und unter verschiedenen Fragestellungen untersucht [Ja98; Lu00].

3 Concept Keyboards zur Algorithmenanimation

Der im BMBF-Projekt ‚Networked Multimedia Systems for Innovative Teaching and Learning Scenarios‘ der Universitäten von Chile und Duisburg verfolgte Ansatz zielt nicht so sehr auf ausgefeilte Visualisierungstechniken. Eher trägt er zur Konstruktion kontextabhängiger Schnittstellen zur Steuerung der Aktionen im Kern des Algorithmus bei, unabhängig von der Eingabe der Daten und der grafischen oder akustischen Ausgabeform der Datenstruktur. Zu bekannten Implementationen von Algorithmenvisualisierungen generiert der Nutzer semiautomatisch CKs, die ihm die schritt- und methodenweise Ausführung des Algorithmus erlauben. Das Interface unterstützt die Durchquerung der Datenstruktur Schritt für Schritt, das Ausführen und Zurücknehmen (Undo) von Teilschritten über den Aufruf der jeweiligen Methoden bzw. Eingabe der nötigen Parameter.

Der Zugang erlaubt es, Lernfortschritte auf verschiedenen Ebenen zu erzielen. Nach einem Kennenlernen des Algorithmus mit einem Schrittinterface dienen die Ausführung seiner Methoden in der richtigen Reihenfolge und die Erzeugung geeigneter Startwerte dem Verständnis. Eine Konfiguration des CK mit Methoden verschiedener Granularität erfordert die Analyse des Algorithmus bis hin zu seiner Reimplementation in einer Form, die die automatische Generierung der CKs unterstützt. Unsere methodische Vorgehensweise kann in der folgenden Form beschrieben werden:

- Entwerfe für jeden Algorithmus ein passendes CK durch Neudefinition von Tasten, z. B. auf einer Spezialtastatur oder durch Buttons auf einem Grafiktablett.

- Die Tasten repräsentieren die Basisoperationen auf den Datenstrukturen, die die Methoden des Algorithmus ausführen. Die Granularität kann an das Lernniveau der Benutzer angepasst werden.
- Der Aufbau der Tastatur sollte die Aufrufhierarchie des Algorithmus berücksichtigen und die Interaktion mit den Datenmodellen erleichtern.
- Spezielle Icons oder Earcons verbessern die Funktionalität der Tastenfelder und das Design der Tastatur.

Wir wollen kurz das System CONKAV vorstellen, das in zwei Bereiche unterteilt ist, die *Erzeugung und Konfiguration des CK* sowie die *Algorithmenvisualisierung (AV)*.

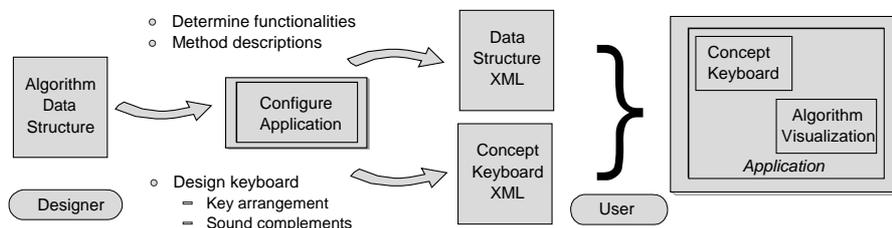


Abbildung 1: Arbeitsphasen mit CONKAV

4 Arbeiten mit CONKAV - Evaluierung

In einer ersten Phase, der Konfigurationsphase, handelt der Lerner als Designer, indem er zunächst den Algorithmus bestimmt und dann im Fenster ‚Configure CK‘ Methoden aus der Implementation wählt und durch Beschreibungen ergänzt, die es ihm erlauben, mit dem Algorithmus je nach Lernfortschritt eher grob- oder feinschrittig zu experimentieren (vgl. Abbildung 2). Wir nutzen ein vorgegebenes Schema in der Sprache Java, an das bestehende Implementationen leicht anzupassen sind. Aus den Eingaben erzeugt das System eine XML-Datei, die die Methoden mit ihren Ein- und Ausgabeparametern beschreibt und zur Visualisierung oder einer anderen Ausgabe genutzt wird. Dann entwirft der Lerner das CK. Dazu bestimmt er die Position der Tasten und verknüpft sie per ‚Drag and Drop‘ mit den Methoden. Die relative Positionierung der Tasten spiegelt die Datenstruktur wider und bietet Raum für weitere Usability-Untersuchungen. Die Beschriftungen der Tastenfelder können angepasst und verschiedene Tonausgaben zugeordnet, die bei Aktionen im Tastenfeld ausgelöst werden. Zusätzlich kann eine Initialisierungsdatei vorgesehen werden, die Startwerte für den Algorithmus und seine Visualisierung enthält und mit einem Editor für neue Aufgabenstellungen leicht modifizierbar ist. Schließlich werden die Eingaben in ein XML-File geschrieben, das das CK spezifiziert. Die Konfiguration kann natürlich auch vorweg vom Lehrer vorgenommen werden.

