

GI, the Gesellschaft für Informatik, publishes this series in order

- to make available to a broad public recent findings in informatics (i.e. computer science and information systems)
- to document conferences that are organized in cooperation with GI and
- to publish the annual GI Award dissertation.

Broken down into the fields of "Seminars", "Proceedings", "Monographs" and "Dissertation Award", current topics are dealt with from the fields of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure the high level of the contributions.

The volumes are published in German or English.

Information: <http://www.gi-ev.de/LNI>

ISSN 1614-3213
ISBN 3-88579-436-5



Gesellschaft für Informatik (Hrsg.): Informatiktage 2005

GI-Edition

Lecture Notes in Informatics

Gesellschaft für Informatik (Hrsg.)

Informatiktage 2005

Fachwissenschaftlicher Informatik-Kongress

8. und 9. April 2005
Schloss Birlinghoven
Sankt Augustin





Gesellschaft für Informatik (Hrsg.)

Informatiktage 2005

Fachwissenschaftlicher Informatik-Kongress

8. und 9. April 2005

**Schloss Birlinghoven
St. Augustin bei Bonn**

Gesellschaft für Informatik 2005

Lecture Notes in Informatics (LNI) - Seminars

Series of the Gesellschaft für Informatik (GI)

Volume S-2

ISBN 3-88579-436-5

ISSN 1614-3213

Volume Editors

Gesellschaft für Informatik e.V.

Ahrstraße 45

53175 Bonn

E-Mail: gs@gi-ev.de

Redaktion: Ludger Porada

E-Mail: porada@gi-ev.de

Series Editorial Board

Heinrich C. Mayr, Universität Klagenfurt, Austria (Chairman, mayr@ifit.uni-klu.ac.at)

Jörg Becker, Universität Münster, Germany

Ulrich Furbach, Universität Koblenz, Germany

Axel Lehmann, Universität der Bundeswehr München, Germany

Peter Liggesmeyer, TU Kaiserslautern und Fraunhofer IESE, Germany

Ernst W. Mayr, Technische Universität München, Germany

Heinrich Müller, Universität Dortmund, Germany

Heinrich Reinermann, Hochschule für Verwaltungswissenschaften Speyer, Germany

Karl-Heinz Rödiger, Universität Bremen, Germany

Sigrid Schubert, Universität Siegen, Germany

Dissertations

Dorothea Wagner, Universität Karlsruhe, Germany

Seminars

Reinhard Wilhelm, Universität des Saarlandes, Germany

© Gesellschaft für Informatik, Bonn 2005

printed by Köllen Druck+Verlag GmbH, Bonn

Vorwort zum Tagungsband der Informatiktage 2005 in Schloss Birlinghoven

Sehr geehrte Damen und Herren, liebe Teilnehmerinnen und Teilnehmer,

wir freuen uns sehr, dass wir in Zusammenarbeit mit Ihnen in diesem Jahr nunmehr zum sechsten Mal die GI-Nachwuchsveranstaltung „Informatiktage“ realisieren können. Seit Beginn der Veranstaltungsreihe im Jahr 1999 haben sich sowohl das Umfeld der Informatikindustrie als auch Inhalte und Struktur der Informatikausbildung gewandelt. Die Gesellschaft für Informatik trägt diesen Veränderungen Rechnung, indem sie Ihnen in diesem Jahr ein neues Konzept der Informatiktage vorstellt.

Zum einen haben wir einen neuen Ort gewählt: Schloss Birlinghoven in Sankt Augustin bei Bonn ist Sitz des Fraunhofer Instituts für Angewandte Informationstechnik (FIT). Wobei „angewandt“ hier das richtige Stichwort ist: In diesem Jahr stellen Studierende zum ersten Mal ihre Studienprojekte vor und demonstrieren damit deutlich stärker als in der Vergangenheit ihren Praxisbezug. Die GI trägt somit aktiv den Ansprüchen der Unternehmen Rechnung, praxisrelevante Studieninhalte zu fordern und zu fördern.

Die in diesem Band vorliegenden Beiträge von Studienprojekten wurden aus einer Vielzahl von Einreichungen ausgewählt; sie stellen also aus unserer Sicht das Beste dar, was derzeit an deutschen Hochschulen gelehrt und gelernt wird. In diesem Zusammenhang danke ich ausdrücklich den Vertrauensdozentinnen und Vertrauensdozenten, die ihre besten Studierenden vorgeschlagen haben, sowie dem Auswahlkomitee, welches die vielen Beiträge gesichtet und klassifiziert hat. Ebenfalls danke ich unserem Gastgeber, dem FIT, und der Firma sd&m, die durch ihre Unterstützung zum Erfolg unserer Tagung beigetragen haben. Ich gratuliere unseren Studierenden zu ihren hervorragenden Arbeiten und bin sicher, dass die Gesellschaft für Informatik auch in Zukunft mit ihrer Nachwuchsförderung eine erfolgreiche Personalpolitik für morgen betreiben wird.

Ihnen und uns wünsche ich spannende Tage in Schloss Birlinghoven und interessante Erkenntnisse bei der Lektüre dieses Tagungsbandes.



Dr. Peter Federer
(Geschäftsführer der Gesellschaft für Informatik e.V.)

Inhaltsverzeichnis¹:

Übersicht über das Rahmenprogramm und Partner der Informatiktage 2005	11
----------------------------------------------------------------------------------------	----

Modelle und Werkzeuge für Softwareentwicklung

Gepting A., Greenyer J., Maas A., Munkelt S., Pales C., Pivl T., Rohe O., Sanders M., Scholand, A., Wagner C. ComponentTools	13
--------------------------------------------------------------------------------------------------------------------------------------------------	----

Jubeh R. Das FUJABA Window Toolkit	17
----------------------------------------------------	----

Winter S. Generierung von dynamischen Web-Anwendungen aus visuellen Spezifikationen	21
-----------------------------------------------------------------------------------------------------	----

Riegg T. Analyse der Anwendbarkeit des V-Modell XT auf kleine IT-Vorhaben	25
-------------------------------------------------------------------------------------------	----

Mergenthaler M., Schneider J. Code Analyse Plugin (für Eclipse)	29
---------------------------------------------------------------------------------	----

Entwurfsmethoden

Volz B. Ein Speicherprofiler für Enterprise Java Beans	33
------------------------------------------------------------------------	----

Monger A., Hofmann S., Bronni J., Kronfeld M. Eine strategierorientierte, modulare Simulationsumgebung für mobile Ad-Hoc-Szenarien	37
-------------------------------------------------------------------------------------------------------------------------------------------------------	----

König C., Hofmann A., Konradi P., Ecker M. Software Engineering für Embedded Systems Innovative Anwendungen	41
-----------------------------------------------------------------------------------------------------------------------------	----

¹ Die Beiträge sind thematisch zugeordnet und folgen der Reihenfolge der Vorträge im Rahmen der Veranstaltung

Innovative Informatikanwendungen

Reimers S., Sonnenberg C., Schilz S.

BOGSY – ein Informationssystem für Botanische Gärten 45

Langhammer T., Lambeck S., Opitz L.

Implementierung von Dosis-Wirkungs-Modellen der Radiotherapie 49

Eggendorfer T.

Möglichkeiten der Prävention der Zusendung unerwünschter Werbe-E-Mails 55

Behrndt J., Brandt C., Homann J., Keuch C., Stempell I.

Studienprojekt Hafenlogistik: Erstellung einer verteilten Simulationsumgebung für ein teilautomatisiertes Containerterminal 59

Wolters R.

Controlling im Hilfsmittelbereich 63

Verteilte Systeme und Architekturen

Heer T.

Einsatz Verteilter Hash-Tabellen in mobilen Netzen 73

Schäfer A.

Harpy – Employing a Process Migration facility and a User Mode Linux for High Throughput Computing 77

Komann M.

Implementing a framework for optimising parameter studies on cluster computing environments 81

Gerdemann S., Fritzsche C.

P2P Netzwerke für dynamische 3d Szenen 85

Birkenheuer G., Hagelweide W., Hagemeyer B., Japs V., Keller M.,

Mayr N., Meyer J., Schumacher T., Voß K., Zajac M.

PIRANHA – Hunter of Idle Resources 91

Theoretische Informatik und Künstliche Intelligenz

Grathwohl P., Rögner J., Fritsch A., Kuhlmann M.

Regelbasierte Analyse und Komposition von Musik 95

Müller C.

Implementierung von Default-Regeln durch optimale konditionale
Rangfunktionen 101

Völker M.

Zeichnen von Begrenzungen ungenau definierter Gebiete 105

E-Learning und Informationssysteme

Metzen J.H., Ploss A., Schellmann M.

Rokkatan – ein massiv mehrspielerfähiges Echtzeitspiel 109

Stiller M.

E-Learning und Datenbanken 113

Weise T.

Entwicklung eines WYSIWYG Editors für das Erstellen von Lehrmaterial
im XML Format 117

Jahnke T., Köstlbacher A., Sporer T.

Softwareframework für audiovisuelle Wissensmedien im Internet 121

Hinzen M., Meesters S., Müller T., Niemeyer G., Robert S.

Konzeption und Entwicklung von O.S.I.R.I.S. – Open source basiertes
integriertes regionales Informationssystem für Virtuelle Rathäuser 125

Fieber F.

Mobiler Client für die E-Learning Plattform „CLIX „ 129

Datenbanken

Brüntje O., Glüpker A., Mertens F., Wolke K.

JDBC Datenbank Beobachter 133

Schulte-Coerne T.

Visualisierung von Distanzfunktionen insbesondere der Earth Movers Distance 139

Mauser A., Bezrukov I., Deselaers T., Keysers D.
Predicting Customer Behavior using Naive Bayes and Maximum Entrop 143

Graphische Datenverarbeitung

Ullrich S.
Interaktive Muskelmodellierung für virtuelle Umgebungen 147

Bastuck C., Hambuerger T., Hofmann T., Mehnert J., Nowak S.
An Open and Extensible Framework for Visualization 151

Rybach D., Keysers D., Ney H.
Erweiterung eines holistischen statistischen Bilderkenners zur Verwendung
von mehreren Merkmalen 155

Förder D., Papazov C.
Simulation dynamischer Prozesse und Kollisionserkennung im
Dreidimensionalen mit einfachen geometrischen Objekten 159

Sandip S.-D.
Normalization and Parameterization of Motion-Captured Data 163

Übersicht über das Rahmenprogramm und Partner der Informatiktage 2005

Neben den Vorträgen und Präsentationen der Studierenden wurde den Teilnehmerinnen und Teilnehmern ein attraktives Rahmenprogramm geboten. Wir bedanken uns außerdem bei unseren Partnern, ohne deren Unterstützung die Veranstaltung nicht möglich wäre.

Rahmenprogramm:

8. April, 9:00 Uhr:

Begrüßung und Einführung: **Prof. Dr. Gottfried Vossen**, Universität Münster, Vizepräsident der GI

9:15 Uhr:

Prof. Dr. Matthias Jarke, Präsident der GI und Leiter des FIT-Institut für Angewandte Informationstechnik:
"Mensch-zentriertes Computing"

16:15 Uhr

Dr. Bernhard Humm, Leiter sd&m Research:

"Integration großer IT-Systeme in der Praxis"

Dipl. Wirt.-Inf. Kai Oey, Universität Köln und AK "Service oriented Architecture (WI-SOA)"

Podiumsgespräch mit den Referenten

Abendveranstaltung:

Dinner-Speech: **Prof. Dr. Gunter Dueck**, IBM Distinguished Engineer, Mitglied des GI-Präsidiums: "Informatiker in artgerechter Haltung"

9. April, 9:00 Uhr

Keynote-Vortrag: **Prof. Dr. Klaus Brunnstein**, Universität Hamburg:

"Die Bedeutung der IT-Sicherheit für die Informationsgesellschaft"

Moderation: Cornelia Winter, Stellvertreterin des Geschäftsführers

Partner der Informatiktage:

Hauptsponsoren:

sd&m

www.sdm.de

FIT-Fraunhofer

www.fit.fraunhofer.de

Partner der Recruiting-Veranstaltung:

CareerVenture von MSW & Partner

www.career-venture.de

Medienpartner:

Computer Zeitung

www.computerzeitung.de

ComponentTools*

Alexander Gepting, Joel Greenyer, Andreas Maas, Sebastian Munkelt, Csaba Pales,
Thorsten Pivl, Oliver Rohe, Markus Sanders, Andreas Scholand, and Christian Wagner

pg-components@upb.de

Supervisors: Ekkart Kindler, Vladimir Rubin, Robert Wagner

University of Paderborn

Abstract: Many techniques from the field of formal methods, e.g. Petri Nets, State Charts or timed automata, are powerful for modelling, analysing and verifying dynamic aspects of all kinds of systems. All these techniques have their particular strengths and weaknesses, which often make it necessary to have models in different notations for the very same system. It might be necessary to have one abstract model and another more detailed model of the same system for performing different analysis tasks. The problem, however, is that the different techniques come with their own composition mechanisms, which make it very expensive to use and to maintain different models of the same system. In this paper, we give a brief overview of the concepts of tool called COMPONENTTOOLS that will help to solve all these problems.

1 Introduction

The COMPONENTTOOLS Project is developed by a group of ten students in two semesters course during the master program at the University of Paderborn. This tool allows us to easily apply formal methods in the development cycle of component based systems. The main objective is to help engineers to design component based manufacturing systems without having to explicitly deal with the different formal methods involved.

Many mechanical systems used today consist of standard hardware components and therefore we want to provide an infrastructure in which the identified components and their connections can be modelled on a high level of abstraction. Behind this abstract formalism to describe a component based system, more detailed formal or structural models can be included, hidden from the user. These are used to support further design tasks, such as verification, validation, or simulation and visualization of the system.

We introduce our concepts using a simple example of a toy train system which can be seen in Fig. 1. But the concepts a very general and can be applied to any component based

*This paper is based on [AG04]

system. The toy train system consists of four different components: a straight track, a curved track, a track with a stop signal and a switch. In order to design toy train systems, these component types and possible connections between them have to be defined in a *component library*. Along with the definition of a component type, one or more models can be included to describe its behaviour or structure. Using this library, an engineer can use the COMPONENTTOOLS editor to model a specific system, which we call a *project*. COMPONENTTOOLS also provides mechanisms to generate an overall model for every formal or structural model included in the component type definition.

Related tools have the disadvantage that they often work on fixed model types which are interpreted by fixed mechanisms and further models cannot be integrated. In COMPONENTTOOLS, any kind of model can be integrated and we provide a general mechanism to transform these models based on triple graph grammars (TGGs) [Sch95]. A transformed overall model can be, for instance, a Petri net and can be used by further applications. In case of the toy train example, we can use a model checker like NuSMV [CCGR] or PNVis [EK04], a program which visualizes Petri nets in a 3D simulation of the toy train system.

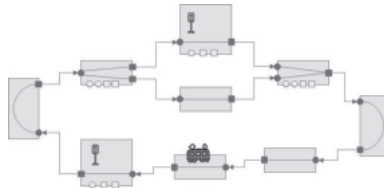


Figure 1: Model of a simple toy train

2 Concepts

This section introduces the main concepts and parts of COMPONENTTOOLS infrastructure.

Component Library The *component library* contains definitions of components that were identified and analysed for a particular system. In each component type definition one or more formal or structural models can be included. Furthermore, in mechanical systems like in our toy train example, the components can be interconnected in certain ways, for example mechanically or by electrical signal wires. Therefore, the component library defines *ports*, the plugs that can exist in the interface of a component type. The definition of these ports and the possible connections between them is called a *connection paradigm*.

To use the component library in a graphical editor, information about the graphical appearance can be stored in the library. It is also possible to include different views on the same type of system for certain users or certain tasks in the engineering process.

Model Generation and Transformation To generate an overall model from a created component system, we use triple graph grammars(TGG). The advantage of triple graph grammars is that they not only describe a graph translation, but also keep track of how nodes of a source graph correspond to nodes of the target graph. This provides the flexibility to use productions for both forward and backward transformation and correspondence analysis of a model. A triple graph grammar, as a declarative definition of the mapping between the two graphs, can be used in the implementation of a translator or interpreter in either direction. The TGG Interpreter integrated in COMPONENTTOOLS is able to transform to and from any particular graph model. To obtain this flexibility, a model adapter has to be implemented for each specific graph model.

Once an overall model has been generated from a component project, further applications can perform different tasks on this model. And also, because we keep track of the component project and model graph relationships, results from these applications can be transformed back into the component project.

Editor As mentioned before, we want to have different views on the same system. These views support different design tasks or may be useful at different stages in the development process.

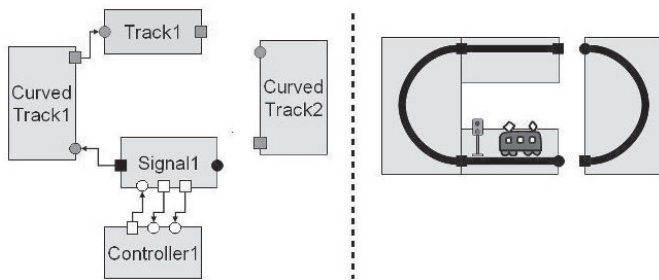


Figure 2: Two example views

For the toy train, for example, there could be one view showing the electrical signal connections and another more physical view for the traffic connections, which shows the specific position of the components. This is illustrated in Fig. 2.

3 Conclusion

COMPONENTTOOLS offers general techniques to support the design process of component based systems. It is possible to utilize various formal models that support further design tasks, such as validation, verification or simulation of the system. The generation of these overall models from a component project is performed using triple graph grammars, which will also make it possible to translate analysis results back into the component project.

References

- [AG04] E. Kindler A. Maas S. Munkelt C. Pales T. Pivl O. Rohe V. Rubin M.Sander A. Scholand C. Wagner R. Wagner A. Gepting, J. Greenyer. Component Tools: A vision for a tool, 2004. E. Kindler (ed.): Algorithmen und Werkzeuge für Petrinetze (AWPN) - Algorithms and Tools for Petri nets. Proceedings of the Workshop AWPN 2004. pp. 37-42.
- [CCGR] Alessandro Cimatti, Edmund Clarke, Fausto Giunchiglia, and Marco Roveri. NUSMV: a new symbolic model checker.
- [EK04] Csaba Páles Ekkart Kindler. *3D-Visualization of Petri Net Models: Concept and Realization*, volume pplications and Theory of Petri Nets 2004: 25th International Conference, ICATPN 2004, Bologna, Italy, June 2125, 2004. Proceedings. Springer, 2004.
- [FUJ04] FUJABA Days 2004 Proceedings. *An adaptable TGG interpreter for in-memory model transformation*, September 2004.
- [Gre03] Joel Greenyer. Maintaining and using component libraries for the design of material flow systems: Concept and prototypical implementation, 2003.
- [Har87] David Harel. *Statecharts: A visual formalism for computer systems*. *Science of Computer Programming*. June 1987.
- [nus] NuSMV: a new symbolic model checker. <http://nusmv.irst.itc.it/NuSMV>.
- [Sch95] Andy Schürr. *Specification of graph translators with triple graph grammars*. In Ernst W. Mayr, Gunther Schmidt, and Gottfried Tinhofer, editors, *Graph-Theoretic Concepts in Computer Science, 20th International Workshop, WG 94, Herrsching, Germany, Proceedings, Lecture Notes in Computer Science 903, pages 151163*. 1995.

Das FUJABA Window Toolkit

Ruben Jubeh

University of Kassel
Research Group Software Engineering
Department of Computer Science and Electrical Engineering
Wilhelmshöher Allee 73
34121 Kassel
rjubeh@gmx.de

Abstract: Das FUJABA Window Toolkit (FWT) ermöglicht die Erstellung von grafischen Anwendungen ohne Quelltext-Programmierung. Dies geschieht durch die Integration in FUJABA, mit dem Anwendungslogik mittels UML modelliert werden kann. FWT kann anschließend interaktiv grafische Elemente wie Buttons oder Eingabefelder erzeugen und diese direkt an die modellierte Anwendung binden. Diese Bindung verteilt Änderungen in beide Richtungen automatisch, so dass Oberfläche und Logik immer synchron sind. Schließlich entsteht ein ausführbares Programm, das auch ohne FUJABA läuft.

1 Motivation

Mit klassischen GUI Designern, z.B. das Eclipse Visual Editor Project [VEP01], kann man durch ein einfaches Baukasten-System interaktiv eine grafische Oberfläche erstellen. Die Möglichkeiten, die das zu Grunde liegende Window Toolkit, wie z.B. Java Swing, bietet, werden weitgehend abgedeckt. GUI Designer erzeugen dann eine Beschreibung der modellierten Oberfläche, z.B. als Java Quellcode oder XML-Beschreibung. Problematisch ist allerdings die Anbindung der Programm-Logik. Diese entsteht in der Regel unabhängig von der grafischen Oberfläche (GUI). Die Verbindung zwischen Logik und GUI muss dann vom Entwickler in Handarbeit erledigt werden: Es ist dafür zu sorgen, dass Änderungen in der Logik auch in der GUI angezeigt werden. Dazu muss die GUI an das Datenmodell der Logik adaptiert werden. Weiterhin müssen Benutzer-Eingaben wie Maus-Klicks oder Tastatur-Eingaben zur Logik weitergereicht werden. In der Regel müssen hierzu Ereignis-Methoden programmiert werden, die dann die Eingabe prüfen und an die Logik weiterreichen. Besondere Herausforderungen sind komplett dynamische Oberflächen, die ihre Elemente je nach Zustand der Logik ändern. Die Anpassungen im Quelltext sind mühsam, fehleranfällig und erfordern detailliertes Fachwissen. Das Konzept von Fujaba und FWT sieht dagegen vor, ohne Quelltext-Programmierung, sondern 'grafisch' zu programmieren.

Die Anwendung entsteht dabei in Fujaba durch UML-Modellierung, die GUI mittels FWT durch Aufbauen von Objekt-Strukturen. FWT hat zum Ziel, auch für Personen mit geringer Programmiererfahrung geeignet zu sein, sowie für die Lehre (siehe [DGZ02]), denn es vermittelt den objektorientierten Aufbau der GUI, nimmt dem Benutzer den Programmieraufwand ab und ermöglicht es, schnell zu brauchbaren Ergebnissen zu gelangen.

2 Idee und Beispiel

Fujaba [Fuj98] ist ein Open Source UML CASE Tool, implementiert in Java. Fujaba erzeugt aus UML-Diagrammen vollständigen Java Quellcode ([Zün01]), der dann transparent kompiliert und ausgeführt wird. Die Ausführung wird ebenfalls grafisch im Objekt-Browser DOBS (Dynamic Object Browsing System) präsentiert. Dort können Objekte zur Laufzeit erzeugt werden und werden grafisch angezeigt. DOBS erlaubt die in Fujaba entworfenen Klassen und Methoden oder interaktiv auszuprobieren oder zu testen.

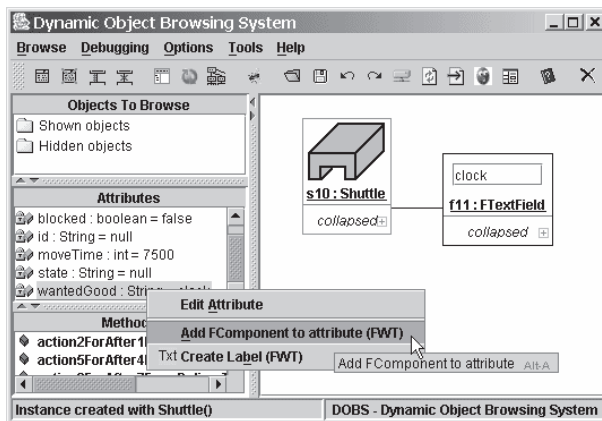


Abbildung 1: Instanz eines Shuttles im DOBS sowie eine FWT Textfeld-Komponente, gebunden auf ein Attribut des Shuttles

Der DOBS kann jedes Java-Objekt, das sich im Heap, also dem Java Speicherbereich befindet, anzeigen. Zu jedem ausgewählten Objekt können Methoden aufgerufen werden, sowie die Attribute geändert werden. Bestehen zwischen Objekten Links, zeigt DOBS diese als grafische Verbindungslinie an (siehe Abbildung 1). Das Arbeiten mit FWT geschieht durch interaktives Aufbauen von Objektstrukturen im DOBS. Von der Handhabung ist FWT vergleichbar mit anderen GUI Designern. Über die Werkzeug-Leiste können FWT Komponenten wie Buttons, Eingabefelder für Text und Zahlen, Auswahl-Listen, Layouts, usw. erzeugt werden. Durch den DOBS kann FWT mit Instanzen von realen GUI-Komponenten arbeiten, die in den DOBS eingebettet werden. FWT Komponenten können durch einfaches Kantenziehen hierarchisch miteinander verbunden werden. So entsteht z.B. ein FWT Fenster, das mehrere Buttons und Listen enthält, die in einem bestimmten

Layout angeordnet sind. Sind die Logik-Objekte ebenfalls im DOBS instanziiert, können die FWT Komponenten einfach an die Logik gebunden werden. Diese Logik-Bindung ist bidirektional, d.h. das Framework sorgt automatisch dafür, dass Änderungen in der Logik sofort an die GUI weitergereicht werden, und Benutzereingaben werden ebenfalls sofort an die Logik übermittelt. Diese Automatik wird mit CoObRA [Schn03] realisiert. Abbildung 1 zeigt eine solche Bindung: Das Attribut *wantedGood* des Shuttles ist dem Textfeld zugeordnet. Eingaben in das Textfeld werden unmittelbar zum Shuttle Objekt übertragen. Änderungen des Attributes im DOBS sind ebenfalls sofort im Textfeld sichtbar. Solche Bindungen sind auch für andere Datentypen möglich.

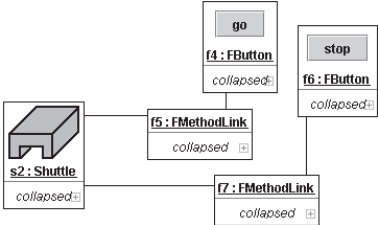


Abbildung 2: Methodenaufrufe durch FWT Buttons

Abbildung 2 zeigt die Bindung von FWT Buttons an Methoden der Logik. Durch Klick auf die Buttons wird die gleichnamige Methode des Shuttles aufgerufen. Auch Methodenaufrufe mit Parametern sind durch die Kombination von Methoden- und Attributbindung möglich.

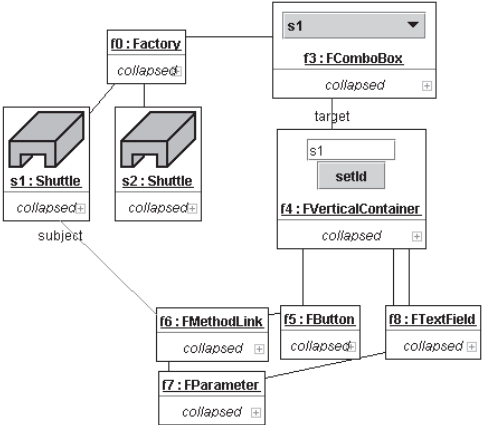


Abbildung 3: Auswahl-Box, gebunden auf eine Assoziation, und dynamische Bindung

Abbildung 3 zeigt eine dynamische Funktion von FWT: Das Factory-Objekt verwaltet eine Menge von Shuttles. Die darauf gebundene FWT Auswahl-Box f3 zeigt die Liste der Shuttles an. Diese passt sich automatisch an, wenn sich die Anzahl der Shuttles ändert.

Die FWT Komponenten im Objekt *f4* sind nun nicht an ein bestimmtes Shuttle gebunden, sondern die Auswahl in *f3* bestimmt die Bindung. Je nach Auswahl zeigt die *subject*-Kante entweder zu *s1* oder *s2*. So lassen sich einfach eine variable Anzahl von Logik-Objekten per GUI steuern.

Die Objekt-Modellierung im DOBS, die sowohl aus Logik- als auch aus FWT-Objekten besteht, lässt sich jederzeit abspeichern. Die Ausführung der Anwendung kann anschließend auch ohne Fujaba und DOBS erfolgen.

3 Zusammenfassung und Ausblick

Schon während der Entwicklungsphase wurden Teile der FWT Funktionalität in anderen Projekten produktiv eingesetzt. Es war z.B. möglich, zu einem in Fujaba modellierten Mensch-Ärgere-Dich-nicht-Spiel eine voll funktionsfähige Oberfläche zu erstellen, so dass das Spiel als eigenständige Anwendung ablaufen konnte. Besonders hervorzuheben ist, dass FWT im Gegensatz zu den klassischen Vorgehen komplett zur Anwendungs-Laufzeit, also an und mit lebendigen Objekten, arbeitet. Das Ziel von Fujaba, ohne klassische Programmierung zur lauffähigen Anwendung zu gelangen, wurde durchgehalten. FWT ist damit besonders in der Lehre bzw. für Anfänger sehr leicht einsetzbar. Es hat sich gezeigt, dass der aktuelle Funktionsumfang für viele in Fujaba entworfene Programme schon ausreicht. FWT ist aber noch nicht in der Lage, für die Weiterentwicklung des DOBS oder Fujaba selbst hilfreich zu sein.