In einer zweiten Phase, der Visualisierungsphase, lädt der Lerner beide XML-Files sowie gegebenenfalls die Startdatei zum Arbeiten auf der Datenstruktur. Neben selbst entworfenen Tastaturen kann er auch ein einfaches Schrittinterface auswählen, das es erlaubt, den Algorithmus in der Standardabfolge auszuführen und zu visualisieren.

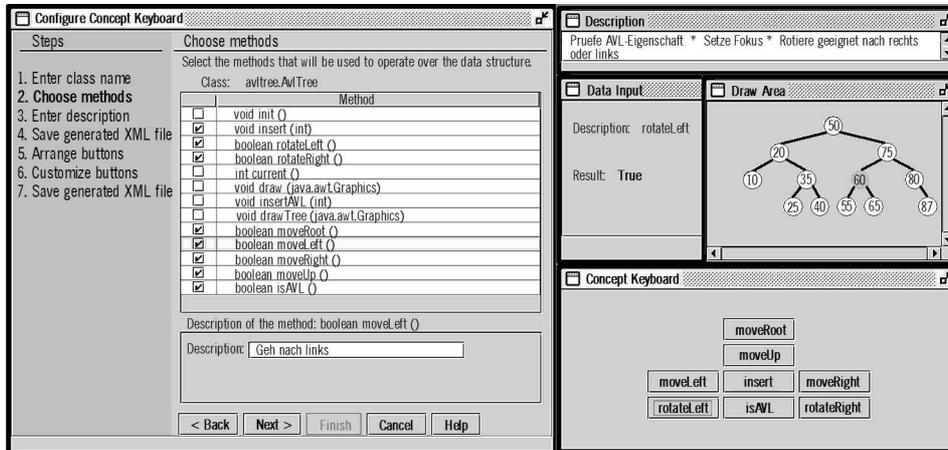


Abbildung 2: Blick auf die Arbeitsfenster von CONKAV

Nach vorbereitenden Befragungen an der Universität von Chile nahmen am 27.1.2004 18 Teilnehmer der Vorlesung ‚Algorithmen und Datenstrukturen‘ an einem Test in Duisburg teil. Nach einer Einführung in CONKAV lösten sie zunächst je eine Aufgabe zu den Algorithmen ‚Quicksort‘, ‚Minimaler Spannbaum‘ (Kruskal), ‚AVL-Baum‘ und ‚Kürzeste Wege‘ (Dijkstra). Freie Arbeit mit dem Schrittinterface für die ersten zwei und mit dem CK für die verbleibenden Algorithmen mündete in eine zweite Übungsphase. Die Studierenden ließen Präferenzen für das Schrittinterface bei Algorithmen erkennen, mit denen sie noch nicht vertraut waren. Ansonsten zogen sie das konfigurierbare CK vor. Die Befragung basierte auf der Taxonomie zum Lernerengagement bei der Nutzung von Visualisierungstechnologien von Naps et al. [Na03]. Die Autoren schlagen vor, Bewertungsmetriken zum System in einer hierarchischen Klassifizierung, darunter die Kategorien ‚Viewing‘, ‚Responding‘, ‚Changing‘ und ‚Constructing‘ zu entwickeln und zu nutzen. CONKAV ist dabei in den Kategorien ‚Sehen‘ über die Visualisierung, ‚Beantworten‘ über eine bewusste Auswahl einer Methode zur Erreichung des nächsten Teilziels, ‚Ändern‘ über Auswahl und Generierung neuer Beispiele, ‚Konstruieren‘ (allerdings nur in Bezug auf die Schnittstelle und nicht auf die Visualisierung) anzusiedeln.

Der nichtparametrische Mann-Whitney Test erlaubt Aussagen, ob die Verwendung der verschiedenen Steuerungen Einfluss auf die Lösung der Testaufgaben hat. Bei der Bearbeitung der Aufgaben zum Schrittinterface traten 7 Verschlechterungen bei 4 Verbesserungen, bei der Bearbeitung der Aufgaben mit dem CK 2 Verschlechterungen bei 8 Verbesserungen auf. Daher wurde die Hypothese, die Leistungsänderungen der Gruppe nach der Übungsphase seien unabhängig von der Wahl der Tastatur, auf dem 95% Signifikanzniveau zurückgewiesen, die Ergebnisse mit dem CK waren also signifikant besser. Dies steht in Einklang mit lernpsychologischen Erkenntnissen zum konstruktiven Lernen [MR98], dass die selbst gesteuerte Exploration und Interaktion gerade mit komplexem Lernstoff – wie unser System zur AV es erlaubt – nicht nur die Motivation der Lernenden erhöht, sondern auch die Verstehens- und Erinnerungsleistung zu steigern vermag.

5 Ausblick

Wir haben mit CONKAV neue Konzepte zur Verbesserung von Standardschnittstellen zur AA vorgestellt, die es erlauben, logische Repräsentationen der Datenstrukturen besser zu rezipieren und zu verstehen. Durch strikte Trennung von Kontrollelementen sowie Ein- und Ausgabe werden multimodale Interaktionen unterstützt. Wie erste Tests zeigen, wird der Fokus von der AV hin zu einem besseren Algorithmenverständnis verschoben. In Übungsphasen exploriert der Lerner die Datenstruktur und wird zu einer Reimplementation von Standardsoftware zur AV angeregt, um eine automatische Erzeugung und Verteilung der CKs über standardisierte Beschreibungen zu unterstützen. Das System wird in Präsenzveranstaltungen zur Einführung der Algorithmen unterrichtsbegleitend eingesetzt. Dabei wird die neue Campus-Infrastruktur für Wireless-Notebooks genutzt.

Literaturverzeichnis

- [Ba04] Baloian, N.; Hoppe, H.U.; Luther, W.; Motelet, O.: Implementing Teaching Strategies in the Classroom. Akzeptierter Beitrag bei der ED-MEDIA, Lugano, 2004.
- [BL01] Baloian, N.; Luther, W.: Visualization for the Mind's Eye. Workshop on Software Visualization, Dagstuhl, 20.–25.05.2001. In (Diehl, St. Hrsg.) Software Visualization, State-of-the-Art Survey LNCS 2269, Springer, Berlin 2002; S. 354–367.
- [BL03] Biella, D.; Luther, W.: Mobile verteilte Dokumentenrecherche in Bibliotheken und Archiven. In (Dittrich, K.; König, W.; Oberweis, A.; Rannenber, K.; Wahlster, W. Hrsg.): Informatik 2003 - Innovative Informatikanwendungen, Bd. 1, 2003; S. 298–302.
- [Cr02] Crescenzi, P.; Faltin, N.; Fleischer, R.; Hundhausen, Ch.; Näher, St.; Röbling, G.; Stasko, J.; Sutinen, E.: The Algorithm Animation Repository. Proceedings of the Second International Program Visualization Workshop, Århus, Denmark, 2002; S. 14-16.
- [Fa02] Faltin, N.: Strukturiertes aktives Lernen von Algorithmen mit interaktiven Visualisierungen. Dissertation. Abteilung Angewandte Informatik, Universität Oldenburg, 2002.
- [Hu99] Hundhausen, C. D.: Toward effective algorithm visualization artifacts: Designing for participation and communication in an undergraduate algorithms course. Unpublished Ph.D. thesis. University of Oregon, 1999.
- [Ja98] Janser, A.: Entwurf, Implementierung und Evaluierung des interaktiven Lehr- und Lernsystems VIACOBI für die Visualisierung von Algorithmen der Computergraphik und Bildverarbeitung. Logos Verlag, Berlin, 1998.
- [Lu00] Luther, W.: Algorithmus-Animation in Lehr- und Lernsystemen der Computergraphik. In (Diehl, St., Kerren, A. Hrsg.): GI-Workshop Softwarevisualisierung SV 2000. Schloss Dagstuhl; S. 103–114.
- [MR98] Mandl, H.; Reinmann-Rothmeier, G.: Auf dem Weg zu einer neuen Kultur des Lehrens und Lernens. In (Dörr G.; Jüngst, K. L. Hrsg.): Lernen mit Medien. Juventa-Verlag, Weinheim, 1998; S. 193-205.
- [Na03] Naps, T.; Röbling, G.; Almstrum, V.; Dann, W.; Fleischer, R.; Hundhausen, C.; Korhonen, A.; Malmi, L.; McNally, M.; Rodger, S.; Valazquez-Iturbide, J.A.: Exploring the Role of Visualization and Engagement in Computer Science Education. SIGCSE Bulletin inroads, Vol. 35, no 2, 2003; S. 131-152.
- [Rö02] Röbling, G.: Kommentierte Sammlung von Algorithmenanimationen und Animations-systemen. <http://www.animal.ahrgr.de/Anims/animations.php3>
- [Rö03] Röbling, G.: Key Decisions in Adopting Algorithm Animations for Teaching. In (van Weert, T.; Munro, K. R. Hrsg.) Informatics and the Digital Society. Kluwer Academic Publishers, Boston, Dordrecht, London, 2003; S. 149–156.