Literatur

- [DGZ02] I. Diethelm, L. Geiger, A. Zündorf: UML im Unterricht: Systematische objektorientierte Problemlösung mit Hilfe von Szenarien am Beispiel der Türme von Hanoi, Paper auf dem ersten Workshop der GI-Fachgruppe Didaktik der Informatik, Bommerholz, 2002
- [DGMZ02] I. Diethelm, L. Geiger, T. Maier, A. Zündorf: Turning Collaboration Diagram Strips into Storycharts; Workshop on Scenarios and state machines: models, algorithms, and tools; ICSE 2002, Orlando, Florida, USA (2002)
- [GZ02] Leif Geiger, Albert Zündorf: Design Level Debugging with Fujaba, Paper auf dem International Workshop on Graph-Based Tools (GraBaTs), Barcelona, 2002
- [Fuj98] Projektgruppe Fujaba: Homepage, Universität-Gesamthochschule Paderborn, 1998, <http://www.fujaba.de/>
- [Schn03] C. Schneider: CASE Tool Unterstützung für die Delta-basierte Replikation und Versionierung komplexer Objektstrukturen; Diploma Thesis, Corolo Wilhelmina zu Braunschweig, Braunschweig, Germany, 2003
- [VEP01] The Eclipse Visual Editor Project: Homepage, Eclipse Foundation, 2001, <http://www.eclipse.org/vep/>
- [Zün01] Albert Zündorf: Rigorous Object Oriented Software Development; Habilitation Thesis, University of Paderborn (2001) (1996)

Generierung von dynamischen Web-Anwendungen aus visuellen Spezifikationen

Stephan Winter für die Projektgruppe PaderWAVE

Universität Paderborn
Fakultät Elektrotechnik, Mathematik und Informatik
swinter@upb.de

Art der Arbeit: Projektgruppe

Betreuer/in der Arbeit: Dipl.-Inf. C. Schmidt, Dr. M. Thies, Prof. Dr. U. Kastens

GI-Fachbereich: Softwaretechnik

Abstract: PaderWAVE [1] ist ein visueller Editor zur Generierung von dynamischen Web-Anwendungen. Er überprüft Spezifikationen auf Konsistenz und generiert daraus automatisch vollständige Webanwendungen bestehend aus Webseiten, Navigation, Links und Datenbankoperationen. Dynamische Inhalte lassen sich über Formulare und eine eigens entwickelte visuelle Abfragesprache für Datenbanken verarbeiten. Das Layout sowie die Positionierung der Webseitenelemente werden über Stylesheets gelöst. Darüber hinaus steht es dem Nutzer frei, eigene Skripte in die Webanwendung zu integrieren.

1 Übersicht

Die PG PaderWAVE hat eine visuelle Entwicklungsumgebung realisiert, mit der Web-Designer Anwendungen für das Web visuell modellieren können, um danach den erforderlichen Anwendungs-Code ohne eigenes Zutun von der Umgebung generieren zu lassen. Der Anwendungs-Code der generierten Webanwendung basiert auf HTML mit Anteilen an PHP [4]-Skripten und SQL-Anweisungen.

Die Entwicklung von Webanwendungen mit Hilfe von PaderWAVE bietet den Vorteil, dass robuster und konsistenter Code erzeugt wird. Die Arbeit an einem graphischen Modell ist zudem unkomplizierter und erhöht das Verständnis für die Abläufe der Webanwendung. Auch Nicht-Experten können mit dem Editor eine Webanwendung erstellen ohne Bearbeitung des Quellcodes.

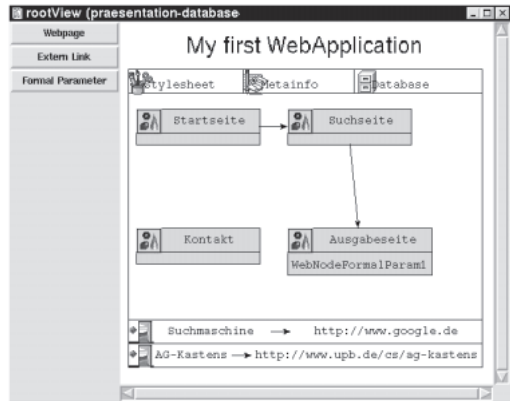
2 PaderWAVE Sprachkonstrukte

Zur übersichtlichen Trennung verschiedener Detailsstufen einer Webanwendung unterstützt der PaderWAVE Editor verschiedene Sichten. Die wichtigsten sind die Webanwendungssicht, die Webseitensicht und die Datenbanksicht.

2.1 Webanwendungssicht

Die Webanwendungssicht dient zur Spezifikation aller benötigten Seiten (ohne Inhalte) sowie der Navigationsstruktur zwischen diesen Seiten und gestattet Zugriff auf untergeordnete Sichten. Neben internen Links zwischen Seiten der Webanwendung können auch Links auf externe Seiten definiert werden.

Der PaderWAVE Editor unterstützt auch eine Einteilung der Webseiten in Zonen. Zonen unterteilen Webanwendungen in Bereiche, die für verschiedene Benutzergruppen (mit verschiedenen Rechten) zugänglich sind. Zonen sind schachtelbar und bei Eintritt in eine Zone wird ein bisher unauthentifizierter Benutzer automatisch auf eine Login-Seite weitergeleitet.

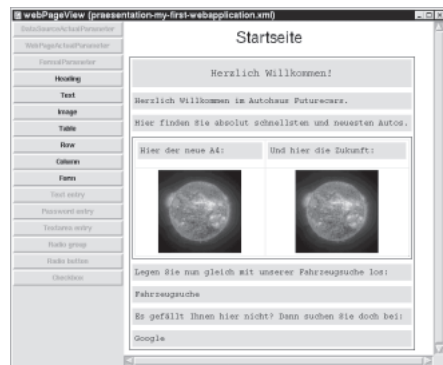


Neben Webseiten unterstützt der Editor auch WebActions, ein Konstrukt zur Ausführung von Datenbankoperationen. Diese können zum Beispiel die Rechte eines Benutzers prüfen und ggf. auf eine Fehlerseite umleiten.

2.2 Webseitensicht

Eine detailliertere Sicht auf eine Webseite bietet die Webseitensicht. In der Webseitensicht kann eine vollständige Seite modelliert werden. Zu den Komponenten, die eine WebPage enthalten kann gehören Überschriften, Tabellen, Texte, Bilder, Formulare und Ausgabeelemente.

Ein Kernelement für eine dynamische Webanwendung ist das Formular, welches es erlaubt, Werte an eine andere Webseite oder WebAction zu übermitteln. Formularen können Konsistenzprüfungen und Initialwerte aus der Datenbank zugeordnet werden. Beim Absenden von Formulardaten werden diese als aktuelle Parameter an andere Webseiten oder WebActions



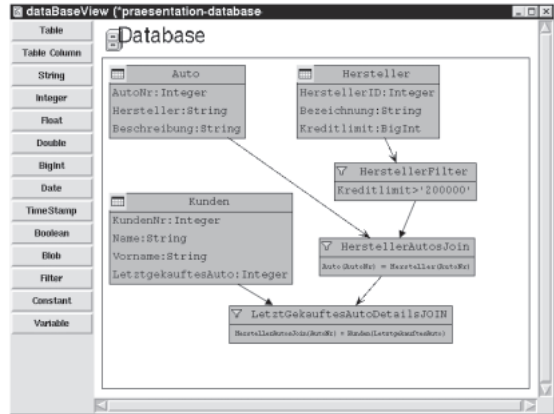
geschickt. Welche Webseite aufgerufen wird, wird in der Webanwendungssicht durch einen Pfeil zwischen den Seiten verdeutlicht.

Damit gleiche Teile einer Webseite (insbesondere z.B. Formulare) nicht auf jeder Seite neu entwickelt werden müssen, können wieder verwendbare Teile in sogenannten Frames abgelegt werden, welche auf mehreren Webseiten verwendet werden können.

2.3 Datenbanksicht

Zur persistenten Speicherung von Daten können in der Datenbanksicht Datenbanktabellen und Abfragen spezifiziert werden.

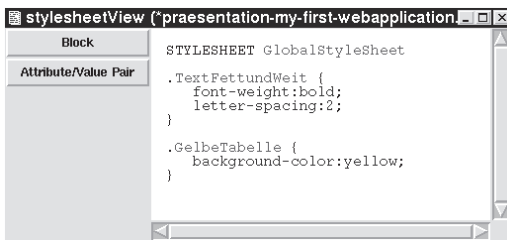
Zur Abfrage der Daten einer Datenbank wird ein Filterkonzept genutzt. Es können ausgehend von einer Datenquelle Filter definiert werden, die wieder neue Datenquellen darstellen. Diese Filter erlauben Operationen wie Projektion, Join, Gruppierung, Selektion und Begrenzung. Zudem gibt es auch Operationen, die zur Datenmanipulation genutzt werden.



Filter sind grundsätzlich kombinierbar, sodass mehrere Filter hintereinander zu so genannten Filterketten eingesetzt werden können. Ebenso wie Webseiten können auch Datenbankfilter formale Parameter besitzen, die beim Aufruf des Filters im Kontext eines Formulars oder eines Ausgabelements an aktuelle Parameter gebunden werden.

2.4 Weitere Sichten

Die Navigationsleistsicht spezifiziert eine Navigationsleiste, die automatisch in alle Webseiten integriert wird. Der Web-Designer definiert, welche Seiten in welcher Reihenfolge in der Navigationsleiste enthalten sind. Zur Laufzeit werden Seiten, auf die der Benutzer keinen Zugriff hat automatisch ausgeblendet. Das Design und die Anordnung der Navigationsleiste können frei angepasst werden.



Zur Positionierung und Formatierung von Elementen einer Webseite können in der Stylesheet-Sicht Stylesheetblöcke definiert werden. Diese können den Elementen von Webseiten zugeordnet werden.

Allgemeine Informationen zu der Webanwendung werden üblicherweise

in den Seiten als Metainformationen angegeben. Hierfür existiert im PaderWAVE Editor eine eigene Sicht, die dies erlaubt.

3 Umsetzung

Der PaderWAVE Editor wird mit dem System DEViL (Development Environment for Visual Languages) erzeugt. DEViL ist ein Generator, der Entwicklungsumgebungen für visuelle Sprachen aus Spezifikationen generiert. Die Spezifikationen beschreiben die abstrakte Struktur, visuelle Darstellung und Übersetzung in eine Zielrepräsentation.

Der PaderWAVE Editor nutzt als Ausgabeformat HTML. Dynamische Webseiten werden mit PHP erzeugt. Der Datentransport über mehrere Webseiten hinweg wird mit Hilfe von PHP-Sessions gelöst. Als Datenbankmanagementsystem wird MySQL verwendet.

Der Editor bietet neben der Nutzung der Datenbank auch die Erstellung der Initialisierungsskripte für die Datenbank.

Um robusten und fehlerfreien Code zu generieren verfügt der Editor über eine Reihe von Konsistenzprüfungen, die die Fehler in der modellierten Anwendung aufdecken.

Literaturverzeichnis

- [1] PaderWAVE Webseite: <http://ag-kastens.upb.de/lehre/paderwave/>
- [2] DEViL Webseite: <http://ag-kastens.uni-paderborn.de/forschung/devil/index.php>
- [3] ELI Webseite: <http://ag-kastens.uni-paderborn.de/forschung/eli.php>
- [4] PHP Webseite: <http://www.php.net>
- [5] Cascading Stylesheets: <http://www.w3.org/Style/CSS/>
- [6] W3 Consortium: <http://www.w3.org/>

Analyse der Anwendbarkeit des V-Modell XT auf kleine IT-Vorhaben

Tobias Riegg
Berufakademie Heidenheim
Studiengang Wirtschaftsinformatik
triegg@wiec.de

Art der Arbeit: Praxisprojekt
Vertrauensdozent: Prof. Dr. Hans Jürgen Ott, Berufsakademie Heidenheim

Abstract: Diese Projektarbeit analysierte, wie sich die Fortschreibung des V-Modells auf die Anwendbarkeit bei kleinen IT-Vorhaben ausgewirkt hat. Durch die Aktualisierung des Standards konnte eine höhere Akzeptanz der (potentiellen) Benutzer erreicht werden. Durch eine neue Grundstruktur, die eine höhere Flexibilität in der Projektdurchführung erlaubt, wurde die Anwendbarkeit deutlich erhöht.

1 Vorgehensmodelle in der Softwareentwicklung

Jede Softwareerstellung sollte in einem geregelten organisatorischen Rahmen ablaufen. Um dies zu gewährleisten existieren verschiedene Vorgehensmodelle für die Softwareentwicklung, um gerade diesen Rahmen festzulegen. In solchen Modellen wird die Reihenfolge des Arbeitsablaufs mit den einzelnen durchzuführenden Aktivitäten beschrieben und festgelegt. Zudem werden neben Fertigstellungskriterien anzuwendende Standards, Richtlinien, Methoden und Werkzeuge vorgegeben, um einen geregelten Verlauf des Projektes zu realisieren [vgl. BAL S.98].

2 Geschichte des V-Modell

1986 beauftragte das Bundesministerium für Verteidigung (BMVg) das Bundesamt für Wehrtechnik und Beschaffung (BWB) Maßnahmen zu ergreifen, um den Softwareentwicklungsprozess wesentlich zu verbessern und somit die Verbesserung der Qualität von Software zu verwirklichen [vgl. DRÖWIE S.2]. 1992 wurde daraufhin das V-Modell für alle Bundesbehörden erlassen. Viele Änderungswünsche der Benutzergemeinde führten zu einer starken Überarbeitung der ersten Fassung und zum V-Modell 97. Mittlerweile ist das V-Modell nicht nur ein Vorgehensmodell für die Softwareentwicklung im öffentlichen Bereich, es hat sich auch Unternehmen als eine Richtschnur für die Organisation von IT-Vorhaben durchgesetzt [vgl. WEIT]. Am 04. Februar 2005 wurde das Ergebnis der Weiterentwicklung in Form der Version 1.0 des V-Modell XT der Öffentlichkeit bei einer Auftaktveranstaltung an der TU München präsentiert.

3 Änderungennotwendigkeiten für das V-Modell 97

Schwerpunkte bei der Weiterentwicklung des V-Modell 97 zum V-Modell XT lagen in der Verbesserung der Benutzbarkeit und der Akzeptanz durch die Benutzer. Dies sollte durch eine inhaltliche Erweiterung realisiert werden, welche sich insbesondere durch das Optimieren der projekt- und organisationspezifischen Anpassbarkeit, also der Vereinfachung des Tailoring-Mechanismus, sowie der Skalierbarkeit auf unterschiedliche Projektgrößen, kennzeichnet.

3.1 Vorgehensbausteine

Bisher bestand das V-Modell aus den vier Submodellen Systementwicklung, Konfigurationsmanagement, Projektmanagement und Qualitätssicherung. Diese vier Tätigkeitsbereiche waren eng miteinander verknüpft. Das Tailoring führte dabei nur dazu, einzelne Schritte in den jeweiligen Submodellen außen vor zu lassen um somit unnötig erscheinende Regelungen nicht durchführen zu müssen und dadurch den Projektablauf effizienter gestalten zu können. Im Gegensatz zum bisherigen Standard wurden keine Submodelle gemacht, sondern diese nunmehr auf 18 modular aufgebaute Vorgehensbausteine erweitert. Wie bisher wird in den einzelnen Bausteinen vorgegeben, welche Produkte in einem Projekt zu erstellen sind, welche Aktivitäten dadurch durchgeführt werden müssen und welche Rolle dafür jeweils verantwortlich ist. Ein Großteil der Bausteine sind hierbei optional und vereinfachen deshalb die Anwendung des V-Modells.

3.2 Projekttyp

Für das Weglassen einzelner Regelungen gab es bisher keine Vorgabe für verschiedene Projekttypen, es musste jede Streichung vom Projektleiter manuell erfolgen und durch die Ausarbeitung einer Streichbedingung begründet werden. Im V-Modell XT wird nun durch die Auswahl von verschiedenen Projektmerkmalen und mit Hilfe des Open Source Projektassistenten automatisch festgelegt, welche Vorgehensbausteine zu wählen sind und welche Projektdurchführungsstrategie zu verwenden ist. Daraufhin kann eine projektspezifische V-Modell Dokumentation exportiert werden, welche nur die für das Projekt benötigten Inhalte besitzt.

3.3 Projektdurchführungsstrategie

Eine relativ statische Entwicklung charakterisierte bisher das V-Modell 97. Im V-Modell XT existiert nun bewusst keine Vorgabe oder Einschränkung bezüglich einer möglichen Abfolge, um eine höhere Flexibilität bei der Softwareentwicklung zu ermöglichen. Um aber einen dynamischen Entwicklungsprozess, der sich daraus ergibt, weiterhin steuern zu können, kann für jeden Vorgehensbaustein kein willkürlicher Ablauf zugelassen werden. Um einen geregelten Ablauf des Projektes zu ermöglichen, stellt das V-Modell einen Katalog von „Projektdurchführungsstrategien“ bereit, welche einen grundlegenden Rahmen für eine geordnete und nachvollziehbare Durchführung des Projektes definieren.

3.4 Entscheidungspunkte und Projektfortschrittsstufen

Eine Projektdurchführungsstrategie legt, wie bereits erwähnt, einen organisatorischen Rahmen für die Durchführung des Bausteins fest. Diese Strategie hängt jedoch stets von einer gewissen Reihenfolge der zu erreichenden Projektfortschrittsstufen ab. Das Erreichen einer solchen Stufe wird im V-Modell durch so genannte „Entscheidungspunkte“ gekennzeichnet, welche den bekannten Meilensteinen eines Projekts entsprechen.

3.5 Tailoring

Um bisher ein Anpassung an ein spezielles Projekt vorzunehmen, mussten im Rahmen des Tailoring individuell und ohne bestimmte Vorgaben nicht benötigte Produkte und Aktivitäten gestrichen werden. Bei der Spezifizierung auf das konkrete Projekt wird beim aktualisierten Vorgehensmodell auf Grund von charakteristischen Projektmerkmalen ein „Projekttyp“ ausgewählt. Durch diese projektspezifischen Angaben in Kombination mit dem Open Source Tool „V-Modell XT Projektassistent“ kann das generische Modell innerhalb weniger Minuten auf die Projektgegebenheiten angepasst werden. Als Ergebnis liefert das Tool eine V-Modell Dokumentation, die nur die benötigten Teile der kompletten Dokumentation beinhaltet, als auch die dazugehörige Projektdurchführungsstrategie zusammen mit den im Projektverlauf zu erstellenden Produkten und durchzuführenden Aktivitäten.

4 Schlussbemerkung

Durch die Fortschreibung des V-Modells wurde ein riesiger Schritt in Richtung einer verbesserten Anwendbarkeit, Skalierbarkeit, sowie Änder- und Erweiterbarkeit des V-Modells gemacht [vgl VXT1]. Durch das durchgeführte Projekt kann festgehalten werden, dass hierbei die Vereinfachung des Tailoring sehr positiv zu bewerten ist. Nun kann innerhalb kürzester Zeit die Anpassung an das eigene Projekt erfolgen. Gerade für unerfahrene Projektleiter war dies bisher mit erheblichem Aufwand verbunden [vgl DRÖWIE]. Zudem wird die Anwendbarkeit für kleine IT-Vorhaben noch dadurch verbessert, dass eine Richtschur für die Projektdurchführung auf Grund der Projektdurchführungsstrategie vorgegeben wird, innerhalb dieser dennoch flexibel verfahren werden kann. Kritisch ist jedoch immer noch der Aufwand bei der Projektorganisation zu bewerten. Durch die Erfahrungen im Projekt war zu erkennen, dass hier immer noch ein relativ hoher Dokumentations- und Einarbeitungsaufwand nötig ist. Dieser ermöglicht einerseits zwar die Einhaltung von Qualitätsrichtlinien, beschert jedoch gerade unerfahrenen Anwendern des V-Modells lange Einarbeitungszeiten. Abschliessend ist zu sagen, dass die Durchführung von kleineren Projekten mit Hilfe des neuen Vorgehensmodells durchaus sinnvoll sein kann, da durch relativ wenig Aufwand eine geordneter Rahmen zur Durchführung von kleinen Projekten erstellt werden kann. Jedoch kommt die zum Teil dürftige Beschreibung einzelner Produkte und die komplexe Verknüpfung der einzelnen Produkte der Anwendbarkeit nicht zu Gute. Ob das V-Modell XT mehr Verwendung finden wird, hängt von der jeweiligen Projektsituation sowie der Bereitschaft der Projektleiter ab, diesen Mehraufwand auf sich zu nehmen.

Literaturverzeichnis

- [BAL] Helmut Balzert: Lehrbuch der Software-Technik – Software-Management, Software-Qualitätssicherung, Unternehmens-Modellierung. Spektrum Akademischer Verlag 2000
- [BRÖDRÖ] Adolf-Peter Bröhl, Wolfgang Dröschel: Das V-Modell – Der Standard für die Softwareentwicklung mit Praxisleitfaden. R. Oldenburg Verlag 1993
- [CHR] Gerhard Chroust: Modelle der Softwareentwicklung. R. Oldenburg Verlag 1992
- [DRÖWIE] Wolfgang Dröschel, Manuela Wiemers: Das V-Modell 97 – Der Standard für die Entwicklung von Ist-Systemen mit Anleitung für den Praxiseinsatz. Oldenburg Wissenschaftsverlag 2000
- [V200X] Michael Meisinger, Andreas Rausch, Martin Deubler, Michael Gnatz, Ulrike Hammerschall, Inga Küffer, Sascha Vogel: Das V-Modell 200x – ein modulares Vorgehensmodell. 11. Workshop der Fachgruppe "Akzeptanz von Vorgehensmodellen" (Gesellschaft für Informatik) 2004
<http://www.kbst.bund.de/Anlage305522/V-Modell+200x+-+ein+modulares+Vorgehensmodell+%28635+kB%29.pdf>
- [VM92] V-Modell 92 Dokumentation - Allgemeiner Umdruck Nr. 250
<http://www.v-modell.iabg.de/vm92.htm>
- [VM97] V-Modell 97 Dokumentation - Allgemeiner Umdruck Nr. 250
<http://www.v-modell.iabg.de/vm97.htm>
- [VXT] V-Modell XT Homepage des BMI KBSt.
<http://www.kbst.bund.de/V-Modell/-,293/V-Modell-XT.htm>
- [VXT1] Manfred Bory: Das neue V-Modell XT. Beiträge des Ansstand e.V. Erfahrungsaustausches 2004.
<http://www.kbst.bund.de/Anlage305739/anlage305739.pdf>
- [WEIT] Homepage des ANSSTAND e.V. - Anwender des Software-Entwicklungsstandards der öffentlichen Verwaltung
<http://www.ansstand.de/alle/2003/ea2003v5.pdf>

Code Analysis Plugin (für Eclipse)

Johannes Schneider, Matthias Mergenthaler

FH Reutlingen
Wirtschafts-Informatik
cap@johannes-schneider.info
matthias.mergenthaler@student.fh-reutlingen.de

Art der Arbeit: Softwareprojekt
Betreuer/in der Arbeit: Prof. Dr. Fritz Laux
Vertrauensdozent: Prof. Dr. A. Zimmermann

Abstract: Im Rahmen des Software-Projekts ist ein Plugin für die Eclipse-Plattform entstanden, welches die Architektur und das Design von Java-Projekten analysiert und bewertet. In einer eigenen Perspektive werden die ermittelten Werte und potentielle Schwachstellen grafisch aufbereitet und dem Entwickler zur Verfügung gestellt.

Was das Code Analysis Plugin leistet

Ziel des Plugins ist es, Kennzahlen zu ermitteln, die das Design von Java-Projekten bewerten. Durch die ermittelten Kennzahlen werden typische Schwächen und Fehler in der Architektur aufgedeckt, welche die Wartbarkeit und die Wiederverwendbarkeit von Java-Klassen negativ beeinflussen.

Diese Kennzahlen stellt das Plugin dem Entwickler bereits so früh wie möglich zur Verfügung stehen, um Designprobleme frühest möglich erkennen und beheben zu können.

Die Grundlagen des Code Analyse Plugins

Das Projekt basiert auf dem 1994 von Robert Martin veröffentlichten Dokument „OO Design Quality Metrics“ [RM00]. In diesem Dokument werden die theoretischen Grundlagen und die zur Bewertung von Softwareprojekten geeigneten Kennzahlen in grundsätzlicher Form beschrieben. Die Firma Clarkware Consulting hat diese theoretischen Überlegungen bereits in dem einfachen Tool „Jdepend“ [MC00] umgesetzt.

CAP erweitert diese Funktionalität einerseits um erweiterte Berechnungen der Kennzahlen – bis hinab auf Klassenebene. Zusätzlich wurde der Fokus jedoch auf eine komfortable und übersichtliche Darstellung der ermittelten Werte in grafischer Form gelegt, wodurch die theoretischen Überlegungen erst praktisch nutzbar werden.

Was das Code Analyse Plugin leistet

Das Code Analysis Plugin für Eclipse ermöglicht es, jederzeit während der Entwicklung in Eclipse eine Analyse des Designs durchzuführen. Damit wird es dem Entwickler ermöglicht, seinen Code sofort zu analysieren, um das Design seines Projekts entwicklungsbegleitend verbessern zu können.

Das Plugin stellt dem Entwickler die zur Bewertung der Architektur benötigten Kennzahlen in übersichtlichen, funktionalen Diagrammen und in Tabellenform dar. Im Zentrum der Analyse steht eine interaktive Grafik in welcher die analysierten Beziehungen zwischen den Packages grafisch aufbereitet werden.

Dem Entwickler wird mit Hilfe dieses Beziehungsbaumes ein „Plan“ des Projekts gezeichnet, was den Vorteil mit sich bringt, dass oftmals schon auf den ersten Blick Schwachstellen der Designs zu erkennen sind.

Das Plugin hilft somit beim Erkennen der „typischen“ Schwächen und Fehler einer Architektur. Die Wartbarkeit und Wiederverwendbarkeit kann damit durch das Optimieren der Kapselung von Softwareprojekten stark verbessert werden.

Berechnungen und Kennzahlen

Verschiedene Kennzahlen werden berechnet, um das Design bewerten zu können. Zuerst werden die efferenten bzw. afferenten Verbindungen zu anderen Packages bzw. Klassen summiert, um dann mit Hilfe der daraus berechneten Werte der „Instability“ und „Abstractness“ die Distanz von einem theoretischen Optimal-Wert zu finden.

Große Abweichungen von diesem optimalen Wert deuten dann auf Schwachpunkte und Problemzonen hin.

Verwendete Technologien

Das Code Analyse Plugin wurde als Plugin für die Eclipse-Plattform in der Version 3.0 in Java entwickelt. Als Java Runtime ist mindestens die Version 1.4 unter Windows bzw. 1.5 unter Linux notwendig.

Neben dem bereits in Eclipse enthaltenen JDT-Plugin werden zusätzlich noch die Plugins „JFreeGraph“ [JF00] für die Darstellung der Balken- und Kuchendiagramme, sowie „GEF“ [GEF] für die grafische Darstellung benötigt.

Sonstiges

Um den Grundgedanken von Eclipse weiter zu führen, wurde das Plugin als Open Source unter der GPL veröffentlicht.

Das Plugin ist bei Sourceforge.net [CAP01] gehostet und kann von dort bezogen bzw. per Eclipse-eigenem Update Manager über die Adresse [CAP02] installiert werden.

Auf der Projekt-Seite [CAP02] sind Links zu allen benötigten Ressourcen zu finden. Außerdem stehen Hilfen zur Installation, Dokumentationen und eine Präsentationen zur Verfügung.

Quellen-Verzeichnis

[CAP00] Sourceforge-Seite: <http://sourceforge.net/projects/cap4e/>

[CAP01] Projekt-Seite: <http://cap.xore.de>

[CAP02] Update-Site: <http://cap.xore.de/update>

[RM00] "OO Design Quality Metrics" - von Robert Martin.
(<http://www.objectmentor.com/resources/articles/oodmetric.pdf>)

[MC00] JDepend - von Mike Clark
<http://www.clarkware.com/software/JDepend.html>

[EC00] "Eclipse Anwendungen und Plug-Ins mit Java entwickeln" - Sherry Shavor, Jim D'Anjou, Scott Fairbrother, Dan Kehn, John Kellerman, Pat McCarthy
Addison-Wesley, ISBN: 3-8273-2125-5 [Ez99] Ezgarani, O.: The Magic Format – Your Way to Pretty Books, Noah & Sons, 2000.

[EC01] "Contributing to Eclipse" - Kent Beck, Erich Gamma
Addison-Wesley, ISBN: 0-321-20575-8

[JF00] JFreeGraph – verwendete Grafik-Bibliothek <http://www.jfree.org/jfreechart/>

[GEF] Graphica Editing Framework – <http://www.eclipse.org/gef/>

Ein Speicherprofiler für Enterprise Java Beans

Bernhard Volz

Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl für Informatik 6 – Datenbanksysteme
sibevolz @ ce.stud.uni-erlangen.de

Art der Arbeit: Bachelorarbeit
Betreuer/in der Arbeit: Prof. Dr. Klaus Meyer-Wegener,
Dipl.-Inf. Marcus Meyerhöfer

Abstract: Die Ermittlung der Ressourcen-Eigenschaften von Komponenten wie zum Beispiel der Speicherverbrauch ist eine Voraussetzung für die Spezifikation nicht-funktionaler Eigenschaften von Softwarekomponenten. Mit Hilfe dieser zusätzlichen Informationen erhalten die Entwickler von Softwaresystemen die Möglichkeit, Komponenten nicht nur bezüglich ihrer Funktion auszuwählen. Diese Arbeit stellt einen Profiler für Enterprise Java Beans (EJBs) vor, der speziell auf die Charakteristika dieses Komponentenmodells angepasst ist. Er vereinfacht eine Fokussierung auf eine Komponente ohne sich dabei in den Details der Klassen zu verlieren, die diese EJB ausmachen und erlaubt gleichzeitig verschiedene Sichten auf die ermittelten Profiling-Daten betreffs der zum Speicherverbrauch angerechneten Teile einer Komponente.

Moderne Enterprise-Anwendungen verwenden Komponentenmodelle wie CORBA [OMG02] oder Enterprise Java Beans (EJB) [Sun03a], um Anwendungsfunktionalität auf mehrere Systeme zu verteilen. Die eigentliche Anwendung wird dann wie bei einem Baukastensystem aus einzelnen Komponenten zusammengesetzt. Bei der Entscheidung, welche Bestandteile in einer Anwendung verwendet werden sollen, spielen neben der eigentlichen Funktion (funktionale Eigenschaften) auch nicht-funktionale Eigenschaften wie Antwortzeit und Speichernutzung eine Rolle bei der Auswahl von Komponenten. Das von der DFG geförderte Forschungsprojekt "COMQUAD" (Components with Quantitative Properties and Adaptivity) beschäftigt sich mit der Thematik nicht-funktionaler Eigenschaften und damit, wie diese in den Softwareentwicklungsprozess eingebunden werden können ([RZ03] und [MWM03]).

Im Rahmen dieses Forschungsprojekts wurde eine Speicher-Profiler für Enterprise Java Beans entwickelt, um den Speicherverbrauch von Softwarekomponenten für die J2EE-Plattform [Sun03] ermitteln zu können. Die implementierte Lösung verwendet das Java Virtual Machine Profiler Interface (JVMPi) [Sun02] von Sun, das auf Ereignis-Reports der virtuellen Maschine beruht. Weil diese Schnittstelle im neuen Java5-Standard, der erst kurz vor dem Ende der Bearbeitungszeit veröffentlicht wurde, durch das Java Virtual Machine Tool Interface (JVMTI) [Sun04] ersetzt wird, beschreibt die Ausarbeitung auch noch die Umstellung der Lösung auf diese neuen Bedingungen.

Mit Hilfe des nun vorliegenden Profilers ist es möglich, den von einer Komponente verbrauchten Speicher unter verschiedenen Gesichtspunkten und in verschiedener Granularität aufzuzeigen. Unter anderem bietet der Profiler die Möglichkeit, Anwendungsserver, die für die Bereitstellung von Enterprise Java Beans benötigt werden, bezogen auf ihre Infrastruktur miteinander zu vergleichen.

Da die auf dem JVMPi basierende Lösung eine nicht unerhebliche Auswirkung auf die Performance eines Anwendungsservers hat, werden Ansätze zu weiteren Analysen vorgestellt, die versprechen, den Performanceverlust einzuschränken. Ebenso wird eine Erweiterung diskutiert, die es ermöglicht, nicht nur eine einzelne Komponenten zu untersuchen, sondern auch komplexere Szenarien, in denen Aufrufbeziehungen zwischen Enterprise Java Beans berücksichtigt werden können.

Die Bachelorarbeit untersucht ferner einige kommerzielle und freie Profiler sowie akademische Frameworks, die ein Speicher-Profiling von Java-Anwendungen ermöglichen. Die beiden Hauptprobleme bei diesen (die eben auch ein Grund waren, diese Arbeit auszugeben) sind: Keiner der untersuchten Profiler (JProfiler, Netbeans Profiler, Borland OptimizeIt, JView 2004) ist in der Lage, Komponenten als solche zu untersuchen und darzustellen oder diese Funktionalität nachzurüsten, da Schnittstellen für die Erweiterung oder die Integration bestehender Agenten in eigene Tools fehlen. Das zweite Problem entsteht durch die zu breit angelegten Ziele und Einsatzzwecke vieler Profiler und Profiling-Frameworks: Daten werden grundsätzlich über alle eintretenden Ereignisse in der virtuellen Maschine gesammelt und an den Profiler weitergeleitet, unabhängig davon, ob diese für das jeweilige Profiling-Szenario benötigt werden oder nicht (vergl. FORM [SMS01]). Einzig die Präsentationsschicht eines Profilers trifft dann die Auswahl, welche Daten in Betracht kommen.

Abgeschlossen wird die Arbeit durch Messungen des entwickelten Systems an verschiedenen komponentenbasierten Anwendungen, die die Performance des derzeitigen Prototyps aufzeigen und Hinweise für künftige Optimierungen liefern.

Literaturverzeichnis

- [Sun03a] Enterprise Java Beans Specification, Version 2.1, Sun Microsystems, November 2003, <http://java.sun.com/products/ejb/docs.html>
- [OMG02] CORBA Component Model, Version 3.0, Object Management Group, Juni 2002, <http://www.omg.org/technology/documents/formal/components.htm>

- [Sun03] Java(TM) 2 Platform Enterprise Edition Specification, Version 1.4, Sun Microsystems, November 2003
- [Sun04] The JVM Tool Interface (JVMTI), Sun Microsystems, <http://java.sun.com/j2se/1.5.0/docs/guide/jvmti/jvmti.html>
- [Sun02] The Java Virtual Machine Profiler Interface (JVMPi), Sun Microsystems, <http://java.sun.com/j2se/1.4.2/docs/guide/jvmpi/jvmpi.html>
- [SMS01] T. Sounder, S. Mancoridis, M. Salah, Form: A Framework for Creating Views of Program Execution. Drexel University, Department of Mathematics and Computer Science, 2001
- [RZ03] S. Röttger, S. Zschaler, A Software Development Process Supporting Non-Functional Properties. Institute for Software and Multimedia Technology, Department of Computer Science, Dresden University of Technology, 2003
- [MWM03] R. Aigner, H. Berthold, S. Göbel, H. Härtig, H. Hußmann, K. Meißner, K. Meyer-Wegener, M. Meyerhöfer, A. Pfitzmann, S. Röttger, A. Schill, T. Springer, F. Wehner, COMQUAD. Komponentenbasierte Softwaresysteme mit zusagbaren quantitativen Eigenschaften und Adaptionfähigkeit, in "Informatik. Forschung und Entwicklung" 18 (2003), Nr. 1, S. 39-40

Eine strategierorientierte, modulare Simulationsumgebung für mobile Ad-Hoc-Szenarien

Stefanie Hofmann Marcel Kronfeld Andreas Monger Jan Bronni
{shofmann|marsecro|monger|bronni}@informatik.uni-tuebingen.de

Betreuer: Olaf Landsiedel, Leo Petrak, Klaus Wehrle
Wilhelm-Schickard-Institut für Informatik, Universität Tübingen

Abstract: Für die Evaluation von Netzwerkprotokollen in mobilen Szenarien entwickeln wir ein Framework, das Funkwellenausbreitung und Verhaltensweisen von Kommunikationsteilnehmern auf einer strukturierten Umgebungskarte realitätsnah simuliert.

1 Einführung

Um Kommunikationsprotokolle für mobile Netze zu evaluieren, werden Simulationen von mobilen Knoten in unterschiedlichen Szenarien benötigt. Gegenwärtige Bewegungsmodelle [1],[2],[3] bilden meist Bewegung auf offenem Feld nach bzw. orientieren sich an Fahrzeugen auf Straßen. Das „Obstacle Mobility Model“ (OMM) [4] beschreibt dagegen wie sich Einheiten auf Wegen und durch Gebäude einer vorgegebenen Karte bewegen, verwendet dabei aber einfache Konzepte. Um möglichst unterschiedliche Szenarien für die Protokollauswertung bereitzustellen, lehnen wir uns mit unserem Simulationsframework an das OMM an und fügen austauschbare, individuelle Bewegungsstrategien für die mobilen Knoten hinzu, sowie die Möglichkeit, dynamisch Gruppen zu bilden. Um die Funkverbindungen zwischen mobilen Knoten auszuwerten, haben wir das „Free Space Model“ (FSM) für Karten erweitert. Als Implementationsgrundlage dient OMNeT++ [5], ein Simulator für diskrete Ereignisse. Unser Projekt gliedert sich damit in die vier Bereiche Umgebungsmodell, individuelle Bewegung, Gruppenbildung und Funkwellenausbreitung, die wir im Folgenden näher ausführen.

2 Umgebungsmodell

Im Umgebungsmodell wird eine Karte durch Objekte repräsentiert, die Hindernisse, Gebäude oder Wegstücke darstellen und in einer Graphenstruktur verknüpft sind. Die Kartenelemente werden nicht nur durch Position und Lage definiert, sondern besitzen verschiedene weitere Attribute, wie zum Beispiel Materialeigenschaften zur Berechnung der Signalabschwächung. Die Karten werden in XML-Dateien beschrieben, in denen die oben genannten Attribute festgelegt sind. Das Umgebungsmodell dient zum Datenaustausch und als logischer Unterbau für die übrigen Module. Für die Bewegungsmodelle werden beispielsweise die Entfernungen zwischen Elementen der Karte vorab berechnet.

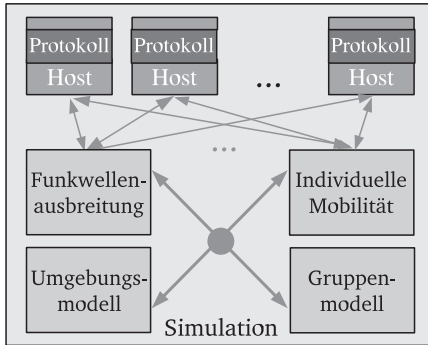


Abbildung 1: Modulare Simulationsumgebung

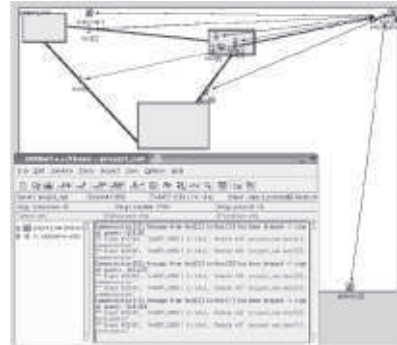


Abbildung 2: Screenshot einer Simulation

3 Individuelles Bewegungsmodell

Zur Bewegung eines Knotens auf der Karte müssen drei Teilprobleme gelöst werden: erstens muss das logische Ziel (ein Kartenelement) gesetzt werden, zweitens muss ein entsprechender Pfad im Graph der Karte bestimmt werden, der dann drittens in tatsächliche Bewegung umgesetzt werden kann. Im OMM wird das Ziel der Bewegung eines Teilnehmers weitgehend zufällig aus der Menge der Gebäude gewählt und der kürzeste Pfad beschritten. Die umgesetzte Bewegung besteht dann aus einer Punkt-zu-Punkt-Bewegung über die Verbindungen der Kartenelemente („Türen“, „Kreuzungen“). Diese Umsetzung behalten wir bei, für Ziel- und Wegewahl wenden wir dagegen realistischere Methoden an. Wir modellieren, dass sich einerseits Kartenelemente in mehr als nur Größe und Lage unterscheiden, und andererseits mobile Knoten unterschiedliche Vorlieben bei ihren Entscheidungen haben. Unser Konzept besteht darin, die potentiellen Ziele durch austauschbare, parametrisierbare Funktionen zu bewerten und anhand einer Selektionsfunktion auszuwählen. Eine einfache Bewertungsfunktion kann beispielsweise jedem Kartenelement einen zufälligen Wert zuordnen, eine komplexere beurteilt spezifische Eigenschaften der Kartenelemente und Teilnehmer; verschiedene Bewertungsfunktionen sind dabei mit variablen Gewichten beliebig kombinierbar. Diesen „evaluate-select“-Mechanismus verwenden wir auch zur Wegewahl. Ein Teilnehmer bestimmt hierbei wiederholt ein *nächstes Kartenelement*, wodurch dynamisches Verhalten begünstigt wird. Als sinnvollen Kompromiss zwischen Einfachheit und Realismus verwenden wir folgende maßgebliche Unterscheidungskriterien bei Ziel- und Wegewahl der Teilnehmer: (1) Position und Distanz zum Kartenelement, (2) eine zeitvariable Attraktivität pro Kartenelement, (3) die Korrelation zwischen spezifischen Attributen von Kartenelement und mobilem Knoten sowie (4) die Empfangsqualität im Kartenelement. Durch gewichtete Kombination dieser Bewertungskriterien lassen sich realistische Verhaltensstrategien nachbilden.

4 Gruppenmodell

Gängige Gruppenmobilitätsmodelle (GMM) wie das „Reference Point Group Mobility Model“ oder das „Pursue Mobility Model“ [1] behandeln die Bewegung einzelner Teil-

nehmer in einer Gruppe, aber nicht wie sich Gruppen bilden und auflösen. Daher legen wir den Schwerpunkt in unserem Projekt darauf, soziales Verhalten der Teilnehmer zu modellieren und durch Vergleiche Übereinstimmungen zur Gruppenbildung zu finden. Dazu werden den Teilnehmern Bewertungsfaktoren zugewiesen: ein Geselligkeitsfaktor, der angibt, wieviel Wert ein Teilnehmer auf Gesellschaft legt, und ein Dominanzfaktor, der die Durchsetzungskraft gegenüber anderen Teilnehmern bestimmt. Diese werden herangezogen, um die Strategien der einzelnen Teilnehmer in näherer Umgebung zu gewichten und zu vergleichen. Kommt es aufgrund ähnlicher Strategien zur Bildung einer Gruppe, greifen die Mitglieder von nun an auf verschiedene GMM zurück: auf Pfaden bewegen sie sich mit mittlerer Geschwindigkeit, in Gebäuden nach dem „Pursue Mobility Model“. Mitglieder können eine Gruppe verlassen, wenn ihre Strategie nicht mehr mit der Gruppenstrategie übereinstimmt. Dies kann geschehen, wenn das Gruppenziel erreicht oder ein neues Mitglied aufgenommen wird.

5 Ausbreitung der Funkwellen

Bisherige Ansätze, wie zum Beispiel das „Mobility-Framework“ [6], beruhen auf dem FSM. Hierbei nimmt die Signalstärke mit der zweiten Potenz der Distanz ab. Ein anderer Ansatz wurde in [7] verfolgt. Auf Grundlage von Messungen in einem Bürogebäude wurde hier eine nahezu exakte Nachbildung eines Stockwerks erstellt und durch eine Erweiterung des FSM simuliert.

Unser Ansatz lehnt sich an beide Modelle an und versucht dadurch, Charakteristika der Funkwellenausbreitung in beliebigen Szenarien nachzubilden. Zusätzlich zur Signalabschwächung durch die Distanz werden Wände und andere Hindernisse in die Berechnung miteinbezogen. Die Berechnung der Signalstärke einer Übertragung ermöglicht es festzustellen, ob und in welcher Qualität eine Nachricht empfangen wird. Die bereitgestellte Netzwerkschnittstelle zur Übertragung von Paketen wurde so generisch gewählt, dass sich bestehende Simulationsmodelle von Protokollen leicht einbinden lassen. Ein weiterer Schritt, mit dem wir eine leistungsfähige Testumgebung bereitstellen, ist die Möglichkeit, verschiedene Geräte (z.B. WLAN, GSM, Bluetooth) zur selben Zeit zu simulieren und damit eine heterogene Geräteinteraktion zu evaluieren.

6 Ausblick

Die vorgestellte Simulationsumgebung vereint die Vorzüge von Bewegungs- und Funkwellenausbreitungsmodellen. Durch den modularen Aufbau erzielen wir erhöhte Flexibilität für realitätsnahe Simulationen. In Zukunft sind verschiedene Erweiterungen geplant. Ein Kartengenerator wird für die einfache Erstellung vielfältiger Schauplätze sorgen. Unter Verwendung eines Gedächtnisses werden individuelle Erfahrungen gesammelt, die sich auf Strategien und zukünftige Gruppenbildungen der mobilen Knoten auswirken. Eine Weiterentwicklung der Funkwellenberechnung sieht vor, die Signalabschwächung vorab zu berechnen, um Aufwand während der Simulationszeit zu sparen.

Literatur

- [1] T. Camp, J. Boleng und V. Davies. "A Survey of Mobility Models for Ad Hoc Network Research", *Wireless Communication & Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking Research, Trends and Applications*, Vol. 2, No. 5, pp. 483-502, 2002.
- [2] Markoulidakis, G. Lyberopoulos, D. Tsirkas, and E. Sykas. Mobility modeling in third-generation mobile telecommunications systems. *IEEE Personal Communications*, 4:41–56, August 1997.
- [3] F. Bai, N. Sadagopan, A. Helmy, "IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of Routing protocols for Adhoc Networks", *IEEE INFOCOM (The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies)*, PP.825-835, March/April 2003, San Francisco.
- [4] Amit Jardosh, Elizabeth M. Belding-Royer, Kevin C. Almeroth und Subhash Suri. "Towards Realistic Mobility Models for Mobile Ad hoc Networks." In the Proceedings of MobiCom, San Diego, CA, September 2003.
- [5] "OMNeT++" by András Varga. In the column "Software Tools for Networking", *IEEE Network Interactive*. July 2002, Vol.16 No.4
- [6] M. Löbbers, D. Willkomm - A Mobility Framework for OMNeT++; Technische Universität Berlin, 2004
- [7] A.L. Cavilla, G. Baron, T.E. Hart, L. Litty und E. de Lara - Simplified Simulation Models for Indoor MANET Evaluation are not Robust; Department of Computer Science - University of Toronto, 2004

Software Engineering für Embedded Systems

Philipp Konradi, Matthias Ecker, Christian Könik, André Hofmann

Fachhochschule Landshut

Fachbereich Informatik

KonradiPh@web.de, mail@matthias-ecker.de

koenikchristian@hotmail.com, andrefox@onlinhome.de

Art der Arbeit: Studienprojekt
Betreuer/in der Arbeit: Prof. Dr. Peter Scholz

Abstract: Der Markt für eingebettete Systeme überholt bald den Markt für normale Software. Gleichzeitig wachsen die Anforderungen an die Entwicklung von eingebetteten Systemen. Um diesen Anforderungen gerecht zu werden, bieten sich neue Ansätze und Techniken an. Diese speziellen Modellierungs-, Spezifikations- und Programmiersprachen werden im folgenden kurz vorgestellt und in einer Anwendungsstudie wird eine der neuen Technologien eingesetzt.

1 Einführung in Embedded Systeme

Eingebettete Systeme(ES) halten verstärkt Einzug in alle Tätigkeitsfelder und Lebensbereiche des Menschen, um nur einige davon zu nennen: Consumer Electronics, Automobilindustrie, Medizintechnik, Robotik. industrielle Automatisierung und Telekommunikation. Dieser Trend ist der Grund dafür, dass ES heutzutage einen sehr stark wachsenden Markt darstellen, der voraussichtlich bald den Markt für Terminalbasierte Anwendungen überholen wird[Mar00]. Das schnelle Wachstum bringt aber auch Herausforderungen mit sich, die neue Ansätze, Methoden und Techniken in der Entwicklung von ES erfordern.

Zuerst sollte man den Begriff ES klären. Ein ES ist eine Kombination aus Hardware und Software, die über Sensoren und Aktuatoren mit einem Gesamtsystem verbunden ist und darin Überwachungs-, Steuerungs- bzw. Regelungsaufgaben übernimmt. Typischerweise sind solche Systeme dem menschlichen Benutzer nicht direkt sichtbar[Sof98]. Einige Vertreter sind beispielsweise das ABS, die Motorsteuerung, das Handy und der DVD-Player. Kennzeichen von ES sind sehr hohen Anforderungen an die Zuverlässigkeit, Vertrauenswürdigkeit(Safety), Robustheit, Effizienz und Rechtzeitigkeit. Neue Einsatzfelder und die massive Verbreitung haben viele Schwachpunkte bei den herkömmlichen Entwicklungstechniken, die immer noch Assembler und C einsetzen, aufgedeckt.

So stellen der steigende Umfang und Komplexität sowie der steigende Anteil der Software im Inneren des Systems ein großes Problem für die herkömmlichen Methoden. Es wird nach Ansätzen gesucht, die eine Entwicklung auf hoher Abstraktionsebene unter Verwendung der Codegenerierung erlauben würden. Die Kosten für die Qualitätssicherung sind ein weiteres Problem. Sie waren bei den ES schon immer immens und angesichts des steigenden Umfangs werden die Forderungen nach besseren (formalen) Testverfahren laut. Auch die Einteilung des Systems in Hardware und Software ist problematisch. Diese kann heutzutage kaum revidiert werden. Man wünscht sich eine gemeinsame Spezifikation für die Soft- und Hardware[Shn03]. Diese kann dann als Einheit getestet, verifiziert und automatisch in die eine oder andere Form überführt werden.

Alle diese Punkte stellen hohe Anforderungen an die Entwicklung von eingebetteten Systemen dar. Aber wie begegnet man Ihnen und welche Möglichkeiten stehen zur Verfügung? Die Produktivität der Entwickler lässt sich nur um ca. 15% pro Jahr steigern[Shü04]. Zu wenig um mit dem Wachsen der Codegröße mitzuhalten. Das Einstellen von mehr Entwicklern ist auf Dauer nicht tragbar[Shü04]. Die einzige echte Alternative um den wachsenden Anforderungen zu begegnen und diese unter Kontrolle zu behalten ist der Einsatz neuer revolutionärer Technologien, Methoden und Prozesse in der Entwicklung von ES.

Aufgrund der steigenden Komplexität von ES versucht man durch Einsatz von Modellierungssprachen verstärkt die Entwicklung auf eine höhere Abstraktionsebene zu verlagern. Aber die herkömmlichen Modellierungssprachen werden nicht allen Anforderungen von ES gerecht. Aus diesem Grund wurde Anfang der 90-er ROOM speziell dafür entwickelt. ROOM sieht formal spezifizierte Semantiken für die Entwurfs-elemente vor, so daß der Entwurf jederzeit ausführbar und testbar ist. Weil die Semantik formal definiert ist, kann aus dem Entwurf automatisch Code generiert werden.

Aber auch bestehende Modellierungssprachen wurden den Anforderungen von ES angepasst. Dies ist z.B. bei UML 2.0, die viele Aspekte aus ROOM aufgegriffen hat, der Fall. Die neue Version bietet unter anderem neue Diagramme(Timing-Diagramme) und verbesserte Semantiken z.B. bei Statecharts an.

Einen ganz anderen Ansatz verfolgen Synchronische Sprachen. Sie entstanden Anfang der 80-er basierend auf dem Modell der „Perfekten Synchronie“. Diese besagt daß die Reaktionen auf Eingaben in der selben Zeitinstanz wie die Eingaben selbst erscheinen. Die synchronen Sprachen konnten schnell Fuß fassen in der Regelungstechnik, da sie sich nicht stark von den dort implizit verwendeten Modellen unterscheiden[Ber00]. Das besondere an den synchronen Sprachen ist, das ihre Semantik formal spezifiziert ist. Dadurch ist eine formale Verifikation und eine automatische Codegenerierung möglich. Die bekanntesten Vertreter von synchronen Sprachen sind Esterel für steuerungsentensive und Lustre für datenflussintensive Programme.

Auch die Implementierungsphase lässt sich durch die Wahl einer speziell für ES angepassten Programmiersprache effizienter gestalten. Assembler und C werden zunehmend von objektorientierten Sprachen, die an die Bedürfnisse von ES angepasst wurden, abgelöst. So wurde C++ bezüglich des Speicherverbrauchs, der Analysierbarkeit und der ROM-Lauffähigkeit optimiert. Dazu wurden beispielsweise die Exceptions und die Run-Time-Type-Identification aus dem Sprachumfang entfernt. Diese optimierte Version, als Embedded C++ bezeichnet, wird vor allem in Japan erfolgreich eingesetzt. Ein großer Vorteil der Sprache ist unter anderem der leichte Umstieg für viele C-Programmierer.

Java bietet speziell angepasste Plattformen, wie z.B. die Microedition(J2ME) oder Java Card, mit jeweils speziellen VM's und API's. Wobei die Portabilität eine große Rolle spielt. So ist beispielsweise eine Java-Card Applikation auf der Karte eines beliebigen Herstellers lauffähig. Des weiteren gibt es mehrere Ansätze um dem Aspekt der Echtzeit gerecht zu werden, von speziellen API's oder VM's bis zu zusätzlichen Sprachmitteln.

Der Einsatz neuer Technologien und Methoden bedingt häufig mit sich auch neue passende Entwicklungsprozesse. So ist beispielsweise Ropes ein UML-basierter objektorientierter Entwicklungsprozess für ES, beim dem am Ende eines iterativen Zyklus über mehrere Phasen ein Prototyp erstellt wird. So entsteht nach mehreren Zyklen ein fertiges System. Das Hardware/Software Codesign, die automatische Codegenerierung sowie das Testen spielen eine wichtige Rolle in diesem Prozess.

2 Anwendungsstudie

2.1 Ziele

Ziel war es eine Schiebedachsteuerung zu entwickeln, die auf mehrere Umwelteinflüsse wie starke Sonneneinstrahlung, Regen oder die Innenraumtemperatur reagiert. So wird im Falle eines plötzlich eintretenden Regens das geöffnete Schiebedach automatisch geschlossen. Je nach Temperatur im Auto und Sonneneinstrahlung schaltet sich die Verdunklung an oder aus. Aus Sicherheitsgründen beinhaltet die Steuerung einen Einklemmschutz, der das Schiebedach wieder öffnet falls sich beim Schließen ein Gegenstand darin befindet. Des weiteren darf die Steuerung das Schiebedach nicht komplett öffnen, sondern kann dieses lediglich zur Belüftung kippen.

2.2 Entwicklung

Für die Entwicklung der Schiebedachsteuerung fiel die Wahl auf die perfekt synchrone Sprache Esterel. Dafür waren mehrere Gründe ausschlaggebend. Der erste Grund ist die Charakteristik der Schiebedachsteuerung. Es nimmt die Umgebung über Sensoren wahr und reagiert auf diese mit der Ansteuerung von Aktuatoren. Speziell für solche steuerungsentensive reaktive Systeme ist Esterel konzipiert und bietet dafür geeignete Sprachkonstrukte.

Der nächste Grund war die Möglichkeit die Steuerung auf einer hohen Abstraktionsebene zu spezifizieren, die Spezifikation mittels Model Checking bereits in dieser frühen Phase zu verifizieren, auf Gegebenheit gewisser (Safety-)Eigenschaften zu prüfen und anschließend automatisch Code zu generieren. Dabei bleibt die Spezifikation unabhängig davon ob sie letztendlich als Software oder Hardware realisiert wird. Diese Wahl wird erst bei der Codegenerierung getroffen. Weiterhin erlaubt Esterel den Einsatz von graphischen Mitteln beim Entwurf. Dies erleichtert das Verständnis und hilft den Überblick zu behalten.

Da die Reaktionen auf Umwelteinflüsse frei konfigurierbar gehalten sind, wurde die Schiebedachsteuerung in mehrere parallel laufende Esterel-Module aufgeteilt. Diese entscheiden anhand der Sensordaten und der aktuellen Konfiguration ob jeweils ein Schließen, Lüften oder Verdunkeln möglich ist. Ein weiteres Modul berücksichtigt diese Entscheidungen sowie den aktuellen Zustand des Schiebedachs und steuert die Aktuatoren an.

Aus Zeit und Kostengründen war die Realisierung der Steuerung in einem realen Schiebedach nicht möglich. Deshalb ist die Steuerung in eine Softwaresimulation eingebettet worden. Diese stellt Sensoren zur Verfügung und visualisiert den Zustand des Schiebedachs als Animation.

2.1 Fazit

In unserem Fall hat sich der Einsatz der synchronen Sprache Esterel als richtig erwiesen. Der hohe Abstraktionsgrad erleichterte die Spezifikation und durch die automatische Codegenerierung konnte daraus sofort die fertige Steuerung erstellt werden. Ein anderes ES mit anderen Eigenschaften und Forderungen würde aber eventuell den Einsatz einer anderen Technologie und Technik erfordern. Die hohe Variation der Eigenschaften und die Forderung nach der effizientesten Realisierung machen spezielle, dem Problem angepasste, Ansätze unumgänglich.

Literaturverzeichnis

- [Sof98] M.Broy, M. von der Beeck, I. Krüger. Softbed: Problemanalyse für ein Großverbundprojekt ‚Systemtechnik Automobil – Software für eingebettete Systeme‘. Abschlußbericht einer Analyse im Auftrag des BMBF, 1998.
- [Mar00] Peter Marwedel, Vorlesung „Prozessrechnertechnik / Eingebettete Systeme“, Uni Dortmund, 2000.
- [Shn03] Prof. Dr. rer. nat. Klaus Schneider, Vorlesung „Entwurf von reaktiven Systemen mit synchronen Sprachen“, <http://ag-rs-www.informatik.uni-kl.de>, 2003
- [Ber00] Gérard Berry, „The Esterel v5 Language Primer“ Version v5 91, <http://www-sop.inria.fr/esterel.org/home.htm>, 2000
- [Shü04] Bernd Schürmann, Vorlesung „Entwicklung Eingebetteter Systeme“, Uni Kaiserslautern, 2004

BOGSY – ein Informationssystem für Botanische Gärten

Stefan Reimers, Stiefen Schilz, Christian Sonnenberg

Institut für Wirtschaftsinformatik
Westfälische Wilhelms-Universität Münster
Leonardo-Campus 3
D-48149 Münster

stre@wi.uni-muenster.de, stiefen.schilz@uni-muenster.de, sonnenbc@uni-muenster.de

Abstract: Die Sammlungen von Pflanzen und Pflanzenpräparaten Botanischer Gärten stellen die Basis für die systematische Forschung botanischer Institute dar. Der Austausch von Pflanzenmaterial steht mittlerweile unter strengen internationalen Auflagen. Dies macht eine umfassende sowie konsistente Speicherung und Übermittlung von Abstammungslinien erforderlich. Um diesen Anforderungen zu genügen, wurde an der Westfälischen Wilhelms-Universität Münster im Rahmen zweier Projektseminare ein datenbankgestütztes Informationssystem entwickelt. Im Folgenden wird der Entwurf und eine erste Realisierung dieses Informationssystems für Botanische Gärten namens BOGSY (**B**OTANIC **G**ARDEN **S**YSTEM) vorgestellt.

1 Anforderungen an ein Informationssystem für Botanische Gärten

Botanische Gärten verfolgen im Wesentlichen zwei Ziele. Einerseits ermöglichen sie der interessierten Öffentlichkeit einen Einblick in die Vielfalt der Pflanzenwelt. Andererseits stehen gerade die Universitätsgärten mit ihren umfangreichen Pflanzensammlungen im Mittelpunkt der botanischen Forschung und Lehre. Besondere Beachtung finden hierbei die Bereiche der Taxonomie und Ökologie. Darüber hinaus erfüllen die heutigen Gärten zahlreiche andere Aufgaben wie Natur- und Artenschutz sowie Umweltbildung und Bürgerberatung [BGB04].

Botanische Gärten sind definitionsgemäß Institutionen, die über eine dokumentierte Kollektion von (lebenden) Pflanzen verfügen. Die Kollektion dient im Kern dem Zweck der wissenschaftlichen Forschung sowie der Konservierung (siehe Herbarien) des Pflanzenreiches (vgl. [WJS99]). Zur Unterstützung der Forschung und zur Sicherstellung der Biodiversität werden Teile dieser Sammlungen und Samen hiesiger Pflanzen zwischen botanischen Instituten ausgetauscht.

Die Botanik als Teilgebiet der Biologie ordnet alle Samen und Pflanzen in einem hierarchischen System, einer so genannten Systematik, ein.¹

¹ Für eine Einführung in die Grundbegriffe der Botanik vgl. [EG+02, Str02, WH98].

Leider existiert derzeit keine allgemein akzeptierte Systematik. Vielmehr werden verschiedene Systeme verwendet, die sich hinsichtlich der Namensgebung als auch in der Festlegung von Hierarchien unterscheiden. Dies erschwert eine gartenübergreifende Informationsintegration. Zudem ist es wichtig, über einen langen Zeitraum Daten und Material zu archivieren und Änderungen am Status mitzuverfolgen. Gärten, die am IPEN (International Plant Exchange Network, [CBD03]) teilnehmen, müssen außerdem umfassende Daten zu einer Pflanze bzw. einem Pflanzenpräparat speichern, beginnend mit der ersten Wildsammlung. In Münster betrifft dies ca. 60.000 Datensätze.

Diese Datenmenge kann effizient nur mit einem Informationssystem verwaltet und verarbeitet werden. Heute existierende Systeme sind zumeist kontinuierlich und unstrukturiert gewachsen; Datenredundanzen und -inkonsistenzen sind keine Seltenheit. Die finanzielle Ausstattung der meisten Botanischen Gärten erlaubt allerdings keine teuren Software-Lizenzen zur Handhabung derartiger Probleme. Als Potenziale für Neuentwicklungen sind bspw. ein automatisierter Katalogaustausch und Online-Bestellungen über das Internet denkbar.

Es existieren einige große Projekte [BGB,SYS], die sich mit der Erstellung von Software für Botanische Institutionen befassen. Sowohl aus finanziellen, als auch aus fachlichen Gründen werden diese Systeme jedoch nicht flächendeckend eingesetzt. Die Konzeption des Informationssystems BOGSY adressiert die geschilderten Rahmenbedingungen explizit und bietet erste Implementierungen der angesprochenen Erweiterungen.

2 Funktionen des Systems

BOGSY ist als Mehrbenutzersystem ausgelegt, in dem verschiedene Benutzer unterschiedlichen Zugriff auf die Funktionen des Systems haben. Die Anwender lassen sich in „Mitarbeiter“ und „Externe“ einteilen. Um die Zugriffsrechte eines Benutzers festzulegen, wird ihm bei der Registrierung eine Rolle wie z.B. „Gärtner“ oder „extern“ zugewiesen. Auf Ebene der Rollen werden dann Zugriffsrechte für Teilfunktionalitäten erteilt. Da BOGSY auch für eine internationale Zusammenarbeit von Gärten konzipiert wurde, unterstützt es mehrere Sprachen; derzeit sind dies Englisch und Deutsch.

BOGSY bildet die Systematik der Pflanzen als flexibles, hierarchisches System ab, um den Anforderungen an sich ändernde Umstände in der Botanik gerecht zu werden. Taxa können neu angelegt, editiert oder in der Hierarchie verschoben werden, wobei der Anwender vom System unterstützt wird, um Fehleingaben zu verhindern.

Akzessionen² beziehen sich auf ein Taxon³ und lassen sich über eine eindeutige Nummer identifizieren. Auch Akzessionen sind hierarchisch gegliedert, um ihre genetische Abstammung vollständig abzubilden.

² Umfasst Pflanzen, Samen und Herbarbelege. Mögliche Eingänge sind neue Sprösslinge, gesammelte Samen oder Pflanzen von Partnerinstitutionen.

³ Bestandteil der Systematik (z.B. „*Bellis perennis* L.“ – das Gänseblümchen)

Zum Austausch biologischen Materials in Form von Samen oder Pflanzen enthält BOGSY ein Bestellwesen. Besonderheiten im Falle der Botanik sind der kostenlose Austausch von Samen und das Fehlen von Bestellmengen. BOGSY unterstützt die Verwaltung für alle Stufen eines Bestellprozesses, von der jährlichen Erstellung eines Kataloges über die Onlinebestellung bis hin zur Abarbeitung einer eingegangenen Bestellung.

3 Implementierung

BOGSY ist als Web-Architektur auf Basis eines 4-Schichten-Modells konzipiert, wobei bei der Auswahl der Komponenten darauf geachtet wurde, Software zu verwenden, die für unterschiedliche Plattformen und nach Möglichkeit kostenlos verfügbar ist. Zugang finden die Benutzer über einen Web-Browser, wie er auf jedem modernen PC installiert ist. Die Oberfläche ist in Abbildung 1 zu sehen. Serverseitig arbeitet das System mit dem Webserver „Apache 2“, PHP und einem relationalen Datenbanksystem [Vos00, CB04].

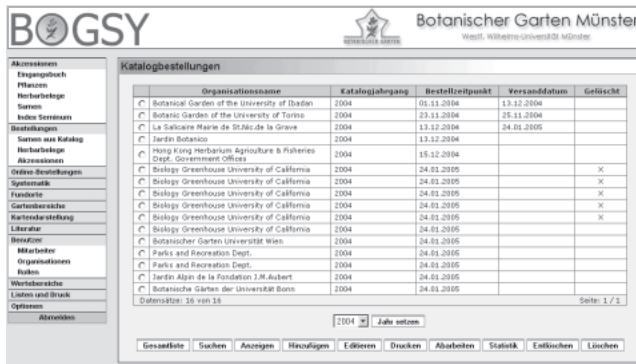


Abbildung 1: BOGSY Weboberfläche

Durch die Verwendung eines Datenbankabstraktionsschemas hat der Betreiber die Wahl bei der einzusetzenden Datenbank. Der Funktionsumfang des Systems ist in Module eingeteilt, welche an das Basissystem angehängt werden. Letzteres stellt den Modulen eine einheitliche Schnittstelle und Grundfunktionen wie den Datenzugriff zur Verfügung und bietet somit auch ein Framework für spätere Erweiterungen des Systems.

4 Fazit

Naturgemäß eröffnet die vollständige Neukonzipierung eines Informationssystems für einen spezifischen Anwendungsbereich, wie hier für den Bereich der Botanik, neue Wege, auf denen die begonnene Entwicklung fortgesetzt werden kann. Denkbar ist für BOGSY z.B. die Ausweitung des Nutzerkreises durch Entwicklung eines öffentlich zugänglichen Informationsportals, das über das Internet oder im Garten verfügbar ist.

Literaturverzeichnis

- [BGB04] Botanischer Garten der Universität Bonn: *Botanische Gärten und Biodiversität*, http://www.botanik.uni-bonn.de/botgart/f+e_hintergrund.html
- [BGB] BG-BASE Collections Management Software, <http://www.rbge.org.uk/bg-base/>
- [CB04] Connolly, T.; Begg, C. (2004): *Database Solutions. A step-by-step guide to building databases*. Second Edition, Addison Wesley / Pearson, Harlow, UK.
- [CBD03] EU Consortium of Botanic Gardens: *International Plant Exchange Network (IPEN) as model for implementation of the CBD by Botanic Gardens*, November 2003 (online: <http://www.bgci.org.uk/files/2/795/ipencodeofconduct.doc>)
- [EG+02] Erhardt, W.; Götz, E.; Bödeker, N., Seybold, S; Zander, R. (2002): *Zander - Dictionary of plant names*. Ulmer Verlag, Deutschland.
- [Str02] Strasburger, E. (2002): *Lehrbuch der Botanik*. Spektrum Verlag, Heidelberg, Deutschland.
- [SYS] SysTax, <http://www.biologie.uni-ulm.de/systax/>
- [Vos00] Vossen, G. (2000): *Datenmodelle, Datenbanksprachen und Datenbankmanagement-systeme*. R. Oldenbourg Verlag, München.
- [WJS99] Wyse Jackson, P.S; Sutherland, L.A. (2000): *International Agenda for Botanic Gardens in Conservation*, Botanic Gardens Conservation International, Kew, Richmond, UK.
- [WH98] Wisskirchen, R.; Haeupler, H. (1998): *Standardlist der Farn- und Blütenpflanzen Deutschlands mit Chromosomenatlas von Focke Albers*. Ulmer Verlag, Deutschland.

Implementierung von Dosis-Wirkungs-Modellen der Radiotherapie

Tim Langhammer, Sandro Lambeck und Lennart Opitz

northpool@west.de, nite@west.de, kugelblitz2y@west.de

Martin-Luther-Universität Halle-Wittenberg

Fachbereich Mathematik und Informatik - Institut für Informatik

Betreuer der Arbeit: Dr. Holger Blaar¹ und Dr. Reinhard Gerlach²

Art der Arbeit: Projektarbeit

Studiengang: Bioinformatik

Zusammenfassung

Bei der Bestrahlung von Tumoren können Schäden an benachbarten Organen (kritische Organe) nicht immer vermieden werden. Mit Hilfe von Dosis-Wirkungs-Parametern lässt sich das Schadensausmaß der Bestrahlungsdosis beurteilen. Gegenwärtig sind diese Parameter nicht für alle Risikoorgane bekannt. Im Rahmen der vorliegenden Arbeit wurde eine Java-Applikation entwickelt, die eine Auswahl von Bestrahlungsplänen von Patienten ermöglicht und über eine Auswertung von Radiotherapiedaten Dosis-Wirkungs-Parameter bestimmen kann.

1. Einleitung

Die Strahlentherapie ist neben der Chirurgie und der Chemotherapie die wichtigste Therapiemethode bei bösartigen Krebstumoren. Um eine Schädigung der kritischen Organe im Bestrahlungsgebiet möglichst gering zu halten, erfolgt, basierend auf einer CT-Untersuchung, eine dreidimensionale Bestrahlungsplanung mit der der Tumor durch die Bestrahlung vollständig erfasst wird und gleichzeitig das umliegende gesunde Gewebe geschont werden soll.

Im Allgemeinen liegt eine nicht-homogene 3D-Dosisverteilung vor, die mit Hilfe von Dosis-Volumen-Histogrammen (DVH) dargestellt wird. Diese werden durch ein Bestrahlungssystemprogramm erzeugt und sind mehrstufig. Die DVHs werden auf einstufige Histogramme reduziert. Dies geschieht unter der Annahme, dass eine Dosisverteilung gemäß dem resultierenden Histogramm mit der gleichen Wahrscheinlichkeit zu einer Komplikation führt wie eine Verteilung gemäß dem ursprünglichen Diagramm. [2]

¹Fachbereich Mathematik und Informatik - Institut für Informatik

²Klinik und Poliklinik für Strahlentherapie

Zentraler Bestandteil dieser Projektarbeit ist die Implementierung eines etablierten Dosis-Wirkungs-Modells, auf dessen Basis es möglich ist, Entscheidungsparameter zur Güte der Bestrahlung zu berechnen. Diese Parameter bestimmen die Normalgewebe-Komplikationswahrscheinlichkeit (NTCP).

2. Modelle

Die zur Berechnung der NTCP nach Lyman benötigten Dosis-Wirkungsparameter „ TD_{50} “ und „ m “ wurden auf Basis dieses Modells mit Hilfe der Maximum-Likelihood-Schätzung bestimmt. Daraus resultiert ein Kurvengebirge (Likelihood Landscape) dessen Maximum gesucht wird. Hierfür wurde der Nelder-Mead-Simplex-Algorithmus [8] implementiert.

Der Parameter „ TD_{50} “ ist die organspezifische Toleranzdosis, bei der die Komplikationswahrscheinlichkeit 50 % beträgt, wobei „ m “ den Anstieg der Dosis-Wirkungs-Kurve in diesem Punkt charakterisiert.

Die Transformation der DVH's erfolgte gemäß dem Dosis-Volumen-Reduktionsschema nach Kutcher [3] welches den für die Berechnung der NTCP-Funktion wichtigen Parameter V_{eff} bestimmt (effektives Volumen zur maximalen Dosis). Für die funktionale Modellierung der Komplikationswahrscheinlichkeit, welche bei der Auftragung im Dosis-Wirkungs-Diagramm ein sigmoides Aussehen hat, existieren zwei verschiedene Ansätze.[2]

Das sogenannte phänomologische Modell von Lyman entspricht dem Gaußschen Fehlerintegral:

$$NTCP(t) = (t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t e^{-\frac{x^2}{2}} dx$$

$$t = t(D, v, d, A) \tag{1}$$

Die Beschreibung der Komplikationswahrscheinlichkeit mit einer logistischen Funktion sieht folgendermaßen aus:

$$NTCP(D, v, d, A) = (D, v, d, A) = \frac{1}{1 + \left(\frac{TD_{50}(v, d, A)}{D}\right)^k} \tag{2}$$

Dabei sind :

- D die Bestrahlungsdosis
- v das bestrahlte Teilvolumen des Referenzvolumens
- d die homogen applizierte Fraktionsdosis
- A die Strahlungsart(Photonen, Neutronen, Protonen usw.)
- k bzw. m beschreibt den Anstieg der Dosis-Wirkungskurve bei TD_{50}

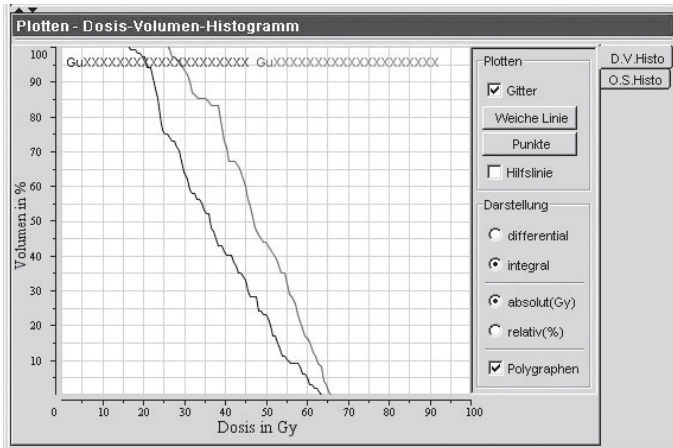


Abbildung 1: Dosis-Volumen-Histogramm für zwei Patienten

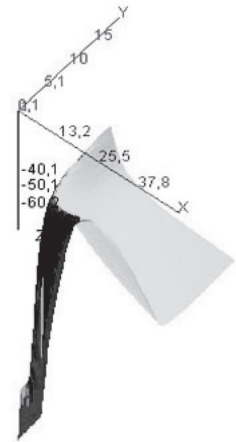


Abbildung 2: MLF^3

3. Konfidenz der geschätzten Parameter

Um die Signifikanz der sensiblen Parameter zu prüfen, wurden verschiedene Ansätze gewählt. [7] Als notwendiges Kriterium erwies sich die Bestimmung des Konfidenzintervalls aus Literaturangaben [5] gemäß einer χ^2 -Verteilung mit einem Freiheitsgrad. Da die vorliegenden Schätzungen der Parameter möglichst unabhängig verifiziert werden sollen, haben wir auch Clusteralgorithmen zur Klassifikation eingebaut. Mithilfe des k-Means Algorithmus wurde eine Vorhersage der Komplikationswahrscheinlichkeit von zu bestrahlenden Patienten anhand vorliegender Patientendaten ermittelt. Die Clusterzentren dabei ergeben sich somit anhand der Verteilung der Datenvektoren.

Bei unzureichender Anzahl an Datenmaterial hat sich die Nutzung der Monte-Carlo-Simulation zur fiktiven Datengenerierung anhand der Stichprobeneigenschaften oder der zu erwartenden Verteilung angeboten. [9]

4. Realisierung

Bisher konnten die Dosis-Wirkungs-Parameter nur mit Hilfe mehrerer Programme sehr zeitaufwändig berechnet werden. Jetzt hat unsere Projektgruppe ein Werkzeug geschaffen, welches die Funktionalität dieser Programme impliziert und darüber hinaus noch weitere Möglichkeiten bieten kann, die vorher nicht möglich waren (Clustering, Monte-Carlo-Simulation). Aufgrund objektorientierter Programmierung in Java [10] ist das Programm leicht pflegbar und erweiterbar. Aktuelle Standards der Informatik wie auch der Bioinformatik wurden implementiert. So kann man die Patientendaten bequem als XML-Datei abspeichern, was eine Schnittstelle zu anderen Dateiformaten (z.B. HTML) und anderen XML-fähigen Programmen darstellt. Auf Anwenderfreundlichkeit wurde großen Wert gelegt. Die Entwicklung des Programms erfolgte dementsprechend in enger Kooperation mit den zukünftigen Benutzern. Alle wichtigen Programmteile sind auf einen Blick erkennbar und leicht bedienbar. Damit wurde ein professionelles Programmsystem entwickelt, was

³3D-Darstellung der Maximum-Likelihood-Funktion im Programm

aktuelle Forschungsarbeiten auf dem Gebiet der Radiotherapie unterstützt.

5. Test des Programms für die Ohrspeicheldrüse (Parotis)

Ziel war die Bestimmung der Dosis-Wirkungs-Parameter für das kritische Organ Parotis. Die Kenntnis dieser Parameter ist eine Grundvoraussetzung zur Optimierung einer dreidimensionalen Dosisverteilung mit dem Ziel der Vermeidung bzw. Minimierung der Xerostomie (Mundtrockenheit). Zur Verfügung standen Daten von 32 Patienten, die einer Strahlentherapie im Kopf-Hals-Bereich unterzogen wurden. Im Verlauf (ein Monat nach Therapiebeginn) und 12 Monate nach der Strahlenbehandlung wurde die Speichelmenge bestimmt. Reduzierter Speichelfluss ist ein Indikator für die Schädigung der Parotis. Folgende Werte konnten durch das Programm berechnet werden:

DATENSATZ	MODELL	TD_{50}	m
1 Monat nach Bestrahl.	Logistisch	32.1	0.14
1 Monat nach Bestrahl.	Gauss	32.3	0.16
1 Jahr nach Bestrahl.	Logistisch	33.4	0.26
1 Jahr nach Bestrahl.	Gauss	34.1	0.35

Wie zu erkennen ist, stimmen die Zahlen zwischen den beiden Modellen gut überein, welches den Angaben der Literatur entspricht, dass beide Modelle als gleichwertig anzusehen sind.

Ein weiterer interessanter Aspekt ist der Vergleich mit den Patientendaten, die ein Jahr nach der Bestrahlung gemessen wurden. Hierbei ist festzustellen, dass sich der „ TD_{50} “ zwar leicht erhöht hat, wobei aber die Betrachtung der Konfidenzintervalle keine Schlussfolgerung zulässt.

Daher konnte im gewählten Nachbeobachtungszeitraum anhand der vorliegenden Daten kein Erholungseffekt für die Parotiden nachgewiesen werden.

6. Weiterführende Anwendung

Mit Hilfe neuer Patientendatensätze wird in Zukunft die Wirksamkeit strahlenprotektiver Substanzen überprüft. Weiterhin könnte anhand des Programms mit Hilfe der Clusterzugehörigkeit eines Patientendatenvektors entschieden werden, ob ein strahlenprotektives Medikament eingesetzt werden sollte. Ausserdem ist eine Evaluation des bislang nicht in der Literatur enthaltenen organspezifischen Parameters „ n “ für die Parotis möglich. Denkbar ist weiterhin die Reduktion der Patientenvektoren mittels Hauptkomponentenanalyse (PCA) und anschliessendem Vergleich.

Literatur

- [1] <http://www.krebsinformation.de>
- [2] M. Menke: Biologische Modelle. In: Dreidimensionale Strahlentherapieplanung. 3D-Workshop '95. Hrsg.: W. Schlegel. Heidelberg: Selbstverlag, 1995
- [3] G. J. Kutcher, C. Burman, L. Brewster, M. Goitein, R. Mohan: Histogram reduction method for calculation complication probabilities for three-dimensional treatment planning evaluations, *Int J Radiat Oncol Biol Phys.* 1991 May 15;21(1):137-46
- [4] Judith M. Roesink et al.: Quantitative dose-volume response analysis of changes in parotid gland function after radiotherapy in the head-and-neck region, *Int J Radiat Oncol Biol Phys.* 2001
- [5] P. van Luijk T. C. Delvigne, C. Schilstra, J. M. Schippers: Estimation of parameters of dose-volume models and their confidence limits, *Phys Med Biol.* 2003 Jul 7;(13):1863-84
- [6] J. T. Lyman, A. B. Wolbarst: Optimization of radiation therapy, IV: A dose-volume histogram reduction algorithm, *Int J Radiat Oncol Biol Phys.* 1989 Aug;17(2):433-6
- [7] Mühlbach: Repetitorium der Wahrscheinlichkeitsrechnung und Statistik. Binomi Verlag, Springe 2002
- [8] <http://www.vs-c.de/vsengine/vlu/vsc/de/ch/7/tc/ps/grundlagen.ps.vlu.html>
- [9] S.A.Roberts, J.H.Henry: *Radiotherapy and Oncology* 20, 1993
- [10] C. Ullenboom: *Java ist auch eine Insel*. 1. Auflage, Gallileo Press GmbH, Bonn 2002
- [11] <http://java.sun.com/j2se/1.4.2/docs/guide/jps/index.html>

Möglichkeiten der Prävention der Zusendung unerwünschter Werbe-E-Mails

Tobias Eggendorfer

FernUniversität Hagen
Informatik
tobias.eggendorfer@fernuni-hagen.de

Art: Eigene, außeruniversitäre Untersuchung
Betreuer: Professor Dr. Christoph Beierle

Abstract: Ziel ist die Untersuchung von Verfahren, die verhindern, daß Versender von unerwünschten Werbe-E-Mails E-Mail-Adressen ihrer Opfer im Internet sammeln können. Dazu werden verschiedene Methoden vorgestellt und in einem Praxistest auf ihre Effektivität geprüft. Dabei wird auch eine proaktive Technik zur Blockade der E-Mail-Sammelprogramme untersucht.

1 Problem des Spammings

Versender von unerwünschten Werbe-E-Mails grasen auf der Suche nach E-Mail-Adressen mit sogenannten Harvestern automatisch Webseiten ab. Insofern arbeiten sie mit vergleichbaren Methoden wie die Spider von Suchmaschinen.

Tatsächlich sind auf Webseiten veröffentlichte E-Mail-Adressen hauptsächlich von Spam betroffen, das ergab eine Studie ([CDT03]). Ihr zufolge nimmt die Intensität von Spam wieder ab, wenn die Adresse von der Homepage entfernt wird. Damit scheint der erste Ansatz zur Vermeidung von Spam zu sein, seine E-Mail-Adresse nicht auf Webseiten zu veröffentlichen.

Allerdings ist das in der Praxis kaum möglich: Zahlreiche Gründe sprechen für eine Angabe von E-Mail-Adressen auf Webseiten. Nicht zuletzt werden Betreiber von geschäftlichen Webseiten in Deutschland durch §6 TDG gezwungen, eine Adresse der elektronischen Post anzugeben, wobei zu vermuten steht, daß der Gesetzgeber mit diesem umständlichen Ausdruck E-Mail-Adressen gemeint haben dürfte.

2 Verfahren zum Verdecken von E-Mail-Adressen

Vor diesem Hintergrund ist es notwendig, Verfahren zu finden, die es Harvestern unmöglich machen, E-Mail-Adressen auf Webseiten zu finden. Eine Auswahl möglicher Verfahren habe ich in [EGG04] (Übersetzungen [EGG04h], [EGG04g] und [EGG04f]) vorgestellt, weitere in [EGG04i]. Jedes dieser Verfahren hat spezifische Vor- und Nachteile: Manche setzen bestimmte Browserfunktionen, andere zusätzliche Software beim Besucher der Seite, häufig in Form eines Browser-Plug-Ins voraus. Einige erfüllen die zunehmend geforderte Bedingung der Barrierefreiheit nicht.

Gleichzeitig gilt es zu beurteilen, wie groß der Aufwand ist, den ein Spammer treiben müßte, um einen Harvester zu konstruieren, der in der Lage wäre, eine solchermaßen getarnte E-Mail-Adresse zu extrahieren. Dadurch lassen sich Vermutungen über die Zukunftssicherheit der Verfahren anstellen.

Eine Alternative wäre, über die Verwendung von robots.txt-Dateien, die den Zugriff auf einzelne Seiten für Spider verbieten, die Harvester aufzufordern, diese Seiten nicht zu indizieren. Allerdings wird dieser Standard von den Entwicklern von Harvestern beharrlich ignoriert, das bestätigt gerade wieder eine aktuelle Diskussion in news://de.admin.net-abuse.mail.

3 Proaktive Verfahren

Das führt zu der Idee, Harvestern eine Falle zu stellen und sie in einer Endlosschleife gefangen zu halten – möglichst so, daß der Harvester dabei langfristig blockiert wird. Eine solche Teergrube habe ich in [EGG04a] vorgestellt.

3.1SMTP-Teergruben

Andere Autoren ([DON04a], [REHWWa]) wollen Spammer lieber beim Versand der Werbemails blockieren und installieren SMTP-Teergruben. Damit soll erst der Versand behindert werden. Erfahrungsberichte der Betreiber solcher Teergruben zeigen, daß Spammer immer wieder in diese Fallen tappen.

Allerdings bezweifle ich die Nachhaltigkeit des Verfahrens: SMTP ermöglicht es, die gleiche E-Mail an zahlreiche Empfänger zu senden, indem lediglich zusätzliche Absenderadressen angegeben werden müssen. Der Verzögerungseffekt durch eine Teergrube ist also nur gering. Zudem wird zu einer Teergrube nur eine Verbindung gleichzeitig geöffnet, wodurch nur ein TCP-High-Port des Spammers blockiert wird und somit der Versand an andere Mailserver durch diesen Spammer nicht beeinträchtigt wird. Daher müßten im Netz sehr viele SMTP-Teergruben aktiv sein, um einen nachhaltigen Effekt zu erzielen.

3.2HTTP-Teergruben

Dagegen reicht es bei einer HTTP-Teergrube, die einen Harvester blockiert, theoretisch aus, nur eine im WWW erreichbar zu halten: Die von mir vorgeschlagene Teergrube erzeugt pro ausgegebener Seite zwanzig Links auf sich selbst. Damit wächst die Zahl der Links auf die Teergrube exponentiell mit der Zahl der Besuche durch den Harvester. Ein Harvester besucht typischer Weise jeden ausgegebenen Link. Nach endlicher Zeit ist damit davon auszugehen, daß jeder Zugriff des Harvesters in die Teergrube führen wird und er somit nicht mehr in der Lage ist, andere Seiten auf der Suche nach E-Mail-Adressen zu besuchen.

Den Spammern bleiben zwei Optionen, um dieser Falle zu entgehen: Entweder entwickeln sie ihre Harvester so, daß sie die robots.txt-Datei respektieren und umschiffen so die Falle, oder aber, sie setzen die Domains, unter denen eine solche Falle betrieben wird, auf eine Blacklist. Allerdings läßt sich die von mir vorgestellte Teergrube sehr schnell auf einem anderen Server unter einer anderen Adresse umziehen, theoretisch könnte sie auch an einer DSL-Leitung, die über einen DynDNS-Namen erreichbar ist, betrieben werden. Damit bleibt dem Spammer in dem Katz- und Mausspiel kaum keine Chance.

4 Auswertung

Derzeit läuft ein Versuch, um die Effektivität der Ideen praktisch zu überprüfen. Erste Ergebnisse werden im Rahmen der Veranstaltung präsentiert.

Literatur

- [AME04] von Ameln, Holger, Schopen, Marcus, Mit Open Source-Tools Spam und Viren bekämpfen, O'Reilly, Sebastopol, 2004
- [BAG04] Bager, Jo, Fischer von Molland, Michael, Wider die E-Mail-Massen. Neue Verfahren gegen Spam in c't, Heise, Hannover, 2004
- [BELL03] Bell, Keith, Obfuscating e-mail addresses, <http://www.december14.net/ways/js/nospam.shtml>, o. A., 2003
- [CDT03] Center for Democracy and Technology, Why am I getting all this spam?, -- ohne --, Washington, D.C., 2003
- [DON04] Donnerhacke, Lutz, Teergruben FAQ, www.iks-jena.de/mitarb/lutz/usenet/teergrube.html, Jena, 2004
- [DON04a] Donnerhacke, Lutz, Teergrubing Wrapper, www.iks-jena.de/mitarb/lutz/usenet/antispam.html, Jena, 2004
- [EGG04] Eggendorfer, Tobias, Privatadresse. Homepages spamsicher gestalten. in Linux User, Linux New Media, München, 2004
- [EGG04a] Eggendorfer, Tobias, Ernte - nein danke. E-Mail-Adressenjägern auf Webseiten eine Falle stellen in Linux Magazin, Linux New Media, München, 2004

- [EGG04f] Eggendorfer, Tobias, Adres zastrzezony. Projektowanie stron internetowych chronionych przed spamem, Linux New Media, München, 2004
- [EGG04g] Eggendorfer, Tobias, Private address. Design spamproof homepages, Linux New Media, München, 2004
- [EGG04h] Eggendorfer, Tobias, Fara spam - Tehnici prin care va puteti face adresa email invizibila pentru spidere, Linux New Media, München, 2004
- [EGG04i] Eggendorfer, Tobias, Spamfreie Homepage, Eigenverlag der Münchner Volkshochschule, München, 2004
- [GUN04] Gunton, Neil, Stopping Spambots: A Spambot Trap. Using Linux, Apache, mod_perl, Perl, MySQL, ipchains and Embperl, http://www.neilgunton.com/spambot_trap/, Ashland, 2004
- [HEM03] Hemenway, Kevin, Calishain, Tara, Spidering Hacks. 100 Industrial-Strength Tips & Tools, O'Reilly, Sebastopol, 2003
- [HIL04] Hilgefert, Ulrich, Mehr als 40 Prozent Spam-Mails in Deutschland, <http://www.heise.de/newsticker/meldung/48827>, Hannover, 2004
- [HONWWW] o. V., honeyd - Homepage, <http://www.honeyd.org/>, n/a, o. A.
- [ISEWWW] Issermann, Mario et. al., antispam.de, <http://www.antispam.de>, Wipperfurth, 2003
- [MÜN02] Münz, Stefan, HTML & Web-Publishing Handbuch (Band 1). HTML - JavaScript - CSS - DHTML, Franzis, Poing, 2002
- [MÜN02a] Münz, Stefan, HTML & Web-Publishing Handbuch (Band 2). XML - DTDs - Perl/CGI, Franzis, Poing, 2002
- [OUD03a] Oudot, Laurent, Fighting Spammers With Honeypots, <http://www.securityfocus.com/infocus/1747>, Calgary, 2003
- [PRO03] Provos, Niels, Honeyd Research: Honeypot against Spam, <http://www.honeyd.org/spam.php>, n/a, 2003
- [REHWWW] Rehbein, Daniel A., Adressensammler identifizieren - Ein Beispiel, <http://spamfang.rehbein.net>, Dortmund, o. A.
- [REHWWWa] Rehbein, Daniel, Gift für Harvester - Aus meiner Entwicklung, <http://www.daniel-rehbein.de/spamgift.html>, Dortmund, 2003
- [RFC2616] Fieldings R., et al., RFC2616: Hypertext Transfer Protocol - HTTP/1.1, <http://www.ietf.org/rfc/rfc2616.txt>, o. A., 1999
- [SCHWA02] Schwartz, Alan, Garfinkel, Simson, Stopping Spam, O'Reilly, Sebastopol, 2002
- [SCHWA04] Schwartz, Alan, SpamAssassin, O'Reilly, Sebastopol, 2004
- [TUCEK] Tucek, Jim, Email Protector Version 4.06. Download Email Protector, <http://www.jracademy.com/~jtucek/email/download.php>, o. A., o. A. (Stand: 3.10.2004)
- [W3CAPPB] W3C, W3C Recommendations. Appendix B: Performance, Implementation and Design, <http://w3.org/TR/REC-html40/appendix/notes.html>, - ohne Angabe -, o. A.
- [WHA01] Whalley, Carl, Spam-free mailto:tags. How to use mailto: functionality whilst preventing web crawlers from leeching your email addresses, <http://www.carmichael-data.com/art20011223.jsp>, Lancashire, 2001
- [WILLI02] Williams, Steve, Thwarting Spambots, <http://www.sbw.org/spambot/>, o. A., 2002

Studienprojekt Hafenlogistik: Erstellung einer verteilten Simulationsumgebung für ein teilautomatisiertes Containerterminal

Johannes Behrndt, Christopher Brandt, Jorge Homann, Carsten Keuch, Ingmar Stempel

Universität Hamburg
Institut für Wirtschaftsinformatik
johannes@behrndt.net
ch.brandt@xtopher.de
jorge-homann@web.de
carsten.keuch@hamburg.de
ingmarstempel@gmx.de

Art der Arbeit: Studienprojekt im Hauptstudium, Studiengang Wirtschaftsinformatik
Betreuer/in der Arbeit: Prof. Dr. Stefan Voß, Dr. Gabriele Schneiderei, René Eisenberg

Abstract: In diesem Studienprojekt wurde für ein (teil-)automatisiertes Containerterminal ein auf mehrere Rechner verteiltes Simulationsmodell konzipiert und erstellt. Dieses umfasst und visualisiert die logistischen Abläufe inklusive der Steuerung der Fahrzeuge und Krane, wobei auch die Planung der Containertransporte enthalten ist. Für die dabei auftretenden Planungsprobleme wurden verschiedene Algorithmen entwickelt und vergleichend untersucht.

1 Einleitung

„Sieben Millionen Container – diese Marke wird 2004 geknackt. Ein Plus von 15 Prozent. Höchster Zuwachs in Nordeuropa. Und der Boom hält an.“ [Za04], titelte das Hamburger Abendblatt in seiner Ausgabe vom 02.12.2004. In Folge dieses Trends wollen die Hamburger Containerterminalbetreiber ihre Kapazitäten durch Neu- und Umbau von Terminals sowie verbesserte Abläufe in den Terminals deutlich steigern. Die Entwicklung der letzten Jahre hat gezeigt, dass diese logistische Herausforderung nur mit massiver Unterstützung der Informationstechnik in Angriff genommen werden kann; vgl. [SVS04].

Ziel dieses Studienprojekts war, ein derartiges Containerterminal mit seinen Transportabläufen zu modellieren sowie verschiedene Strategien für die unterschiedlichen Planungsaufgaben zu entwickeln und zu analysieren. Daher bestand die Aufgabe für die Studierenden darin, ein auf mehrere Rechner verteiltes Simulationsmodell zu erstellen sowie Algorithmen zur Lösung der Planungsprobleme zu entwickeln und zu untersuchen.

2 Problembeschreibung

Das abzubildende Containerterminal besteht aus mehreren Lagerblöcken, einem Kaibereich zur Be- und Entladung der Containerschiffe sowie einem Bereich, in dem fahrerlose Transportfahrzeuge (so genannte AGVs; Automated Guided Vehicle) die Transporte zwischen dem Kaibereich und den Lagerblöcken durchführen. Innerhalb der einzelnen Lagerblöcke werden die Container jeweils mittels so genannter DRMGs (Double Rail-Mounted Gantry Crane) transportiert und ein- bzw. ausgelagert. Ein DRMG besteht immer aus zwei Einzelkränen (RMGs), die übereinander verfahrbar sind, so dass jeder der beiden Krane alle Stellplätze des entsprechenden Blocks bedienen kann. Darüber hinaus gibt es an den Lagerblöcken Übergabespuren für den landseitigen Verkehr (LKW und Bahn).

Auf einem derartigen Terminal fallen mehrere Typen von Transporten an:

- wasserseitige Transporte zwischen einem Schiff und einem Lagerblock, wobei ein RMG im entsprechenden Block sowie ein AGV benötigt werden,
- landseitige Transporte zwischen einer landseitigen Übergabespur (LKW oder Bahn) an einem Lagerblock und einem Stellplatz innerhalb dieses Blocks, wobei nur ein RMG benötigt wird,
- Umstapelungen von Containern innerhalb eines Blocks, z.B. um an einen darunter liegenden Container zu gelangen.

Für diese Transporte müssen extern vorgegebene Termine am Schiff sowie an den landseitigen Übergabespuren möglichst eingehalten werden. Um dies zu gewährleisten, werden passende Termine für die einzelnen Terminalbereiche ermittelt, wodurch eine terminalweite Synchronisation erreicht werden soll. Für die operationale Planung der Transportaufträge in den einzelnen Bereichen ergeben sich Reihenfolge- und Zuordnungsprobleme mit dem Ziel Terminabweichungen zu minimieren.

3 Inhalt des Projekts

Die AGVs ebenso wie die DRMGs sind (teil-)automatisiert, weshalb dieser Bereich das Kernstück des Simulationsmodells darstellt. Die im Rahmen des Projekts erstellte Simulationssoftware besteht daher aus drei Teilen: dem AGV-Bereich, dem DRMG-Bereich und einem Vorgangsmanager.

Die Studierenden haben zuerst in kleinen Gruppen (2-4 Personen) ein stabiles verteiltes Simulationsframework erarbeitet und implementiert, wobei jede Gruppe für einen der drei Teilbereiche AGV, DRMG bzw. Vorgangsmanager ein Konzept erstellt, präsentiert und implementiert hat. Zusätzlich wurden die Kommunikation und Schnittstellen zu den anderen Teilbereichen entworfen und implementiert.

Für den AGV-Teil haben die Studierenden ein Wegenetz für die AGVs, eine Routenplanung und Ausweichregeln für die Fahrzeuge entworfen und implementiert. Zusätzlich wurde eine Deadlock-Erkennung und -Behebung mittels Wegegraphen integriert. Der gesamte Bereich wurde darüber hinaus so visualisiert, dass alle Fahrzeuge mit ihrer Beladung und den Fahrtrouten sichtbar sind.

Der DRMG-Bereich beinhaltet analog dazu die Bewegungen der DRMGs inklusive der Behandlung von möglichen Blockierungen zwischen den beiden Kranen eines DRMGs. Analog zum AGV-Bereich wurde eine Visualisierung jedes Lagerblocks erstellt, die auch die einzelnen Containerstellplätze im Lager umfasst.

Der Vorgangsmanager beinhaltet folgende Funktionen: das Einlesen bzw. Generieren von Auftragsszenarien und Anfangsbelegungen der Lagerblöcke, das Initiieren einzelner Aufträge am Kai bzw. an der Landseite, die gesamte Kommunikation zwischen den Software-Teilen und den beteiligten Rechnern. Zusätzlich wurde die Datenbank MySQL benutzt, um Aufträge und Containerstandorte zu speichern. Darüber hinaus wurden Statistikmodule hinzugefügt, ein abschließendes Refactoring vorgenommen und eine Vereinheitlichung der drei Teile (insbesondere hinsichtlich Aufbau der Programm-Module) für eine bessere künftige Erweiterbarkeit durchgeführt.

In der Terminalsimulation mussten folgende Planungsprobleme gelöst werden: Zuordnung von Aufträgen zu AGVs bzw. RMGs und eine Reihenfolgeplanung der Aufträge auf den Geräten sowie die Ermittlung von geeigneten Lagerblöcken und Stellplätzen für einzulagernde und umzustapelnde Container. Hierfür wurden jeweils diverse Online-Algorithmen [Gu02] auf Basis von Prioritätsregelverfahren, Genetischen Algorithmen, Simulated Annealing und Branch-and-Bound erstellt; vgl. [Ca91, DLP02, Go02, Ho75, KGV83, Sch00, WR99]. Diese wurden anschließend vergleichend untersucht, um eine Aussage über die relative Leistungsfähigkeit der einzelnen Verfahren treffen zu können.

Das Simulationsmodell läuft verteilt auf mehreren PCs, wobei die Kommunikation mittels RMI [Su03] erfolgt. Die Software wurde ausschließlich von den Studierenden selbstständig in Java implementiert ohne bereits bestehende Module oder Dritt-Anbieter-Bibliotheken zu nutzen. Die Software ist dabei so modularisiert, dass einzelne Komponenten (z.B. die Planung) leicht austauschbar sind.

4 Fazit und Ausblick

Zur Bearbeitung dieser Aufgabe sind Kenntnisse und Fähigkeiten aus unterschiedlichen Bereichen nötig gewesen, z.B. Logistik, Operations Research (Online-Algorithmen) und Informatik (Software-Entwurf und -Implementierung). Des Weiteren war aufgrund der Aufgabenteilung ein hohes Maß an Teamfähigkeit und Selbstorganisation wichtig für den Erfolg des Studienprojekts. Für Studierende der Wirtschaftsinformatik stellte diese Vielseitigkeit und die Praxisnähe der Problemstellung eine gute Gelegenheit dar, die im Studium erworbenen theoretischen Kenntnisse praktisch einzusetzen. Derzeit werden zum einen die Software auf ereignisorientierte Simulation und zum anderen die Planungskomponenten mit den Algorithmen auf nebenläufige Planung umgestellt.

5 Literaturverzeichnis

- [Ca91] Cao, B.: Operations Research in der Container-Stellplatz-Verwaltung. Arbeitspapier, Universität der Bundeswehr, Hamburg, 1991.
- [DLP02] Dubrovsky, O.; Levitin, G.; Penn, M.: A Genetic Algorithm with a Compact Solution Encoding for the Container Ship Stowage Problem. *Journal of Heuristics* 8, 2002, S. 585-599.
- [Go02] Goldberg, D.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Boston, 2002.
- [Gu02] Gutenschwager, K.: *Online-Dispositionsprobleme in der Lagerlogistik*. Physica, Heidelberg, 2002.
- [Ho75] Holland, J. H.: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Michigan, 1975.
- [KGV83] Kirkpatrick, S.; Gelatt, C.; Vecchi, M.: Optimization by Simulated Annealing. *Science* 220, 1983, S. 671-680.
- [Sch00] Schrecker, A.: *Planung und Steuerung fahrerloser Transportsysteme*. Gabler, Wiesbaden, 2000.
- [SVS04] Steenken, D.; Voß, S.; Stahlbock, R.: Container Terminal Operations and Operations Research – A Classification and Literature Review. *OR Spectrum* 26, 2004, S. 3-49.
- [Su03] Sun Microsystems Inc.: Java™ Remote Method Invocation (RMI). 2003, <http://java.sun.com/j2se/1.4.2/docs/guide/rmi/>
- [WR99] Wilson, I.D.; Roach, P.A.: Principles of Combinatorial Optimization Applied to Container-Ship Stowage Planning. *Journal of Heuristics* 5, 1999, S. 403-418
- [Za04] Zamponi, R.: Rekord im Hamburger Hafen. *Hamburger Abendblatt*, 02.12.2004, <http://www.abendblatt.de/daten/2004/12/02/371389.html>

Controlling im Hilfsmittelbereich¹

Regine Wolters
wolters@grafikprogrammierung.de
Fachhochschule Dortmund
Fachbereich Informatik

Betreuer der Arbeit: Prof. Dr. Klaus Zeppenfeld
Art der Arbeit: Bachelor-Thesis
GI-Fachbereich: Wirtschaftsinformatik

Zusammenfassung

Die immer knapper werdenden finanziellen Mittel der Gesellschaft setzen auch Verbände, öffentliche Unternehmen und andere Non-Profit-Organisationen (NPOs) unter Druck. Über das aus gewinnorientierten Unternehmen bekannte Finanzcontrolling gehen die Anforderungen an ein Controlling in NPOs weit hinaus. Hier wird vielmehr ein Fachcontrolling notwendig, das in den fachlichen Besonderheiten der NPOs begründet ist und eine wesentlich individuellere und differenziertere Gestaltung als das Finanzcontrolling verlangt. Die Arbeit entstand im Anschluss an ein Praxissemester bei der sd&m AG in Kooperation mit der AOK Systems GmbH und der FH Dortmund.

1. Controlling

Das aus dem Englischen stammende Verb „to control“ besitzt im deutschen Sprachgebrauch weitaus mehr Bedeutungen als „kontrollieren“: Es ist vielmehr zu übersetzen mit einer Reihe von Begriffen wie „steuern“, „lenken“, „führen“, „beeinflussen“ und „im Griff haben“. Ebenso vielfältig wie die Übersetzungen des Verbs sind die in der Literatur angeführten Definitionen des Begriffs Controlling. Daher sei an dieser Stelle nur die Definition nach Horváth zitiert: „Controlling ist dasjenige Subsystem der Führung, das Planung und Kontrolle sowie Informationsversorgung systembildend und systemkoppelnd ergebniszielorientiert koordiniert und so die Adaption und Koordination des Gesamtsystems unterstützt.“ [HOR01]

Um die Vielschichtigkeit des Begriffs „to control“ zu erfassen, unterscheidet die klassische Auffassung des Controllings vier Kernaufgaben, von denen die eigentliche Kontrolle nur einen Aspekt darstellt:

¹ Der Vortrag wurde auf den Informatiktagen 2003 gehalten

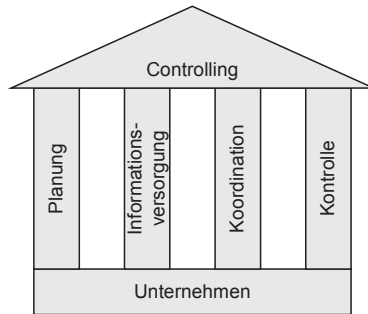


Abb. 1: Kernaufgaben des Controllings

Während es die Aufgabe der Planung ist, realistische Controlling-Ziele zu formulieren und diese schriftlich zu fixieren, muss die Informationsversorgung und den Betrieb der Informationsversorgungssysteme eines Unternehmens sicherstellen. Die Koordination beschreibt eine zielorientierte Abstimmung verschiedener Handlungen aufeinander. Innerhalb des Controllings sollen Kontrollen Planungs- und Ausführungsfehler sowie Störeinflüsse der Umwelt und Unvollkommenheit der Informationen aufdecken.

Das Controlling als Unternehmensaufgabe kann in zwei Teilbereiche eingeteilt werden: das strategische und das operative Controlling. Das strategische Controlling unterstützt das Führungssystem in seiner Vorgehensweise, um die Effektivität der Unternehmungen zu erhöhen, während das operative Controlling die Vorgehensweise bezüglich der Rentabilität und Effizienz der Prozesse überprüft, wobei die einzelnen Ziele nicht hinterfragt werden.

Die wichtigsten Instrumente des Controllings sind Analysen, wie z.B. die ABC-Analyse, die Balanced Scorecard, die Deckungsbeitragsrechnung und die Kosten- und Leistungsrechnung. Die Ergebnisse der einzelnen Analysen führen zu Daten und Informationen, die z.B. in Form von Berichten dargestellt werden können. Auf diese Weise können die relevanten Informationen an das Management weitergegeben werden, wobei der Umfang des Berichtswesens in direkter Abhängigkeit zu Größe und Komplexität des Unternehmens steht. Inhalt der Analysen sind Kennzahlen, die numerische Informationen zur Planung, Kontrolle und Steuerung des Unternehmens enthalten. Sie geben relevante Zusammenhänge in verdichteter und quantitativ messbarer Form wieder.

1.1. Controlling in Non-Profit-Organisationen

Alle NPOs werden dem nicht-erwerbswirtschaftlichen Sektor zugeordnet, da sie nicht gewinnorientiert, sondern bedarfsdeckend arbeiten. Dabei wird häufig ein sozial oder gesellschaftlich relevantes Gut zur Verfügung gestellt, welches in der Regel nicht auf dem freien Markt zu erwerben ist, wie z.B. eine qualitativ gleichwertige medizinische Versorgung nach dem Solidaritätsprinzip oder gesundheitliche Prävention. NPOs lassen sich anhand folgender grundsätzlicher Merkmale charakterisieren:

- Die Ansprüche der Mitglieder müssen bedarfsgerecht unter Berücksichtigung von Effizienz und Gerechtigkeit gedeckt werden.
- Häufig muss das notwendige Kapital selbst aufgebracht werden (Beiträge, Spenden, Sponsoring, ...).
- Knappe Ressourcen erfordern eine sorgfältige Planung und einen sparsamen Umgang.
- Erwirtschaftete Gewinne kommen ausschließlich der Organisation selbst zu Gute; es erfolgt keine Ausschüttung von Überschüssen an Kapitalgeber oder Eigentümer.
- Da NPOs nicht auf den Verkauf ihrer Leistungen bzw. Produkte angewiesen sind, weisen sie eine hohe Konjunkturresistenz auf.

Gemäß des Grades der Amtlichkeit, der Finanzierung, des Grades der Selbstständigkeit und der Möglichkeit zur Privatisierung lassen sich NPOs in öffentliche, verselbstständigt öffentliche, staatsergänzende, konventionell gemeinnützige und alternative NPOs unterscheiden.

1.2. Controlling in der AOK

Aufgrund dieser Einteilung wird die AOK als gesetzlicher Krankenversicherer den verselbstständigten öffentlichen NPOs zugeschrieben. In ihrem wirtschaftlichen Handeln ist die AOK durch gesetzliche Vorgaben eingeschränkt:

§ 70 Qualität, Humanität und Wirtschaftlichkeit

(1) Die Krankenkassen und die Leistungserbringer haben eine bedarfsgerechte und gleichmäßige, dem allgemein anerkannten Stand der medizinischen Erkenntnisse entsprechende Versorgung der Versicherten zu gewährleisten. Die Versorgung der Versicherten muss ausreichend und zweckmäßig sein, darf das Maß des Notwendigen nicht überschreiten und muss in der fachlich gebotenen Qualität sowie wirtschaftlich erbracht werden.

Der Paragraph 70 des 5. Sozialgesetzbuches [SGB_5] fordert die gesetzlichen Krankenkassen zu einem effektiven und effizienten Handeln auf. Um diesen Anforderungen gerecht zu werden, hat die AOK Gesundheitsziele auf drei verschiedenen Ebenen definiert:

- Gesundheitsoberziele, die als prioritäre Gesundheitsziele z.B. anhand von Prävalenz, Inzidenz und Kosten für konkrete Erkrankungen vereinbart werden.
- Gesundheitsunterziele, die als konkrete, aus den Oberzielen abgeleitete Forderungen definiert sind.
- Gesundheitsindikatoren, die die Forderungen der Unterziele operational umsetzen.

Durch die Gesundheitsindikatoren können aussagekräftige Messwerte festgelegt werden, durch welche die faktische Umsetzung der Forderungen kontrolliert werden kann.

2. Die Controlling-Anwendung

Die Controlling-Anwendung basiert auf dem operativen System, das im Leistungsmanagement der AOK zur Unterstützung der Genehmigung und Verwaltung von Hilfsmitteln eingesetzt wird. Durch das Auswerten von Leistungsdaten, produktspezifischen Daten, Fall-Daten und Service-Daten soll die Anwendung ermöglichen,

- Einsparpotentiale zu erkennen,
- Aussagen über die Wirtschaftlichkeit des Hilfsmittelpools zu treffen und
- Statistiken mit fest definierten Kennzahlen und benutzerdefinierten Filtern zu erstellen.

Die für das Controlling-System bereitgestellten Daten werden aus dem operativen System extrahiert und durch wöchentlich und monatlich ablaufendes Nachladen durch spezielle Datenbank-Jobs regelmäßig aktualisiert. Zur Speicherung der Daten werden flache Tabellen in einem relationalen Datenbanksystem verwendet. Anschließend werden diese Basistabellen zu so genannten Ereignistabellen aggregiert, wobei die Aggregation unter Berücksichtigung verschiedener Dimensionen wie Hilfsmittel, Zeiträume usw. erfolgt.

Die durch den Benutzer erstellten Berichte können entweder sofort ausgedruckt oder zur Weiterverarbeitung nach MS Excel exportiert werden.

Die technische Architektur von der Anwendung lässt sich somit wie folgt darstellen:

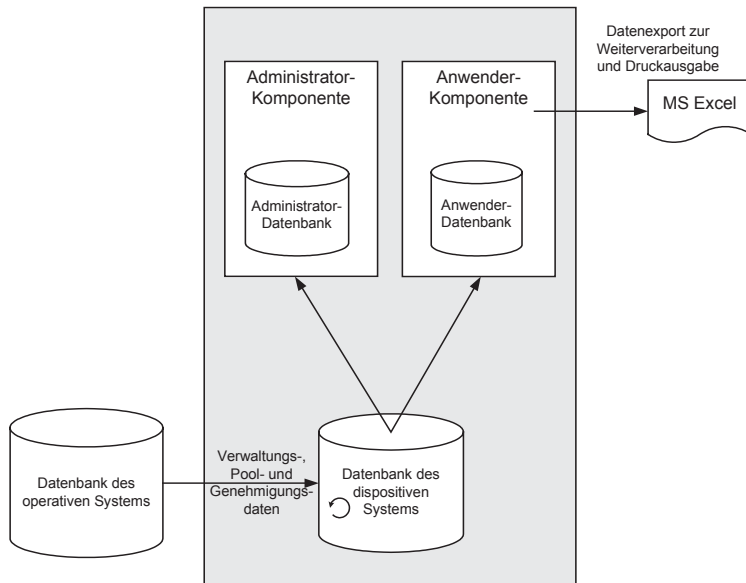


Abb. 2: Technische Architektur [SYS02]

2.1. Re-Design und Implementierung eines Moduls zur Marktbeobachtung

Neben gezielten Analysen der Service-Daten ist die Marktbeobachtung ein wichtiges Kernstück der Anwendung. Ganz allgemein versteht man unter einer Marktbeobachtung eine zeitraumbezogene Analyse, die die Veränderungen des Marktes feststellen soll: „Die Marktbeobachtung verzichtet auf eine stichtagbezogene Betrachtung, sie ist eine Zeitrumbetrachtung, eine kontinuierliche Untersuchung des Marktes. Sie registriert das Geschehen auf dem Markt, seine Veränderungen, versucht sie zu begründen und offensichtliche Gesetzmäßigkeiten in ihrem Verlauf festzustellen.“ [EHR99]

Die Untersuchung der bisher im Rahmen der Marktbeobachtung angebotenen Analysen ergab, dass die vorhandenen Auswertungsmöglichkeiten nicht für eine multidimensionale Untersuchung aller beeinflussenden Faktoren ausreichen und zudem die Benutzerführung nicht intuitiv war. Daraus lässt sich für das Re-Design und die Implementierung der Marktbeobachtung folgendes Projektziel ableiten:

„Mit der Neustrukturierung sollen die für das **umfassende Controlling** des Hilfsmittelmanagements **erforderlichen Strukturen** geschaffen werden. Die vorhandenen Sourcen sollen dazu unter **Beibehaltung der Funktionalität** so optimiert werden, dass zum einen eine Reduzierung der Anzahl der Formulare und Maskenübergänge und zum anderen eine Verbesserung der Wartbarkeit erreicht werden kann.“ [SYS02]

2.1.1. OLAP – Technik zur mehrdimensionalen Datendarstellung

Die Technik, mehrdimensionale Abfragen auf Basis eines Datenbestandes auszuführen, wird im Controlling mit dem Begriff OLAP bezeichnet. Diese Abkürzung steht für „On-Line Analytical Processing“ und hat das Ziel, neue oder unerwartete Beziehungen zwischen einzelnen Dimensionen aufzudecken. Die wörtliche Übersetzung „dynamische Analyse-Prozesse“ deckt aber nur einen Teil der im OLAP gestellten Anforderungen ab. E. F. Codd, der Erfinder des relationalen Datenmodells, definiert zwölf Anforderungen, die eine OLAP-Anwendung erfüllen muss. Zu den wichtigsten dieser Forderungen gehören Multidimensionalität, Transparenz, Client/Server-Architektur, Mehrbenutzerfähigkeit, dynamische Behandlung dünn besetzter Matrizen, intuitive Datenverarbeitung und ein flexibles Berichtswesen. (vgl. [HYP96])

Voraussetzung für die Anwendung der OLAP-Technik ist die Festlegung der Dimensionen, welche die Analyse bestimmen. Innerhalb dieser Dimensionen müssen die Objekte hierarchisch strukturiert sein. Besitzt das Datenmodell drei Dimensionen, so ergibt sich ein Datenwürfel. Mehr als drei Dimensionen sind graphisch nicht darstellbar und werden bei der Umsetzung auf Oberflächen des Programms z.B. durch übereinanderliegende Schieberegler, Listboxen oder Auswahlfelder dargestellt.

Für die Arbeit mit den Daten des Würfels werden folgende wichtige Operationen in der OLAP-Technik definiert:

- **Pivotieren/ Rotieren:** Drehen des Datenwürfels durch Tausch der Dimensionen.
- **Slicing:** „Schnitt“ durch den Datenwürfel, der die Dimensionalität um mindestens eine Dimension verringert.
- **Dicing:** Herauslösen einzelner beliebig großer Würfelemente als Teilausschnitte aus dem Gesamtwürfel. Die Dimensionalität der ursprünglichen Abfragen bleibt erhalten, jedoch wird die Hierarchiestufe einer Dimension detaillierter dargestellt.

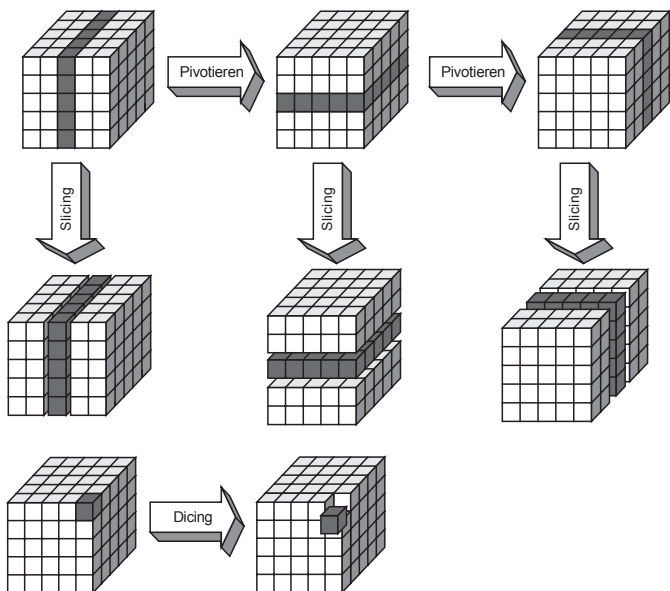


Abb. 3: Pivotieren, Slicing und Dicing

- **Drill-Down:** Entlang der Hierarchie der einzelnen Dimensionen kann von aggregierten Daten zu Detail-Daten navigiert werden.
- **Roll-Up:** Gegenteil des Drill-Down; entlang des Konsolidierungspfades werden die Detaildaten zu neuen Daten aggregiert.

Die Anwendung der OLAP-Technik erlaubt dem Anwender einen interaktiven Zugriff, auf dessen Grundlage er sich durch eine Vielzahl verschiedener Betrachtungsweisen einen Überblick über die Daten verschaffen kann. Die Anwendung der Technik ist dabei nicht zwingenderweise an die Verwendung einer multidimensionalen Datenbank gebunden. Vielmehr ist es möglich, OLAP auch auf Basis einer relationalen Datenbank zu implementieren (ROLAP = relationales OLAP). ROLAP-Systeme benutzen als Abfragesprache SQL (SQL = Standard Query Language), wobei die erforderlichen Abfragen zur Laufzeit des Programms aus den Eingaben des Benutzers generiert werden. Somit können ROLAP-Systeme die flexiblen Operationen der Datenanalyse mit der Datenspeicherung eines relationalen Datenbanksystems kombinieren.

2.1.2. Anwendung der OLAP-Technik in der Marktbeobachtung

Die Controlling-Anwendung kann technisch als eine Client/Server-ROLAP-Anwendung mit entfernter Datenhaltung betrachtet werden, die die erweiterten Aggregatfunktionen von SQL nutzt (z.B. GROUP BY). Die über die Eingabeoberfläche festgelegten Auswahlmöglichkeiten werden beim Start einer Abfrage überprüft und aus diesen wird dynamisch ein SQL-Statement generiert. Dieses Statement wird an die Oracle-Datenbank abgesetzt und das Abfrageergebnis wird dem Benutzer in Form einer Tabelle angezeigt.

Nach dem Re-Design ist es in der Marktbeobachtung möglich, Analysen durchzuführen, die folgende Dimensionen berücksichtigen: Verordner, Hilfsmittel, Leistungserbringer, Zeit, Versorgungsform und Kostenträger.

Marktbeobachtung

Die TOP: Hilfsmittel Leistungserbringer Verordner

Pos.-Ebene: Produkt (Krankenfahrzeuge) | Zul.-Gruppe: <Alle>

VP-Art: <Alle> | Fachrichtung: <Alle>

Hitliste: <Temporär> | Sortierung: Antragsvolumen (EUR) - Aufsteigend

Zeiteinheit: Jahr 2001 | Kriterien: Antragsvolumen (EUR)>100

Hilfsmittel: [] | Versorgung: Neuversorgung (poolverw.)

Leistungserbringer: [694] [694 Sonstige Erbringer von Lei.stu.] | KV/PV: KV

Verordner: [2052] [2052 Allgemeine Ärzte] |

Lfd. Nr	Positionsnummer	Bezeichnung	Antragsfälle	Antragsvolumen (EUR)	Genehmigungsfälle	Genehmigungsvolumen (EUR)	Genehmigungsvolumen je Fall (EUR)
1	1846020005	Toilettenrollstuhl Mod. 2.176-99-02	1	166,17	1	166,17	166,17
2	1850022058	Faltrollstuhl Format 2 Vario 1.940-819	4	2.760,81	4	2.760,81	690,20
3	1850030044	Aktivrollstuhl Easy 300 Nr. 763 900 000 Code Nr. 150	1	2.922,46	1	2.922,46	2.922,46
4	1850030109	Rollstuhl K4	1	3.214,90	1	3.214,90	3.214,90
5	1846050009	Elektronrollstuhl Derby 2.433 mit Ladegerät	1	4.511,19	1	4.511,19	4.511,19

Abb. 4: Multidimensionale Hitliste der Marktbeobachtung

2.1.3. Fazit

Mit der Neustrukturierung der Marktbeobachtung konnten alle gestellten Ziele erreicht werden:

Das geforderte Ziel der Strukturverbesserung wurde durch die Reduktion der Formularanzahl und die Beschränkung der Wechselmöglichkeiten erreicht. Trotz dieser Einschränkungen werden die fachlichen Funktionalitäten beibehalten und dem Benutzer zur Verfügung gestellt.

Durch die multidimensionale Hitliste können einfache Ursachenanalysen innerhalb einer Dimension sowie mehrerer Dimensionen gleichzeitig durchgeführt werden. Die Einschränkungsalternativen, die sich aus den Hierarchiestufen der einzelnen Dimensionen ergeben, erlauben das Pivotieren, Slicing und Dicing im Sinne der OLAP-Technik. Somit ergibt sich für das Controlling vor allem im Bereich der Koordination ein deutlicher Mehrwert, da nun neue oder unerwartete Beziehungen zwischen einzelnen Dimensionen aufgedeckt werden können.

Literatur

- [EHR99] Ehrmann, H.: Marketing-Controlling. Friedrich Kiehl Verlag, Ludwigshafen, 1999.
- [HOR01] Horváth, P.: Controlling. Verlag Franz Vahlen, München, 2001.
- [HYP96] E. F. Codd Associates: Providing OLAP to User-Analysts: An IT Mandate, Whitepaper der Hyperion Solutions Corporation, 1996.
- [SGB_5] Sozialgesetzbuch, AOK Bundesverband, Stand 1. August 2003
- [SYS02] AOK Systems: Konzept zur Neustrukturierung, AOK Systems, Troisdorf, 2002.

Einsatz verteilter Hash-Tabellen in mobilen Ad-Hoc-Netzen

Tobias Heer

Wilhelm-Schickard-Institut für Informatik
Universität Tübingen
heer@informatik.uni-tuebingen.de

Art der Arbeit: Studienarbeit

Betreuer der Arbeit: Dr. Klaus Wehrle, Stefan Götz, Simon Rieche

Abstract: Verteilte Hash-Tabellen eröffnen durch die Eigenschaften der Skalierbarkeit, Flexibilität und Zuverlässigkeit hervorragende Möglichkeiten zur dezentralen Koordination und strukturierten Verwaltung großer Datenmengen in verteilten Systemen. Im Bereich des infrastrukturellen Internets erreichen sie ein hohes Maß an Stabilität und Performanz. Insbesondere dieser Eigenschaften bedarf es auch im Bereich der spontanen Vernetzung drahtlos angebundener, mobiler Geräte (Ad-Hoc-Netze), um in verschiedensten Einsatzszenarien schnell und flexibel leistungsfähige Dienste anbieten zu können. In dieser Arbeit wird untersucht, ob sich die grundsätzlichen Annahmen verteilter Hash-Tabellen auf Ad-Hoc-Netzwerke abbilden lassen und in wie fern sich aus der Kombination beider Technologien entscheidende Synergieeffekte ergeben.

1 Einleitung

Peer-to-Peer (*P2P*) Netze [1] haben sich in den letzten Jahren rasant entwickelt. Zu ihren Vertretern gehören Anwendungen wie Skype oder Gnutella, die inzwischen eine enorme Popularität erreicht haben. Die selbstorganisierenden P2P Netze eignen sich besonders für die Verwaltung von verteilt gespeicherten Informationen und stellen in vielen Bereichen eine Alternative zu einer zentralen Serverlösung dar. P2P Netze können in strukturierte und unstrukturierte Netzwerke gegliedert werden. Unstrukturierte P2P Netze haben einen geringen organisatorischen Overhead, jedoch ist zum Finden einer Information ein hoher Aufwand nötig. Strukturierte P2P Netze können aufgrund der ihnen zugrunde liegenden strukturierten Topologie effizient auf Informationen zugreifen, jedoch muss diese zuerst aufgebaut und erhalten werden. Dies führt zu einem erhöhten Organisationsaufwand.

Hash-Tabellen sind weit verbreitete Datenstrukturen, die in vielen Bereichen der Informatik schnelles Suchen und gezielten Zugriff auf Schlüssel-Wert-Paare ermöglichen. Verteilte Hash-Tabellen (*Distributed Hash Tables*, *DHTs*) sind verteilte Datenstrukturen, die dieses Verfahren auf ein strukturiertes P2P Netz übertragen. Dabei werden Teile der Hash-Tabelle auf verschiedene Knoten im Netz aufgeteilt. Die eigentliche Herausforderung bei der Suche nach einer Information besteht darin, den Knoten, auf dem sich die Information befindet, möglichst schnell kontaktieren zu können, obwohl jeder Knoten nur

wenige andere Knoten kennt. Um dies zu erreichen werden die Verbindungen zwischen den einzelnen Knoten anhand eines speziellen Systems aufgebaut. In [2] werden einige Topologien vorgestellt, die einen Zugriff mit logarithmischem Aufwand erlauben. Sie sind darüber hinaus robust gegenüber Knotenausfällen und gut skalierbar.

Aufgrund der zunehmenden Verbreitung von mobilen Geräten mit drahtlosen Kommunikationsmöglichkeiten gewinnen mobile Ad-Hoc-Netze (*MANETs*) [3] zunehmend an Bedeutung. Besonders in Situationen wie Polizeieinsätzen, Business-Meetings, Katastropheneinsätzen oder in Unfallsituationen, in denen keine vorhandene Kommunikationsinfrastruktur gegeben ist, sind MANETs oftmals die einzige Möglichkeit zur Kommunikation. Um auch in solchen Situationen effektiv auf verteilt erfasste oder gespeicherte Informationen zugreifen zu können, bietet sich der Einsatz einer P2P Lösung an.

2 Verteilte Hash-Tabellen in mobilen Ad-Hoc-Netzen

Wir möchten in dieser Arbeit die Möglichkeit des Einsatzes von DHTs in mobilen Ad-Hoc-Netzen untersuchen und feststellen, welche Schwierigkeit sich durch den Wechsel des zugrunde liegenden Netzwerks ergeben. Diese Arbeit soll Schwierigkeiten beim Einsatz von DHTs in MANETs aufzeigen und Lösungsansätze geben, um eine Kombination beider Technologien möglich zu machen.

2.1 Infrastrukturelle Netze und Ad-Hoc Netze

Da sich mobile Ad-Hoc-Netze stark von infrastrukturellen Netzen unterscheiden, gelten bestimmte Annahmen, die bei der Entwicklung von DHTs gemacht wurden, in MANETs nicht mehr. Im Gegensatz zum Internet, in dem eine feste und robuste Topologie vorausgesetzt werden kann, hängt die Konnektivität der Knoten in MANETs stark von Knotenbewegungen, Umweltfaktoren und Energiereserven ab. Da MANETs über keine dedizierte Infrastruktur verfügen und jedes Endgerät deshalb auch ein Router ist, verändert sich die Netztopologie sehr viel stärker als dies im Internet üblich ist. Als Folge dessen ändern sich Kontaktmöglichkeiten, Hop-Distanzen und Übertragungszeiten häufig und im großen Maß. Dies kann, wie in Abb. 1 zu sehen ist, sogar zu einer Partitionierung oder einem Zusammenschluss von Netzwerken führen, wobei viele Knoten gleichzeitig unerreichbar bzw. erreichbar werden. DHT-Protokolle können zwar mit Topologieveränderungen in gewissem Rahmen umgehen, jedoch wurde diese Anpassungsfähigkeit für infrastrukturelle Netze ausgelegt. Für den Einsatz in MANETs ist es daher notwendig zusätzliche Mechanismen zu integrieren und bestehende Fähigkeiten für eine schnelle Anpassung zu optimieren.

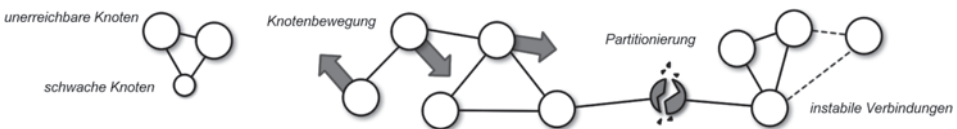


Abb. 1 Gegebenheiten in mobilen Ad-Hoc-Netzen

2.2 Wahl einer verteilten Hash-Tabelle für mobile Ad-Hoc-Netze

Wie in [4] gezeigt wurde, haben die verschiedenen DHT Topologien unterschiedliche Eigenschaften bezüglich Robustheit und Anpassungsfähigkeit. Darüber hinaus entscheidet die Topologie, wie viel Spielraum für Erweiterungen und Anpassungen bleibt. Durch Erweiterungen wie Proximity Neighbor Selection und Proximity Route Selection lässt sich z.B. eine bessere Anpassung an die Topologie des Trägernetzwerkes erreichen, wodurch Verbindungszeiten und Gesamtsendevolumen verringert werden können. Diese Konzepte lassen sich jedoch nicht in allen Topologien implementieren. Wir haben uns aufgrund seiner hohen Robustheit und guten Erweiterbarkeit für das DHT-Protokoll Chord [5] als Basis für weitere Betrachtungen und eine Evaluierung durch Simulation entschieden. Es basiert auf einer Topologie, die alle Knoten zu einem logischen Ring anordnet.

2.3 Partitionierung und Vereinigung

Durch Partitionierungen und Zusammenschlüsse von MANETs kann es zur Aufspaltung einer DHT in mehrere Teile kommen. Mit Hilfe von redundanten Verbindungen und der Wahl einer robusten DHT kann dafür gesorgt werden, dass diese selbst nach einer Partitionierung des Netzwerkes weiterhin operabel bleibt. Es ergibt sich daraus jedoch das Problem, dass es, durch eine Wiedervereinigung der MANETs, zu einem Aufeinandertreffen mehrerer eigenständiger DHTs kommen kann. Da dieses Problem in infrastrukturellen Netzen sehr selten ist, haben DHT-Protokolle keine Mechanismen um mit einer solchen Situation umzugehen.

Zur Erkennung eines Aufeinandertreffens mehrerer DHTs muss eine Identifikationsmöglichkeit geschaffen werden, bei der jeder DHT-Instanz ein eindeutiger Name zugewiesen wird. Welche Knoten an der Namensgebung beteiligt sind kann anhand der Position in der DHT festgelegt werden. Eine verteilte Namensgebung durch mehrere Knoten verhindert dabei Manipulations- und Störversuche. Zur Fusionierung zweier DHTs werden zuerst Knoten von der kleineren in die größere DHT verschoben. Die Knoten behalten dabei ihre Position, um möglichst wenige Informationen mit anderen Knoten austauschen zu müssen. Im günstigsten Fall muss dabei ein Knoten nur die Verbindungen entlang des Chord-Rings neu aufbauen.

2.4 Service-Discovery und Beitritt zur DHT

In infrastrukturellen Netzen wird zur Kontaktaufnahme mit einem P2P Netz meist ein festgelegter bekannter Knoten kontaktiert. Da in MANETs keine globale Erreichbarkeit gegeben ist, müssen die Knoten über Mechanismen zum Finden einer vorhandenen DHT verfügen, was zum Beispiel über ein beschränktes Fluten des Netzes möglich ist. Werden mehrere DHTs gefunden, entscheidet sich der Knoten anhand deren Größe und Konnektivität für eines der verfügbaren DHT-Netze.

2.5 Verzögerte Reparaturmaßnahmen und eigenschaftsabhängige Knotenauswahl

Ein- und Austritte von Knoten verursachen in einer DHT einen hohen Verwaltungsaufwand. Aufgrund von Knotenbewegung oder Umwelteinflüssen kommt es in MANETs zu häufigen Ein- und Austritten von Knoten, was zu einer Belastung der DHT führt. Um den dadurch entstehenden Aufwand zu verringern, muss die Eignung eines Knoten als aktiver Teilnehmer im DHT geprüft werden. Eigenschaften wie z.B. die stabile Verbindungsdauer zum MANET, die Verbindungsqualität und die Energiereserven sind Indikatoren für zuverlässige bzw. unzuverlässige Knoten. Durch eine Unterscheidung zwischen aktiv routenden und passiven DHT Teilnehmer können Instabilitäten in der DHT vermieden werden. Ein absehbarer Ausfall von schwachen Knoten führt dadurch zu keiner Belastung des Netzes.

Um den Aufwand durch unnötige Reparaturmaßnahmen nach unerwarteten Knotenausritten zu verringern, kann die Reparatur der Routing-Tabellen verzögert werden. Wird ein Knoten nach kurzer Zeit wieder verfügbar, so kann er, ohne erneuten Eintritt und damit ohne weiteren Aufwand, erneut an der DHT teilnehmen. Besonders in Situationen in denen ein Knoten über längere Zeit wechselnd erreichbar ist, kann dadurch eine dauerhafte Belastung der DHT vermieden werden.

3 Umfang der Arbeit

Bei der vorgestellten Arbeit handelt es sich um eine noch nicht abgeschlossene Arbeit. Die theoretischen Betrachtungen wurden zum großen Teil abgeschlossen. Zur praktischen Evaluierung wurde ein Modell in der Simulationsumgebung OMNeT++ [6] erstellt. Im Moment werden die beschriebenen Modifikationen an der DHT vorgenommen, um die Möglichkeit des effizienten Einsatzes in infrastrukturlosen Netzen anhand von Simulationen zu bewerten.

Literaturverzeichnis

- [1] R. Steinmetz und K. Wehrle. Peer-to-Peer-Networking & -Computing. Informatik-Spektrum, 27(1):51 – 54, Februar 2004. Springer, Heidelberg.
- [2] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, und I. Stoica. Looking Up Data in P2P Systems. Communications of the ACM, Vol. 46, No. 2, Feb. 2003.
- [3] C.E. Perkins. Ad-Hoc Networking. Addison-Wesley, 2000.
- [4] K. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, und I. Stoica. The Impact of DHT Routing Geometry on Resilience and Proximity. In Proc. of ACM Sigcomm, 2003.
- [5] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. F. Kaashoek, F. Dabek, und H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. IEEE/ACM Transactions on Networking, 11(1):17–32, Feb. 2003.
- [6] OMNeT++ Discrete Event Simulation System, <http://www.omnetpp.org/>

Harpy — Employing a Process Migration facility and a User Mode Linux for High Throughput Computing

Andreas Schäfer — February 4, 2005

”‘CLUSTER COMPUTING’”

Studienarbeit · Institut für Informatik · FSU Jena

Abstract

The available computing power is – according to Moore’s Law – doubling every 18 months. Nevertheless a typical engineer’s or physicist’s demand for computing power is doubling with the speed of thought. Obviously this causes a certain gap and most budgets have proven to be unable to fill it. Commonly researchers run their computations on a dedicated mainframe or cluster computer. Meanwhile – during lunchtime or at night – huge capabilities lie dormant, since in this time the staffs personal computers are not in use. HARPY aims at harnessing their free cycles to transparently and dynamically speed up the cluster computer.

1 Introduction to Harpy

When an engineer is designing a new workpiece, he usually simulates his design and tries to find a combination of parameters such that the outcome delivers the best solution to problem. When designing a new aircraft, he might test, how the aerodynamics change when the winglet-height is changed, the jet engines are tweaked or the tail unit is shifted. Since all possible combinations of these variations are interesting, he has to sweep the whole parameter space.

As you might guess, this is commonly referred to as a *parameter sweep* and does occur not only in the engineering environment but e.g. also when biochemists design new drugs and want to predict the active agent’s reactions in the body or when discrete mathematics are used to optimize the manufacturing schedule in companies. Generally spoken, parameter sweeps are a frequent requirement in research and development departments.

Cluster computers designed to process a large number of rather small jobs are referred to as High-Throughput clusters (HTP). There are already some other projects dealing with HTP Clusters. HARPY differs from them in the respect that it does not try to reinvent the wheel. Perhaps most important is the fact that it won’t create a completely new and virtual environment which the user is unfamiliar with. It rather tries to seamlessly integrate into the existing cluster, allowing the user to make transparently use of it’s capabilities. This is especially important, when the user relies on commercial, closed source solvers, as it is often the case. When he cannot modify the code or even recompile it, many other approaches like the one of the Condor project [1] are simply unfeasible for him. HARPY is the first project which caters for such a setup.

2 Basic Components and Work flow

This section will give you a view of HARPY from 10000 feet. We will have a glance at the most important components and discuss, how they are meant to work together.

In figure 1 you can see, that HARPY consists of three blocks. The first one is naturally formed by the master execution nodes, who are the dedicated cluster. As mentioned above, the key idea

is to avoid bothering the user with details of the system. He should be able to start his jobs on the master execution nodes just as he is used to.

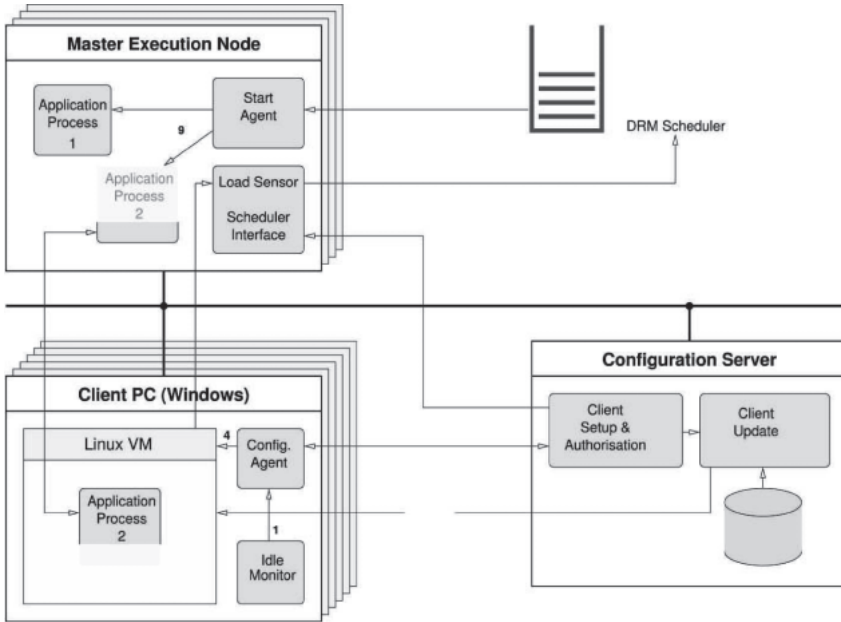


Figure 1: The basic structure of HARP Y .

HARP Y comes into play, when there are free computing resources. This is the case when, say, someone is at home or off for lunch. His idling computer could assist the cluster during the computations. We will refer to these spare PCs as the client PCs. HARP Y 's task is to detect them and migrate running jobs from the master nodes to the clients as long as those are not reclaimed by their owners. At the moment a user comes back to his PC, it should evict all of it's jobs, so that he can use his machine just as normal.

Obviously all this dynamic job migration would end up in total and utter chaos if it wasn't someone's job to keep track of what's going on. Therefore the need for the Configuration Server (CS) is obvious. His tasks are the following:

1. When a client wants to join the cluster, the server has to check the software installed on the client side. This is to ensure, that it is up to date and unmodified.
2. When the software is acceptable, the server provides the client with the information needed to configure itself. This includes e.g. the IP addresses of the other nodes and such.
3. The CS notifies all of the other nodes (masters as well as other clients), that a new client is available and how they can contact him.
4. The same is to be done, when a client leaves the cluster. Only, this time the other way round. The server tells the remaining nodes to delete the leaving node from their databases.
5. For the case that a client does not properly log off from the HARP Y pool, the CS has to check all of the nodes, if they are alive and kicking and remove those, who are not.

3 Job Migration

Until now we have left out one of the crucial mechanisms needed for the whole project: How can we migrate jobs from one machine to another? As already said, we don't want to reinvent the wheel when we can make use of well known, proven software.

There are some projects dealing with the job migration question. One of them is openMosix [2]. It's main competitors are openSSI [4] and Kerrighed [3].

It's advantage compared to openSSI is, that it does only light modifications to the operating system kernel. Kerrighed features a nicer theoretical approach but is not yet able to dynamically add and remove nodes from the cluster. Furthermore openMosix (oM) is a well tested middle-ware and stable enough for production use.

4 Client Side Architecture

So does openMosix alone solve our migration problem? The answer is obviously no. The problem is, that oM is designed for Linux. While this is absolutely fine for the master execution nodes – most decent clusters run some sort Linux – it isn't for the clients. The typical desktop PC runs some Version of Microsoft Windows. So how could we get our job migration facility to run on the client PCs? Basically there are three approaches, who do all have their own advantages and drawbacks:

- Porting openMosix to Windows
- Installing Linux on the Client PCs
- Use a Virtual Machine to Run openMosix on Windows

Porting openMosix to Windows This would be, from a theoretical point of view, the cleanest and most performant way. The pity is, that we would have to modify the client's operating system kernel. Since Windows is a pretty closed source world, this is no option.

Installing Linux on the Client PCs The favorite choice from an Linux users POV. One could build a dual boot solution, enabling the user to switch the running operating systems. However, this is no option for the common user, since he relies on his Windows and rebooting the machine whenever he leaves his PC unused is something he is not looking forward to. Surely, but of course purely accidentally, he would forget to do so very recently.

Use a Virtual Machine to Run openMosix on Windows One could insert a layer between openMosix and the really OS on the machine, such that oM could use it like any Linux machine while the user can keep his favorite OS from Redmond. The drawback is, that virtual machines tend to reduce performance. Still, this is the only feasible solution we're left with.

With the availability of Colinux [5] we are provided with a exciting new approach. Colinux is, unlike other Virtual Machines, able to run a Linux kernel within Windows without emulating virtual hardware but just by re-routing the hardware access of the Linux kernel towards the Windows side. This is achieved mainly by a Linux kernel patch and a Windows device driver.

5 Summary

Summing up, our approach employs two projects: openMosix is used as a job migration facility and Colinux allows openMosix to run on top of Windows. The challenge is to make the two softwares work together smoothly and to implement a configuration server which is able to cope with this highly dynamic setup.

References

- [1] Douglas Thain, Todd Tannenbaum, Miron Livny et al. *Condor High Throughput Computing*
<http://www.cs.wisc.edu/condor/>.
- [2] Moshe Bar, Bruce Knox et al. *openMosix*
<http://www.openmosix.org>.
- [3] Renaud Lottiaux et al. *Kerrighed*
<http://www.kerrighed.org>.
- [4] Moshe Bar, Bruce Knox, et al. *openMosix*
<http://www.openmosix.org>.
- [5] Dan Aloni and the Colinux community *Colinux*
<http://www.colinux.org>.

Implementing a framework for optimising parameter studies on cluster computing environments

Marcus Komann

Friedrich-Schiller-University Jena, Institute of Computer Science,
Ernst-Abbe-Platz 2, D-07443 Jena, Germany
marcus.komann@web.de

Type of project: Extension of a study thesis (2+ semesters)
Supervisor: Prof. Dr. Dietmar Fey

Abstract. A framework to carry out optimising parameter studies on a cluster computer has already been published. The intention of such computation-intensive studies is to find an optimal parameter set concerning a specific objective function. We applied this framework to an example of an optimised design of a sophisticated composed optoelectronic detector. We achieved a significant decrease in required computation time and achieved a detector design with improved characteristics after executing the test runs, proving the feasibility of the theoretical idea. The current project aims on carrying over the framework to a practical tool for the execution of arbitrary parameter studies, and enhancing the system in order to have the ability to also run parallel simulation programs. Furthermore, our aim is to create a Web portal, where users can use the tool in an easy and understandable way without knowledge of the underlying infrastructure or the cluster/grid management.

1 Introduction

The goal of our current work is to turn a generic framework for optimising parameter studies on a cluster computer into a customisable and applicable tool. Optimising parameter studies means that we look for a specific parameter set which optimises the behaviour of a simulated system. It is useful to carry out such studies on a cluster because one has to simulate a high number of different parameter sets what needs a lot of computation resources [1], [2]. Until now, there is low tool support for such classical high throughput computing applications. Only a few solutions exist, e.g. in the ILAB project [4], ZOO [5] or ZENTURIO [6].

One of the new approaches in our work is the emphasis on *optimising* scientific computing experiments by combining parameter studies with optimisation methods like for instance evolutionary algorithmic approaches on a cluster system

The frame work we presented before in [7] is based on four components. The first one is a master program which controls the parameter studies, i.e. it initiates single parameter runs, evaluates their results and creates new parameter sets with improved characteristics. The second component is a job management system which is used by

our master program to submit the simulation jobs to our cluster in order to achieve maximal processor utilisation. The third component is a system utility which starts the master program in periodic intervals. The fourth component is the simulation program to carry out a single parameter run.

In our current project, the intention is to offer a solution to users without making them learn a new language but by specifying the necessary requirements as far as possible through a Web portal. This portal should be set on top of the tool and provide a user-friendly GUI along with the possibility to submit the simulation program, a custom cost function, the parameter files and a termination constraint. The underlying infrastructure should take care of the computation and the usage of the available resources and hide it from the user.

The remainder of this paper is structured as follows. In section 2 we present our developed framework for the optimised parameter studies on cluster systems. Topic of section 3 is the current project development and a look on future goals. Finally, we will conclude with a summary.

2 Framework for optimised parameter studies

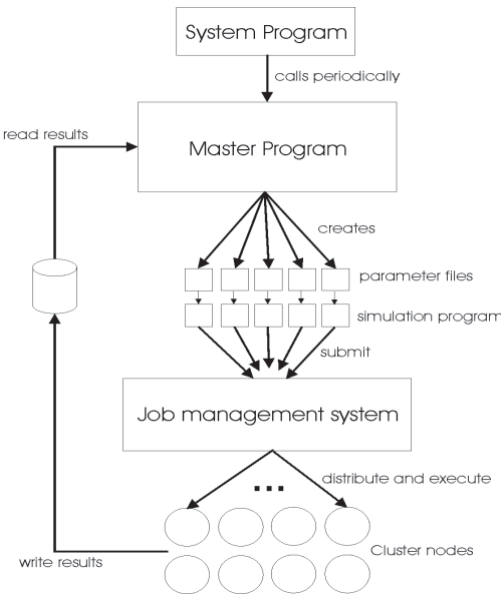


Figure 1: Overview of the framework

management system supplies for the optimal distribution of jobs on the available computation resources. For each parameter run parameter files and their corresponding job scripts are written into separate directories as well as result files and possible error files. After each following invocation the master program looks for results of terminated simulation runs. By means of a custom objective function and an evolu-

Figure 1 shows the interaction of the main components for the framework. The central part is the master program which was developed by us. A system utility invokes the master program periodically after a certain amount of time. In comparison to continuous running of the master program, this approach increases robustness in case of temporary system errors or crashes. Furthermore, it avoids unnecessary processor usage produced by a most of the time out-of-work master process running in the background.

The first time the master program is called, it creates the initial parameter files for the first simulation runs and generates the corresponding job scripts for the submission of the simulation programs to the job management system. The job

tionary optimisation algorithm the results are evaluated, new parameter sets are generated and the corresponding jobs are submitted.

3 Current project development and outlook

Currently, we have three different issues to work on. In the past, we only were able to apply serial numerical simulation programs to the framework. Therefore, the first goal is the extension of the framework for the application of parallel simulation tools. This is the topic where we have advanced most concerning the three issues mentioned here. The original framework was intended to run several single PC jobs concurrently on a cluster in order to achieve a higher job throughput and faster results by using the increased computation power. Instead of plugging a serial tool into the framework, it is now possible to use parallel numerical simulation programs which are divided automatically by the job management system. This feature extends the feasibility of the framework to the future generation of parallel numerical programs of which some have already been invented recently.

The second topic is the implementation of the generic tool. This is a sophisticated topic because the tool should hold for customisability and generality. It should be able to cope with any simulation program, objective function and parameter file. Furthermore, it should provide different optimisation methods like evolutionary algorithms, simulated annealing or maybe even user-defined techniques. Parameter studies have become more and more important in different areas, but especially in physics and engineering such a tool would provide a widely desired easy-to-use and fast opportunity for the computation of such problems.

And the third subject is the construction of the web portal. This portal should give a user from around the globe the possibility to easily access and use the tool described above. The portal should have a user-friendly GUI where all options could be entered. The computation of the entered processes should be fully hidden from the user. He should just get the results after some or a defined time and should not have to care for anything else. This is the last step in the development process and is therefore in state "to be done".

4 Summary

We have presented the current development of a framework for optimising parameter studies on cluster computers and shown the basic ideas of the system. The current work was described and the feasibility of the ongoing project was outlined. There is still a lot of work to do but the prospect of various and extensive applicability encourages the further development.

References

1. Abdalhaq, B., Cortés, A., Margalef, T., and Luque, E.: Evolutionary Optimization Techniques on Computational Grids. In: Sloot, P.M.A. et. al. (Eds.): ICCS 2002, LNCS 2329, Springer-Verlag Berlin Heidelberg (2002) 513-522.
2. Linderoth, J. and Wright S.J: Computational Grids for Stochastic Programming, Optimization. Technical Report 01-01, UW-Madison, Wisconsin-USA (2001).
3. Abramson, D., Susic, R., Giddy, R., and Hall, B.: Nimrod: A tool for performing parameterised simulations using distributed workstations high performance parametric modeling with nimrod/G: Killer application for the global grid? In *Proceedings of the 4th IEEE Symposium on High Performance Distributed Computing (HPDC-95)*, 520–528, Virginia, Aug. 1995. IEEE Computer Society Press.
4. Yarrow, M., McCann, Biswas, K.M., R., and der Wijngaart, R.F.V.: Ilab: An advanced user interface approach for complex parameter study process specification on the information power grid. In *Proceedings of Grid 2000: International Workshop on Grid Computing*, Bangalore, India, Dec. 2000. ACM Press and IEEE Computer Society Press.
5. Ioannidis, Y.E., Livny, M., Gupta, S., and Ponnkanti, N.: ZOO: A desktop experiment management environment. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, editors, *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases*, pages 274–285, Mumbai (Bombay), India, 3–6 Sept. 1996. Morgan Kaufmann.
6. Prodan, R., Fahringer, T.: ZENTURIO: An Experiment Management System for Cluster and Grid Computing. In *Proceedings of the 4th International Conference on Cluster Computing (CLUSTER 2002)*, Chicago, USA, September 2002. IEEE Computer Society Press.
7. Fey D., Komann M., Kauhaus C: A framework for optimising parameter studies on a cluster computer by the example of microsystem design, In D. Kranz Müller, P. Kaczuk and J. Dongarra (editors): *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, number 3241 in LNCS, pages 436 – 441, Springer, 2004.

P2P-Netzwerke für dynamische 3D-Szenen

BCompSc. Sebastian Gerdemann, BCompSc. Christian Fritzsche

mastersese@gmx.de, fritzsche@upb.de

Universität Paderborn

Projekttyp: Projektgruppe, 12 Teilnehmer, Dauer 2 Semester

Betreuer: Prof. Dr. Friedhelm Meyer auf der Heide, Dipl. Inform. Jan Klein, Dipl. Inform. Jens Krokowski, Dipl. Inform. Mirosław Korzeniowski & Asst. Prof. Dr. Christian Schindelhauer

Abstract: Entwicklung einer Massiv-Multiplayer P2P-3D-Engine für hochdynamische, virtuelle Szenen unter Verwendung effizienter Algorithmen für P2P-Architekturen, Computergrafik und verwaltende Datenstrukturen.

1 Einleitung

Die Visualisierung verteilter, hochdynamischer, virtueller Szenen stellt die Anforderung alle mobilen Objekte (Spieler) unabhängig voneinander in einer Szene bewegen zu können und ihre Bewegungen bei allen Mitspielern in Echtzeit darzustellen. Dazu müssen Bewegungsinformationen zwischen den Mitspielern ausgetauscht werden, welche stark reduziert werden müssen, um verteilte Spiele mit einer besonders großen Anzahl von Teilnehmern zu realisieren. Im Rahmen der Projektgruppe „Peer-to-Peer-Netzwerke für dynamische 3D-Szenen“ sollte hierzu eine neuartige 3D-Game-Engine entwickelt und evaluiert werden.

2 Architektur

Die entwickelte Game-Engine basiert auf einem Peer-to-Peer-Netzwerk, welches die Probleme der klassischen Client-Server-Architektur, wie beispielsweise den Flaschenhals durch erhöhte System- und Netzwerkbelastung eines Servers oder dem Single-Point-of-Failure Problem, umgehen soll. Um den Informationsaustausch zwischen den Rechnern zu minimieren, wurde eine spezielle, so genannte Mobile Datenstruktur entwickelt, die Vorhersagen bezüglich Bewegungen und Sichtbarkeit anderer Objekte ermöglicht. Für eine flüssige Darstellung einer 3D-Szene mit extrem vielen Objekten wurden verschiedene typische Grafikalgorithmen verwendet und erweitert. Ziel dieser Algorithmen ist es, die Anzahl der

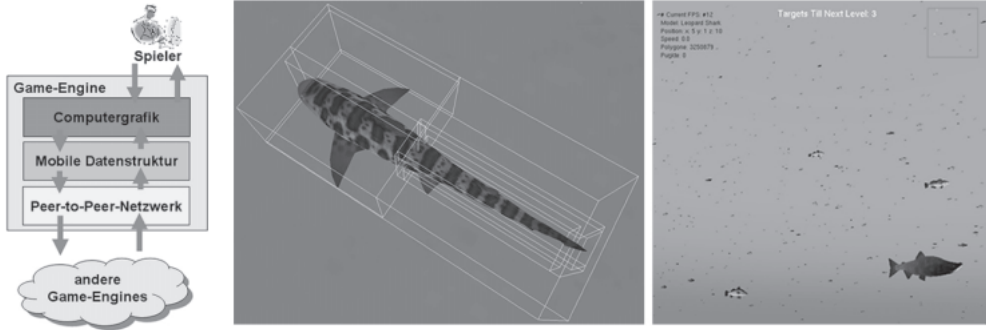


Abbildung 1: Links: 3-Schicht Architektur der Game-Engine. Mitte: Fischmodell - Polygone unterteilt in Knoten des Color-Cube-Tree. Rechts: Szene mit 1000 Fischen.

zu rendernden Polygone in der Szene effizient zu reduzieren, ohne dadurch einen Verlust der Bildqualität hinnehmen zu müssen.

2.1 Peer-2-Peer Schicht

Die P2P-Schicht verfügt über die Funktionalität des Routings von Nachrichten und der Verwaltung von Verbindungen zu anderen Peers.

Routing

Ein Peer routet eine Nachricht abhängig von ihrer Art weiter, wenn die eindeutige ID dieser Nachricht nicht in der im Peer gespeicherten Liste der letzten n Nachrichten-IDs enthalten ist. Somit werden nur noch nicht empfangene Nachrichten weitergeleitet, was das Broadcasten wichtiger Informationen, wie die Neuanmeldung eines Spielers im gesamten Netzwerk ermöglicht. Nachrichten geringerer Priorität werden an alle Nachbarn gesendet, wenn deren *Time-To-Live* (N_{TTL}) ($N_{TTL} \in [1, 3]$) > 1 ist. Andernfalls wird die Nachricht nur noch mit Wahrscheinlichkeit $p_k = p_{\min} + (1 - p_{\min}) \frac{1}{\sqrt{n_{\text{conn}} - 1}}$ weitergeleitet, wobei $0 < p_{\min} < 1, n_{\text{conn}} \in N$. Hierbei ist $n_{\text{conn}} > 1$, da bei nur einer Verbindung nichts weiter geroutet werden kann. p_k ist also bei kleinerer Verbindungsanzahl (n_{conn}) größer. Mit dieser Strategie wird der Traffic in Ballungsgebieten reduziert, denn in diesen Gebieten werden aufgrund der höheren Anzahl interessanter Peers mehr Verbindungen gehalten.

ConnectionManagement und Zusammenhang des Netzes

Das *ConnectionManagement* (CM) ist ein zentraler Teil der P2P-Schicht und ist verantwortlich möglichst zu jedem Zeitpunkt die optimalen Kommunikationspartner (Nachbarn) zu finden und Verbindungen zu diesen zu verwalten. In dem verwendeten Konzept wird ein Großteil der für diesen Zweck benötigten Informationen bei der MDS erfragt, was die P2P-Schicht und somit das Routing-Konzept stark anhängig von der MDS macht. Jede Verbindung wird durch ein *P2PConnection-Objekt* repräsentiert, welches eingehenden und ausgehenden Verbindungen differenziert. Die aktuellen Verbindungen werden in einer *ConnectionList* (CL) verwaltet, welche hauptsächlich zum Speichern von Zustands-

Informationen einer Verbindungen dient. Dies ist erforderlich um z.B. Informationen mit der MDS synchronisieren zu können. Um die Netzwerklast über das gesamte Netzwerk zu verteilen, existiert für jeden Peer eine minimale und eine maximale Anzahl von Verbindungen. Es wird stets angestrebt, die Anzahl der Verbindungen in der ConnectionList zwischen der minimal und maximal gewünschten Anzahl von Verbindungen zu halten. Ein angesprochener Peer muss jede Verbindung annehmen, auch wenn er bereits mehr Verbindungen als die maximal gewünschte Anzahl an Verbindungen hält. Tritt dies ein, versucht er Verbindungen abzubauen. Um hierbei keine Brückenkante abzubauen, wird über diese Kante eine Testnachricht geschickt welche nach n Hops wieder empfangen werden muss. [Cas03],[KLXH],[RD01]

2.2 Mobile Datenstruktur

MDS-Baum

Die *Mobile Datenstruktur* (MDS) basiert auf [LRSV03] und dient zur Speicherung und Verwaltung aller dynamischen und statischen Objekte der Szene am Peer. Hierzu spannt die MDS einen 6-dimensionalen, mobilen Raum über die Ortskoordinaten (x, y, z) und die Geschwindigkeitskoordinaten (v_x, v_y, v_z) auf. Anhand der Koordinaten in diesem Raum, der Größe und der Maximalbeschleunigung der Objekte sortiert die implementierte Datenstruktur, der MDS-Baum, alle Objekte in die Knoten des Baumes ein. Jeder Knoten besitzt hierbei in jeder Dimension drei Kindknoten, also $3^6 = 729$ Unterknoten. Diese Knoten, oder auch Würfel teilen die Szene in immer kleinere Bereiche auf, je tiefer der Würfel im Baum angesiedelt ist. Der Wurzelknoten beschreibt die gesamte Ausdehnung der Szene. Alle Objekte der Szene werden nun in diesen Baum anhand ihrer Position (x,y,z) so einsortiert, dass die Ausdehnung eines Würfels der nächst tieferen Ebene die Maße des Objekts (Geschwindigkeit, Größe und Maximalbeschleunigung) nicht aufnehmen kann.

Delta-Shuffle-Operation

Diese Sortierung wird durch Updates, die über das Peer-to-Peer-Netzwerk empfangen werden, aktualisiert. Treffen keine Informationen ein, kann der MDS-Baum durch eine periodisch ausgeführte Delta-Shuffle-Operation alle Positionen der Objekte rekonstruieren, und in Abhängigkeit der letzten Geschwindigkeits- und Beschleunigungsdaten in den MDS-Baum neu einordnen. Bei dieser Shuffle-Operation wird die Dreiteilung des Baums in jeder Dimension ausgenutzt. Alle Knoten die eine hohe positive Geschwindigkeit haben, werden in positive Richtung umgehungen, negative Geschwindigkeiten analog. Dabei führt jeder Knoten alle Objekte mit, die sich in ihm (bzw. im Baum unter ihm, d.h. alle Kindknoten und deren Kindknoten) befinden. Mittels dieser kontinuierlich aktuellen Informationen aller Objekte kann nun die Sichtbarkeit für den Spieler berechnet werden. Ein Objekt wird als potentiell sichtbar angenommen, wenn es größer ist als ein Pixel oder nicht innerhalb von zwei Bildern den Sichtbarkeitswinkel γ durchqueren kann. Für die 3D-Projektion ist es also nur notwendig ein Objekt in einem Rechteck mit Ortsausdehnung $\frac{2s}{\gamma}$ und einer Bewegungsausdehnung $\frac{4s}{b\gamma}$ zu betrachten. Da in jeder Ebene des MDS-Baumes Objekte mit ähnlicher Ausdehnung in gleichen Knoten liegen, können diese Objekte durch geeignete Traversierung des Baumes in einem zusammenhängenden Teil der Datenstruk-

tur gefunden werden. Mittels dieses Verfahrens ist es möglich die darzustellenden Objekte für die CG effizient zu reduzieren und eine flüssige Darstellung der Szene zu garantieren. [BKI03]

2.3 Computergrafik Schicht

Die Grafik Engine hat die Aufgabe die von der MDS vorsortierten Objekte weiter zu reduzieren und zu zeichnen. Hierzu werden Computergrafik Algorithmen zur Reduzierung der Polygonanzahl eingesetzt. Neben Standard Methoden wie Back-Face Culling und View-Frustum Culling, wurde ein Level of Detail(LOD) Verfahren namens Color-Cube-Tree(CCT) eingeführt.

Der implementierte CCT basiert auf einem erweiterten Binary-Space-Partitioning-Tree (BSP-Baum). Der BSP-Baum ist eine rekursive, hierarchische Aufteilung des 3-dimensionalen Raumes in konvexe Unterräume und wird von uns zur Verwaltung der Spielerobjekte eingesetzt. Zur Erzeugung des Baumes wird der Raum durch eine Hyperebene rekursiv in abwechselnd x,y und z-Ebene geschnitten und in Unterräume aufgeteilt. Die Schnittebene wird dabei immer so gewählt, dass die Polygonmenge mittig geteilt wird. So entsteht ein balancierter Binärbaum. Zusätzlich enthält jeder Knoten des Baumes ein Color-Cube. Er enthält pro Würfelseite (Blickrichtung) einen Farbwert, welcher einen Mischwert der Texturfarben der im Knoten enthaltenen Polygone für die entsprechende Blickrichtung auf den Knoten darstellt.

Das durch den CCT realisierte Level of Detail (LOD) ist angelehnt an das in [MS95] vorgestellte Konzept. In unserer Implementierung soll die Anzahl der darzustellenden Polygone reduziert werden, indem Polygone mit einer geringen auf der Darstellungsebene projizierten Fläche durch Punkte ersetzt werden. Hierzu wird allerdings nicht jedes Polygon einzeln betrachtet, sondern die in den Knoten des CC-Tree gespeicherten Polyongruppen. Bei der Traversierung des Baumes wird überprüft ob die Projektion der Bounding-Sphere des Knoten kleiner als ein Pixel ist. Ist dies der Fall, wird mit Hilfe der Informationen des Color-Cube des Knoten ein Punkt gezeichnet, und die Traversierung an dieser Stelle abgebrochen. Andernfalls wird die Traversierung fortgesetzt. Sie endet spätestens bei den Blättern des Baumes, mit dem Rendern der Polygone des Blattes.

3 Evaluierung und Fazit

Als Beispielanwendung wurde von uns ein Spiel implementiert, bei dem sich Spieler als Fischmodelle durch eine Unterwasserlandschaft bewegen und mit anderen Objekten interagieren können. Zu Evaluierungszwecken wurden mittels Bots auf ca. 50 Rechnern mehrere Hundert Mitspieler simuliert. Hierbei hat sich gezeigt, dass das P2P-Netzwerk erfolgreich bis zu einer Größe von etwa 500 Teilnehmern funktioniert, und das Netz dabei zusammenhängend bleibt. Allerdings übersteigt der Traffic in bestimmten Fällen die Bandbreite einer DSL-Verbindung, was durch Interpolation der Bewegung behoben wer-

den könnte. Weiterhin wäre ein intelligentes Routing denkbar, bei dem nur ein Teil der Nachbarn informiert werden muss.

Die Mobile Datenstruktur kann die Vorteile der Delta-Shuffle-Operation in hochdynamischen Szenen schwer umsetzen, da Bewegungen hier kaum voraussagbar sind.

Die entwickelte Grafikengine konnte in erweiterten Tests, mit bis zu 1000 Objekten in der näheren Umgebung (ca. 7Mio. Polygone), die Anzahl der zu rendernden Polygone bis zu 93% reduzieren. Die Approximation von Polyongruppen durch Punkte bietet ein effektives LOD ohne Verlust von Bildqualität.

Literatur

- [BKI03] Christoph Beierle und Gabriele Kern-Isberner. *Methoden wissensbasierter Systeme: Grundlagen, Algorithmen, Anwendungen*. Computational Intelligence. Vieweg Verlag, Braunschweig/Wiesbaden, 2. Auflage, 2003.
- [Cas03] M. Castro. An evaluation of scalable application-level multicast built using peer-to-peer overlay networks, 2003.
- [KLXH] Björn Knutsson, Honghui Lu, Wei Xu und Bryan Hopkins. Peer-to-Peer Support for Massively Multiplayer Games.
- [LRSV03] T. Lukovszki, S. Rührup, C. Schindelhauer und K. Volbert. Worst Case Mobility in Ad-Hoc Networks. In *ACM Symposium on Parallelism in Algorithms and Architectures*, Seiten 230–239, 2003.
- [MS95] Paulo W. C. Maciel und Peter Shirley. Visual navigation of large environments using textured clusters. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, Seiten Pages: 95 – ff., Monterey, California, United States, 1995.
- [RD01] Antony Rowstron und Peter Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, Seiten 329–350, 2001.

PIRANHA - Hunter of Idle Resources

Georg Birkenheuer, Wilke Hagelweide, Björn Hagemeier, Victor Japs, Matthias Keller,
Nikolas Mayr, Jan Meyer, Tobias Schumacher, Kerstin Voß, Markus Zajac

{birke, wilkeh, bjoernh, smoker, mkeller,
zark, janny, tobe, kerstin, zajac}@uni-paderborn.de

Universität Paderborn
Paderborn Center for Parallel Computing (PC²)
Fürstenallee 11, 33102 Paderborn

1 Einleitung

Am Paderborn Center for Parallel Computing (PC²) der Universität Paderborn hat sich im Zeitraum von Oktober 2003 bis September 2004 eine zehnköpfige Projektgruppe damit beschäftigt, die ungenutzten Rechenkapazitäten des Instituts für Informatik im Rahmen des High-Throughput-Computing (HTC) nutzbar zu machen [1]. Typische Anwendungen aus dem HTC sind zeitunkritische und lose gekoppelte Applikationen, bei denen Rechenressourcen über einen langen Zeitraum benötigt werden. Dazu zählen Raytracing, paralleles Kompilieren, Parameterstudien (z. B.: Suchen von Primzahlen) sowie Aufgaben aus der Kryptographie.

Das Ziel der Projektgruppe war, die ungenutzten Rechenressourcen der auf Linux basierten Rechnerpools für wissenschaftliche Berechnungen im HTC-Umfeld bereitzustellen und die Auslastung der PCs zu Zeiten geringer Nutzung (z. B. nachts, am Wochenende oder in Ferienzeiten) zu erhöhen. Dabei soll das zu entwickelnde System so gestaltet sein, dass die Anwender möglichst wenig Wissen über dessen Funktionsweise oder andere technische Aspekte benötigen.

2 Systemarchitektur

Das Ergebnis ist das Paderborn Idle Resource Allocation Harness (PIRANHA), ein Zaumzeug zum Suchen und Auslasten nicht verwendeter Ressourcen. Die Software ist ein Client-Server-System, welches als Middleware die Schnittstelle zwischen Benutzer und Ressourcen darstellt. Die Aufgaben dieser Middleware bestehen einerseits aus der Annahme neuer und andererseits aus der Verwaltung bereits vorhandener Rechenaufträge, indem diesen

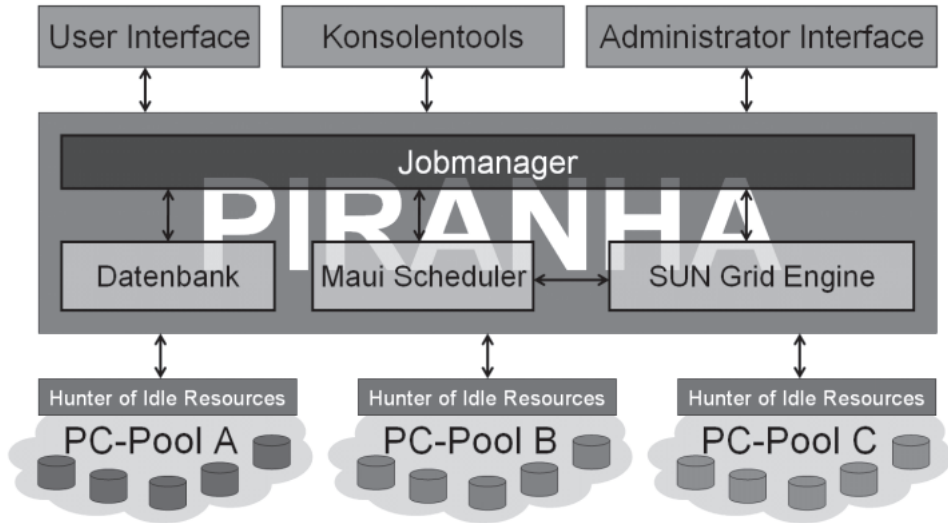


Abbildung 1: Der Aufbau von PIRANHA

geeignete Ressourcen zugewiesen werden. Die Benutzer müssen somit nur noch Rechenaufträge an das System vergeben und die Ergebnisse später abrufen. Alle anderen Zwischenschritte werden automatisch von PIRANHA erledigt. Der Aufbau des Systems wird in Abbildung 1 dargestellt, es gliedert sich in die Serverkomponente Jobmanager und die grafischen und kommandozeilenbasierten Benutzerschnittstellen. Die Serverkomponente besteht aus dem selbst entwickelten Jobmanager, der auf den Softwareprodukten Sun Grid Engine (SGE) [2] und dem Scheduler Maui [3] basiert. Die beiden Produkte wurden in der Planungsphase der Projektgruppe ausgewählt, da sich mit ihrer Funktionalität eine optimale Ausgangsposition für die Arbeit ergab. Sie wurden dabei den Anforderungen der Projektgruppe entsprechend angepasst und konfiguriert. Der Jobmanager übernimmt Verwaltungs- und Konfigurationsfunktionen. Er stellt für die Kommunikation mit den Client-Komponenten eine leicht erweiterbare CORBA-Schnittstelle zur Verfügung.

Das System bietet jeweils eine Schnittstelle für Benutzer und Administratoren. Die Schnittstelle für einen Benutzer stellt Funktionen zum Eingeben und Überwachen von Aufträgen zur Verfügung. Die Schnittstelle der Administratoren enthält zusätzliche Funktionalität zur Konfiguration und Administration des Systems. Dazu zählen sämtliche Monitoringinformationen, die Benutzerverwaltung und Konfiguration der Systemkomponenten. Die beiden Komponenten sind über die Benutzerverwaltung, die auf der Linux-Benutzerverwaltung basiert, zugriffsbeschränkt. Die Client-Komponenten nutzen die über den CORBA-Service bereitgestellte Schnittstelle des Jobmanagers, um den Zugriff auf die Serverkomponente zu erlangen. Der Benutzer kann die Kommandozeilenprogramme zum Einfügen, Löschen und zur Statusabfrage von Aufträgen in eigene Skripte einbinden. Umfangreiche Parameterstudien lassen sich so einfach und effizient durch PIRANHA berechnen.

Eine zusätzlich zu der Client- und Serverkomponente benötigte Software, bestehend aus einem mit Skripten angepassten Sun Grid Engine Execution Daemon, koordiniert und überwacht die Bearbeitung von Aufgaben auf jedem in das System integrierten Rechenknoten. Die Skripte überwachen dabei ständig den Zustand ihres PCs und melden eine Veränderung sofort an den Execution Daemon (z. B. ein Benutzer meldet sich am System an, kommt von der Mittagspause zurück, etc.). Dieser reagiert auf die Ressourcenanforderung und hält die laufenden Rechenaufträge an. Bei längerer Benutzung wird der von PIRANHA zugewiesene Auftrag abgebrochen und auf einem anderen freien PC erneut ausgeführt. Dadurch kommt es zu keiner Einschränkung des an dem Rechner angemeldeten Benutzers. Die Zusammenarbeit der Execution Daemons mit dem Jobmanager ermöglicht, die vorhandenen freien Rechenressourcen zu identifizieren und für Berechnungen bereitzustellen. Das System kann deshalb auch als Hunter of Idle Resources bezeichnet werden.

3 Ausblick

Das Konzept und die Implementierung von PIRANHA sind geprägt durch den Fokus auf komponentenbasierte Entwicklung und Modularität sowie stabile und robuste Lauffähigkeit, um die Wiederverwendbarkeit der Komponenten sicherzustellen und die Zuverlässigkeit der Software zu gewährleisten. Aufgrund des einfach gestalteten Aufbaus von PIRANHA und der Dokumentation können weitere Komponenten schnell und einfach in das System integriert werden.

Seit dem erfolgreichen Abschluss der Projektgruppe im Herbst 2004 befindet sich das PIRANHA-System im Betrieb auf einem Rechnerpool der Universität und hat dort bereits gute Dienste geleistet. Die Ausweitung auf weitere Rechner des Instituts ist geplant. Eine Anbindung des PIRANHA-Systems an bestehende Ressourcen Management Systeme wie das am PC² entwickelte CCS (Computing Center Software) [4] oder an Grid Middleware Infrastrukturen wie UNICORE [5] oder Globus [6] ist möglich. Hierdurch kann das PIRANHA-System in eine globale Umgebung integriert werden.

Literatur

- [1] Homepage der Projektgruppe: <http://wwwcs.upb.de/pc2/projects/piranha/main.htm>
- [2] Sun Grid Engine: <http://gridengine.sunsource.net/>
- [3] Maui Scheduler: Cluster Resources, Inc. Maui Scheduler Administrator Guide: <http://www.supercluster.org/mauidocs/mauiadmin.shtml>
- [4] CCS: <http://wwwcs.upb.de/pc2/projects/ccs/main.htm>
- [5] UNICORE: <http://unicore.sourceforge.net>
- [6] Globus: <http://www.globus.org>

Universität Paderborn

Projekttitel:

PIRANHA

Projekttyp:

Projektgruppe

Vertrauensdozent:

Prof. Dr. Uwe Kastens

Betreuung durch:

Prof. Dr. Odej Kao

Teilnehmer:

Name	Vorname	E-Mail	Semester
Birkenheuer	Georg	birke@uni-paderborn.de	9
Hagelweide	Wilke	wilkeh@uni-paderborn.de	9
Hagemeier	Björn	bjoernh@uni-paderborn.de	9
Japs	Victor	smoker@uni-paderborn.de	11
Keller	Matthias	mkeller@uni-paderborn.de	9
Mayr	Nikolas	zark@uni-paderborn.de	9
Meyer	Jan	janny@uni-paderborn.de	9
Schumacher	Tobias	tobe@uni-paderborn.de	9
Voß	Kerstin	kerstinv@uni-paderborn.de	9
Zajac	Markus	zajac@uni-paderborn.de	9

Regelbasierte Analyse und Komposition von Musik

Paul Grathwohl, Armin Fritsch, Jeremy Rögner, Mirco Kuhlmann

Universität Bremen
Angewandte Informatik
Postfach 33 04 40
D-28334 Bremen
pagra@informatik.uni-bremen.de
afritsch@informatik.uni-bremen.de
jeremy@informatik.uni-bremen.de
opti@informatik.uni-bremen.de

Art der Arbeit: Projekt über vier Semester im Hauptstudium Informatik (Diplom)

Betreuer der Arbeit: Prof. Dr. Karl-Heinz Rödiger

Abstract: In der Musikausbildung nimmt der Palestrina-Kontrapunkt eine wichtige Stellung ein. LerM ist ein Lehrsystem, das sowohl die Inhalte vermittelt, als auch interaktive Übungen ermöglicht. Schwerpunkt ist eine Kompositions-umgebung, mit einer an die Prädikatenlogik angelehnten Regelrepräsentation, in der Kompositionen auf Fehler geprüft werden können; mit ihr ist Musik auch generierbar.

1 Einleitung

Im Projekt LerM (Lehrsystem für ernstzunehmende Musik) wird ein Lehrsystem für den Kontrapunkt im Stil Palestrinas entwickelt. Der Komponist Giovanni Pierluigi da Palestrina (1525 - 1594) führte im 16. Jahrhundert die Vokalpolyphonie zu ihrem Höhepunkt. Kompositionen setzten sich aus einem von den gregorianischen Chorälen abgeleiteten Cantus Firmus¹, sowie mehreren weitgehend eigenständigen Stimmen, den Kontrapunkten² zusammen. Fux [Fu74] veröffentlichte 1725 das Lehrwerk „Gradus ad Parnassum“, in dem er die Kontrapunkttechnik Palestrinas anhand von mehreren Gattungen erklärt. Das Lehrsystem vermittelt die Theorie des Kontrapunktes nach Fux; der Schwerpunkt liegt dabei auf der praktischen Anwendung in Übungen. Daneben wird eine Umgebung für freie Komposition entwickelt, in der Benutzer anhand von selbst definierten Regeln eigene Stücke komponieren können.

¹ lat. feststehender Gesang

² lat. entgegengesetzt

2 Lehrsysteme für Musik

Es existieren bereits zahlreiche Lehrsysteme für Musik. Viele vermitteln Grundlagen der Musiktheorie (z.B. Notenkunde, Harmonielehre, Rhythmuslehre); andere befassen sich mit der Gehörbildung. Lehrsysteme für Komposition gibt es nur wenige. Die Firma Sibelius hat mit Sibelius Compass eine umfangreiche Lernsoftware für Komposition entwickelt, die jedoch das Thema Kontrapunkt nicht behandelt.

Einzig das von Huang [HC03] entwickelte Programm Palestrina Pal besitzt Teile, die den Zielen des Projektes LerM ähnlich sind. Mit diesem System können Komponisten und Musikstudierende ihre Kompositionen im Palestrina-Stil auf Stilfehler prüfen lassen. Jedoch wird weder die Theorie des Kontrapunktes vermittelt, noch werden Benutzer schrittweise an das Thema herangeführt.

3 Zielsetzung und Inhalte von LerM

Mit Hilfe des hier vorgestellten Lehrsystems können Benutzer ein solides Verständnis und einen sicheren Umgang mit der Palestrina-Kontrapunktlehre erwerben. Das generelle Lernziel ist, zu einem vorgegebenen Cantus Firmus einen bezogen auf den Palestrina-Stil fehlerfreien und möglichst auch musikalisch ansprechenden Kontrapunkt komponieren zu können. LerM kann unterrichtsbegleitend im Musik-Oberstufenunterricht und im Musikstudium eingesetzt werden. Kenntnis der Kontrapunkttechnik ist eine wichtige Grundlage für ein tiefes Musikverständnis und für eigene Schritte als Komponist. Im Musik-Grundstudium wird meist der Kontrapunkt im Palestrina-Stil behandelt. Dieser Stil beinhaltet viele klar definierte Regeln, welche musikalischen Strukturen erstrebenswert, welche unschön oder sogar verboten sind. Diese für die Musik sehr eindeutigen Regeln lassen sich im Unterricht gut schrittweise einführen; andererseits können sie relativ leicht in einer Wissensbasis erfasst werden.

Inhaltlich werden in einem ersten Block Intervalle, die Kirchentonarten sowie Kenntnisse über die damalige Musikpraxis und die Bedeutung des Kontrapunkts in der Musikgeschichte bis heute behandelt. Der zweite Block folgt der von Fux benutzten Gliederung in einzelne Kontrapunktgattungen (Note gegen Note, Note gegen zwei Noten, ...); mit ihm werden die allgemein anerkannten Regeln des Palestrina-Kontrapunktes vorgestellt.

3.1 Kompositionsübung

In LerM sind mehrere Übungsarten realisiert. Die wichtigste davon ist die Kompositionsübung, in der der Lernende zu einem vorgegebenen Cantus Firmus mit typischerweise ca. zehn Noten einen Kontrapunkt entwickeln soll. Er kann dabei seiner Kreativität - innerhalb der von den Regeln gesteckten Grenzen - freien Lauf lassen. Da es anfangs schwierig sein kann, alle Regeln gleichzeitig einzuhalten, bietet das Lehrsystem die Möglichkeit, die an der aktuell bearbeiteten Stelle erlaubten Töne anzuzeigen. Falls sich ein Teil der Komposition als besonders schwierig herausstellt, kann man sich Lösungsvorschläge hierfür generieren lassen.

Zur Selbstkontrolle bietet die Kompositionsübung dem Benutzer an zu prüfen, ob seine Komposition den Palestrina-Stil verletzt. Diese Prüfung kann nach dem Setzen jeder Note in Anspruch genommen werden oder nach Fertigstellung der Komposition. Wird eine Regel verletzt, wird dies dem Benutzer mit einer kurzen Erklärung angezeigt.

3.2 Freie Komposition

Die Umgebung für freie Komposition bietet dem Benutzer ein breites Experimentierfeld an. Er kann hier Regeln aus den vordefinierten auswählen und um von ihm entwickelte ergänzen. Während des Kompositionsprozesses, der von einem leeren Notenblatt ausgeht, werden die Regeln geprüft; der Benutzer erhält dabei eine Rückmeldung, ob seine Komposition dem von ihm über die Regeln definierten Stil genügt. Auf Wunsch können mit Hilfe der Regeln einzelne Noten bis hin zu ganzen Teilen generiert werden. Dies geschieht intern über einen rekursiven Suchalgorithmus, der Note für Note setzt und bei Sackgassen einzelne Schritte zurückgeht, um einen anderen Weg zu suchen. Dem Explorations- und Spieltrieb sind durch das Experimentieren mit der verwendeten Regelmenge und den daraus resultierenden automatisch generierten Kompositionen kaum Grenzen gesetzt.

4 Regelsystem

Die im Lehrsystem verwendeten Kontrapunktregeln sind intern in einer der Prädikatenlogik ähnlichen Form dargestellt:

$$P_1(t_1, t_2, \dots) \wedge \dots \wedge P_i(t_a, t_b, \dots) \Rightarrow P_j(t_{\bar{1}}, t_{\bar{2}}, \dots) \wedge \dots \wedge P_{\bar{z}}(t_{\bar{a}}, t_{\bar{b}}, \dots)$$

Jedes P ist dabei ein Prädikat, t sind Terme. So ergäbe z.B. $\text{Quinte}(c', g')$ wahr, da das Intervall zwischen den Noten c' und g' tatsächlich eine Quinte ist. Die Prädikate können jeweils noch negiert werden, ein komplizierter Formelaufbau ist aber nicht erlaubt. Mit Hilfe dieser Formeln können z.B. Kontrapunktregeln wie „Zwei Quinten dürfen nicht direkt aufeinander folgen“ oder auch „Der Umfang einer Stimme darf maximal eine Dezime sein“ ausgedrückt werden.

Neben der Formel besitzt jede Regel noch ein Maß zwischen 0 und 1, das beschreibt, wie strikt diese Regel beachtet werden muß. Manche musikalischen Situationen verwendet Palestrina nie (z.B. Quintparallelen), andere bevorzugt er. Ein Maß von 1 bedeutet, dass die Regel immer zutreffen muß. Für ein Maß von 0 gilt analog, dass diese Regel nie zutreffen darf. Im Bereich dazwischen gibt es kein richtig oder falsch, sondern nur eine Abstufung, wie gut die Komposition in den Palestrina-Stil passt. Diese weichen Regeln können somit neben der harten Fehleranalyse benutzt werden, um Lernenden Rückmeldungen zu geben, wie gut sie den Stil getroffen haben.

Diese Regeln sind die Grundlage der Generierung und der Fehlerprüfung. Beide basieren darauf, dass Noten auf Regelkonformität geprüft werden und ihnen dadurch eine Qualität zugeordnet wird. In der Generierung wird eine Tiefensuche über dem Suchraum der möglichen Kompositionen verwendet. In jedem Suchschritt wird aus der Menge der potentiell möglichen Noten eine gültige ausgewählt, wobei die Qualität einer Note die Wahrscheinlichkeit angibt, mit der sie ausgewählt wird. Ergebnis ist eine Komposition im vorgegebenen Stil.

Kompositionen werden auf Fehler geprüft, indem sequentiell alle Noten getestet und bewertet werden. Eine Komposition ohne Regelverletzungen wird als korrekt betrachtet; sie erhält auf Basis der einzelnen Notenqualitäten eine Bewertung. Bei einer Regelverletzung sind die genaue Stelle und die verletzte Regel bekannt; sie kann auf Verlangen ausgegeben werden.

5 Automatische Regelextraktion

In einem weiteren Ansatz des Projekts wird versucht, mit speziellen bayesschen Netzen, den Hidden Markov Modellen [Ra89], aus Stücken eines bestimmten musikalischen Stils automatisch Regeln zu extrahieren, die dann die Parameter des trainierten Netzes bestimmen. Die Extraktion findet unter Verwendung des Baum-Welch-Algorithmus statt. Ziel ist die automatische Generierung neuer Stücke, die den Stil möglichst gut repräsentieren. Notwendig hierfür ist eine große Trainingsmenge mit Stücken im MIDI-Format und die Nutzung des Forward-Algorithmus. Jedes Netz repräsentiert genau einen Stil. Wenn mehrere Hidden Markov Modelle für unterschiedliche Musikstile aufgebaut sind, können auch unbekannte Stücke klassifiziert werden.

6 Ausblick

Zwar spielt die Mehrzahl der teilnehmenden Informatik-Studierenden ein Instrument; dennoch wäre das LerM-System nicht ohne die begleitenden Lehrveranstaltungen und die Unterstützung von Dozenten des Musik-Fachbereichs in der jetzt vorliegenden Form zustande gekommen. In diesem Fachbereich soll es auch demnächst evaluiert werden. Ziele sind, die Regelbasis und die Benutzbarkeit des Lehrsystems zu überprüfen.

Weiterführende Arbeiten, z.B. in Form von Diplomarbeiten, bieten sich an, bez. einer Erweiterung des Lehrsystems um andere Musikstile, bez. einer Verbesserung der Regelbasis und bez. anderer Ansätze der automatischen Regelextraktion.

Literaturverzeichnis

- [Fu74] Fux, J.J.: Gradus ad Parnassum. Nachdr. d. Ausg. Leipzig 1742. Hildesheim, 1974
- [HC03] Huang, C.Z.A.; Chew, E.: Palestrina Pal - A Grammar Checker for Music Compositions in the Style of Palestrina. 2003, <http://www-scf.usc.edu/~chengzh/ahec.pdf>, Abruf am 05.01.2005
- [Ra89] A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. IEEE 77 (1989) No. 2, pp. 257-286

Mitglieder des Projekts LerM

Oliver Ahlbrecht, Marcus Barann, Armin Fritsch, Paul Grathwohl, Aysun Gündogan, Sonja Hildebrandt, Dennis Kaspuhl, Marcel Klaas, Daniel Kramer, Mirco Kuhlmann, Lukas Kumai, Gang Pan, Jeremy Rögner, Florian Scholz, Andreas Seewaldt, Fabian Zweidar

Implementierung von Default-Regeln durch optimale konditionale Rangfunktionen

Bachelor Abschlussarbeit im Fach Informatik (FernUniversität Hagen)

Christian Müller (Christian.Mueller@cm99.de) - Betreuerin: Prof. Dr. Gabriele Kern-Isberner

1 Motivation und Ziele

Eine der bedeutendsten Fähigkeiten des Menschen ist seine Fähigkeit intelligent denken zu können. Der Mensch ist in der Lage, in „unsicheren“ Situationen die richtigen Entscheidungen zu treffen und neues Wissen zu lernen. Im Gegensatz hierzu steht ein Computer, der grob gesprochen eine elementare Maschine ist. Es stellt sich nun die Frage, ob wir unsere menschliche Intelligenz auf einen Computer übertragen können, mit dem Ziel, eine intelligente Maschine zu erhalten?

Ich glaube, dass heutzutage niemand diese Frage wirklich beantworten kann, ob dies einmal möglich sein wird; jedenfalls wenn wir von der Intelligenz des Menschen als Ganzes ausgehen. Anders kann es aber aussehen, wenn wir nur ein bestimmtes Problem betrachten und „portieren“ dieses Problem auf einen Computer. Als Problemstellung wählen wir die Aufgabe: Wenn eine beschränkte Menge an Wissen gegeben ist, dann soll nach Möglichkeit neues Wissen entwickelt werden und auch auf Fragen über dieses Wissen eine Antwort gegeben werden können.

Eine passende Methode für diese Aufgabe sind die qualitativen Wahrscheinlichkeiten (Rangfunktionen [Spo87]). Hier wird Wissen in der Form von Default-Regeln (Konditionalen) dargestellt, d.h. die Regeln werden als richtig bewertet solange wir über kein gegenteiliges Wissen verfügen. Die Idee einer Rangfunktion ist, dass alle Regeln einen Rang zugewiesen bekommen. Dieser drückt die Plausibilität der entsprechenden Regel aus. Ein Rang 0 bedeutet, dass wir diese Regel für plausibel halten; je höher der Rang wird, desto unplausibler wird die Regel. Nachdem die Ränge für die Regeln berechnet sind, kann jeder möglichen Welt ein Rang zugewiesen werden, der äquivalent die Plausibilität dieser Welt repräsentiert.

Eines der ersten Systeme welches Rangfunktionen umsetzt, ist System-P von Adams [Ada75]. Als weiteres System wurde von Pearl [GP96] das System-Z eingeführt, welches mit einer eindeutigen und minimalen Rangfunktion κ^Z arbeitet. Als weiterer Ansatz wurde nun von [GMP93] das System-Z* vorgestellt, welches die Methode der maximalen Entropie ([Par95], [SW49]) und das System-Z kombiniert. Dieses System wurde anschließend von [Bou99] und [KI01] erweitert bzw. verallgemeinert.

Diese Erweiterungen führen uns nun zu einem neuen Problem. Hier kann es vorkommen, dass eine Berechnung der Ränge kein eindeutiges Ergebnis mehr liefert. Die existierende Implementierung dieses Ansatz von Bourne [Bou99] berechnet für diesen Ansatz zwar ein minimales Ergebnis, aber auch immer nur eins. Falls nun aber mehrere minimale Ergebnisse existieren, werden diese von der Bourne Implementierung nicht gefunden. In dieser Bachelor-Arbeit wird ein Algorithmus entwickelt und implementiert, welcher die Einschränkungen der bisherigen Implementierung umgeht und in der Lage ist, alle existierenden minimalen Lösungen zu berechnen.

2 Grundlagen

Als Basissprache betrachten wir eine endliche Menge \mathcal{L} bestehend aus Atomen und den beiden aussagenlogischen Konstanten \top (= wahr) und \perp (= falsch). \mathcal{L}_p ist eine Menge von wohlgeformten Formeln (WFF), die mit den üblichen Junktoren der Aussagenlogik gebildet wird. Eine Welt ω ist eine Interpretation für die Atome in \mathcal{L} . Die Menge der möglichen Welten wird durch Ω dargestellt. Die Erfüllungrelation für eine Formel $\varphi \in \mathcal{L}_p$ ist wie üblich definiert und wird dargestellt durch $\omega \models \varphi$. Eine Welt ω ist ein

Modell für φ gdw. die Welt ω die Formel φ erfüllt. Regeln (Konditionale) der Form $(\psi|\varphi)$ bzw. $\varphi \rightarrow \psi$ stehen für die Aussage „Wenn φ , dann ist ψ (plausibel)“ mit $\psi, \varphi \in \mathcal{L}_p$. Eine Welt ω *verifiziert* eine Regel $(\psi|\varphi)$ gdw. $\omega \models \varphi \wedge \psi$, *falsifiziert* eine Regel $(\psi|\varphi)$ gdw. $\omega \models \varphi \wedge \neg\psi$ und *erfüllt* eine Regel $(\psi|\varphi)$ gdw. die Regel durch ω nicht falsifiziert wird. Nach diesem begrifflichen Vorbemerkungen wollen wir nun formal eine Rangfunktion κ wie folgt definieren:

Definition 2.1 (Rangfunktion) Eine Rangfunktion κ weist jeder Welt in Ω eine ganze nicht-negative Zahl zu, so dass für mindestens eine Welt $\omega \in \Omega$ gilt: $\kappa(\omega) = 0$.

Um Ränge auf Formeln $\varphi \in \mathcal{L}_p$ zu erhalten, erweitern wir die Definition 2.1 wie folgt:

$$\kappa(\varphi) = \begin{cases} \min_{\omega \models \varphi} \kappa(\omega) & : \varphi \text{ ist erfüllbar} \\ \infty & : \text{sonst} \end{cases} \quad (1)$$

Zusätzlich fordern wir von einer Rangfunktion κ , dass die Ränge von „normalen“ Welten kleiner sind, als die von „Ausnahmewelten“. Dies wird durch die Zulässigkeit einer Rangfunktion festgehalten:

Definition 2.2 (Zulässigkeit) Gegeben: Wissensbasis $\Delta = \{\varphi_i \rightarrow \psi_i\}, 1 \leq i \leq n$. Eine Rangfunktion κ ist zulässig für Δ gdw. $\kappa(\varphi_i \wedge \psi_i) < \kappa(\varphi_i \wedge \neg\psi_i)$ für jede Regel $\varphi_i \rightarrow \psi_i \in \Delta$ gilt.

Definition 2.3 Eine Wissensbasis Δ ist konsistent gdw. eine zulässige Rangfunktion κ für Δ existiert.

3 c-Repräsentation

Als Basis für den Algorithmus benutzen wir die Erweiterung von Kern-Isberner (vgl. [KI01]). Den Ausgangspunkt der Erweiterung bildet die indifferente Rangfunktion:

Theorem 3.1 Gegeben: Wissensbasis $\mathcal{R} = \{(B_1|A_1), \dots, (B_n|A_n)\}$. Eine endliche Rangfunktion κ ist indifferent zur Wissensbasis \mathcal{R} gdw. es rationalen Zahlen $\kappa_0, \kappa_i^+, \kappa_i^- \in \mathbb{Q}, 1 \leq i \leq n$ gibt, so dass für alle $\omega \in \Omega$ gilt:

$$\kappa(\omega) = \kappa_0 + \sum_{\substack{1 \leq i \leq n \\ \omega \models A_i B_i}} \kappa_i^+ + \sum_{\substack{1 \leq i \leq n \\ \omega \models A_i \bar{B}_i}} \kappa_i^- \quad (2)$$

Mit Hilfe dieses Theorems können wir die c-Repräsentation wie folgt definieren:

Definition 3.2 Rangfunktionen κ , die eine Wissensbasis \mathcal{R} repräsentieren und indifferent in Bezug zu dieser sind, werden **c-Repräsentationen allgemeinen Typs** genannt.

Theorem 3.1 liefert uns ein klares Schema um c-Repräsentationen zu konstruieren. Beispielweise legen wir κ gemäß Gleichung 2 fest und wählen κ_i^+, κ_i^- entsprechend um sicherzustellen, dass $\kappa \models \mathcal{R}$ gilt, d.h. so dass $\kappa(A_i \bar{B}_i) - \kappa(A_i B_i) > 0$ für alle $i, 1 \leq i \leq n$ (vgl. Def. 2.2 und Formel 1). Außerdem muss κ_0 zur Normierung geeignet gewählt werden. Wir erhalten somit: $\min_{\omega \models A_i \bar{B}_i} \kappa(\omega) - \min_{\omega \models A_i B_i} \kappa(\omega) > 0 + (m_i)$ Durch Umformen sowie durch wählen der Einschränkungen $\kappa_i^+ := 0$ und $\kappa_i^- \geq 0$ für jede Regel $r_i \in \mathcal{R}$ erhalten wir ein verallgemeinertes System- Z^* :

$$\kappa_i^- > \min_{\omega \models A_i B_i} \sum_{\substack{j \neq i \\ \omega \models A_j \bar{B}_j}} \kappa_j^- - \min_{\omega \models A_i \bar{B}_i} \sum_{\substack{j \neq i \\ \omega \models A_j \bar{B}_j}} \kappa_j^- \quad (3)$$

Definition 3.3 (c-Repräsentation) Rangfunktionen gemäß Formel 3 nennen wir c-Repräsentationen speziellen Typs, bzw. im weiteren Verlauf einfach c-Repräsentation.

Nachdem die κ_i^- berechnet worden sind, läßt sich die zugehörige Rangfunktion κ_c wie folgt berechnen:

$$\kappa_c(\omega) = \sum_{\substack{1 \leq i \leq n \\ \omega \models A_i \bar{B}_i}} \kappa_i^- \quad (4)$$

4 MinV und MinF

Für den Algorithmus benötigten wir einige zusätzliche Funktionen. Hierzu definieren wir die Funktionen $MinV$ und $MinF$ für Regeln $r_i = A_i \rightarrow B_i$ wie folgt:

$$MinV(r_i) := \min_{\omega \models A_i B_i} \sum_{\omega \models A_j \bar{B}_j} \sum_{j \neq i} \kappa_j^- \quad MinF(r_i) := \min_{\omega \models A_i \bar{B}_i} \sum_{\omega \models A_j \bar{B}_j} \sum_{j \neq i} \kappa_j^-$$

bei gegebenen $\{\kappa_1^-, \dots, \kappa_n^-\}$. Durch Einsetzen der Formeln $MinV$ und $MinF$ in die Formel 3 erhalten wir: $\kappa_i^- > MinV(r_i) - MinF(r_i)$. Da wir nur minimale Lösungen berechnen wollen, können wir die Formel wie folgt abändern: $\kappa_i^- = 1 + MinV(r_i) - MinF(r_i)$.

5 Die eindeutige MinF-Funktion und die Alternativen Werte

Bei der Berechnung der MinF-Funktion ist es innerhalb des Algorithmus möglich, dass bei der Berechnung der min-Funktion ein unendlicher Wert enthalten ist und zwar gdw. mehrere Lösungen existieren können. Die Berechnung könnte z.B. wie folgt aussehen: $MinF = \min\{4, 3, 1 + \infty, \infty\}$ Das Vorhandensein von unendlichen Werten bedeutet, dass in die Berechnung noch nicht berechnete κ_i^- -Werte eingeflossen sind. Da wir zu diesem Zeitpunkt noch kein Wissen über diese κ_i^- -Werte haben, wäre es natürlich möglich, dass diese Werte den MinF-Wert verändern könnten. Wenn z.B. der „erste“ ∞ -Wert gleich 1 wäre, dann wäre der MinF-Wert 2 und nicht 3.

Dies zeigt, dass wir diese „möglichen“ Alternativen in Betracht ziehen sollten. Hierzu wird eine Menge \mathcal{A} von *Alternativen Werten* berechnet. Bei der Berechnung dieser Alternativen Werte müssen nun einige Punkte berücksichtigt werden, damit wir nicht unnötig doppelte oder auch falsche bzw. nicht minimale Lösungen berechnen.

6 Der Algorithmus c-Ordnung

Die Abbildung 1 zeigt den Algorithmus c-Ordnung, welcher bei Eingabe einer Wissensbasis alle minimalen Lösungen findet und ausgibt. Dieser Algorithmus terminiert immer und findet bei einer gegebenen konsistenten Wissensbasis eine oder mehrere endliche nicht-negative Lösungen.

7 Ideen und Ausblick

In dieser Arbeit wurde ein Algorithmus und eine Implementierung vorgestellt der Lage ist, alle möglichen minimalen Lösungen einer c-Repräsentation zu erkennen und zu berechnen. Die Basis dieses Algorithmus bilden die neu eingeführten Konzepte der *eindeutigen MinF-Funktion* und der *Menge der Alternativen Werte*. Aufgrund der eindeutigen MinF-Funktion ist der Algorithmus in der Lage, in jedem Durchlauf zu erkennen, ob für eine Regel mehrere oder nur ein κ^- -Wert existiert. Wenn der Algorithmus nun erkennt, dass mehrere κ^- -Werte existieren können, übernimmt die Menge der *Alternativen Werte* die Berechnung dieser Werte. Durch mehrere rekursive Aufrufe des Algorithmus für jeden bestimmten Wert berechnet der Algorithmus mehrere Lösungen.

Durch diese Implementierung waren wir in der Lage, die c-Repräsentation und das Verhaltens dieser unter mehreren minimalen Lösungen mit komplexen Wissensbasen genauer zu betrachten. An den durchgeführten Berechnungen konnte man erkennen, dass es, natürlich abhängig von der gegebenen Wissensbasis, eine Vielzahl von minimalen Lösungen geben kann. Wir konnten Tendenzen erkennen, dass wenn die Anzahl der „abhängigen“¹ Regeln in einer Wissensbasis steigt, auch die Anzahl der minimalen Lösungen steigt. Eine ähnliche Tendenz konnte man auch an der Anzahl der Welten, die unterschiedliche Ränge erhalten, erkennen.

Die Möglichkeit, komplexere c-Repräsentation zu berechnen, bringt wieder neue Fragen ans Licht: Kann man im Vorfeld berechnen, ob und wie viele minimalen Lösungen es gibt? Gibt es eine „ideale“ minimale Lösung bzw. welche minimale Lösung wählt man? Kann man mehrere minimale Lösungen unterbinden

¹Regeln, die bei mehreren minimalen Lösungen verschiedene κ -Werte erhalten

algorithmus c-Ordnung

Eingabe: \mathcal{R} Menge von Regeln $\{r_i : a_i \Rightarrow b_i\}$ mit $1 \leq i \leq n$

Ausgabe: Lösungsmenge \mathcal{L} von κ -Tupeln $(\kappa_1^-, \dots, \kappa_n^-)$

Initialer Aufruf: c-Ordnung-Rekursiv((∞, \dots, ∞))

begin

algorithmus c-Ordnung-Rekursiv(Eingabe: $(\kappa_1^-, \dots, \kappa_n^-)$)

Wenn Regel mit $\kappa = \infty$ in \mathcal{R} vorhanden **dann**

1. Für jede Regel $r_i \in \mathcal{R}$ mit $\kappa_i^- = \infty$, berechne $MinV(r_i)$
2. **Wenn** minimaler $MinV(r_i)$ -Wert = ∞ **dann** Abbruch, Regelmenge ist inkonsistent
3. Für jede Regel r_i mit minimalem $MinV(r_i)$, berechne $MinF(r_i)$
4. Wähle erste Regel r_j mit minimalem $MinF(r_i)$ -Wert

(a) **Wenn** $MinF(r_j)$ eindeutig **dann**

- i. $\kappa_j^- = 1 + MinV(r_j) - MinF(r_j)$
- ii. Wenn $\kappa_j^- < 0$ dann $\kappa_j^- = 0$
- iii. C-Ordnung-Rekursiv($(\kappa_1^-, \dots, \kappa_n^-)$)

(b) **Sonst**

- i. Berechne Menge \mathcal{A} von Alternativen Werten für $MinF(r_j)$
- ii. Für jeden Wert $x \in \mathcal{A}$:
 - A. $\kappa_j^- = 1 + MinV(r_j) - x$
 - B. c-Ordnung-Rekursiv($(\kappa_1^-, \dots, \kappa_n^-)$)

else Wenn $(\kappa_1^-, \dots, \kappa_n^-) =$ gültige Lösung **dann** Füge $(\kappa_1^-, \dots, \kappa_n^-)$ der Lösungsmenge \mathcal{L} hinzu

end algorithmus c-Ordnung-Rekursiv

Berechne für jede Welt die Ränge gemäß Formel 4 für jede Lösung in \mathcal{L} **end**

Abbildung 1: Algorithmus c-Ordnung

ohne die Ränge der Welten zu verändern (ggf. durch „Bereinigung der Wissensbasis“)? Diese Frage bleiben aber, jedenfalls in dieser Arbeit, offen. Diese Arbeit hat allerdings den Grundstein dafür gelegt, sich diesen Fragen effizienter zu stellen.

Literatur

- [Ada75] Ernest Wilcox Adams. *The Logic of Conditionals*. D. Reidel, Dordrecht, 1. Auflage, 1975.
- [Bou99] R. A. Bourne. *Default reasoning using maximum entropy and variable strength defaults*. Dissertation, University of London, 1999.
- [GMP93] M. Goldszmidt, P. Morris und J. Pearl. A Maximum Entropy Approach to Nonmonotonic Reasoning. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 15(3):220–232, mar 1993.
- [GP96] Moisés Goldszmidt und Judea Pearl. Qualitative Probabilities for Default Reasoning, Belief Revision, and Causal Modeling. *Artificial Intelligence*, 84(1–2):57–112, 1996.
- [KI01] Gabriele Kern-Isberner. Handling Conditionals Adequately in Uncertain Reasoning and belief revision. *Journal of Applied Non-Classical Logics*, 11:1–90, 2001.
- [Par95] J. B. Paris. *The Uncertain Reasoner's Companion: A Mathematical Perspective*. Cambridge University Press, Cambridge, England, 1995.
- [Spo87] W. Spohn. Ordinal Conditional Functions: A Dynamic Theory of Epistemic States. In W. Harper und B. Skyrms, Hrsg., *Causation in Decision, Belief Change and Statistics*, Jgg. volume 2, Seiten 105–134. D. Reidel, Dordrecht, 1987.
- [SW49] Claude E. Shannon und Warren Weaver. *The mathematical theory of communication*. University of Illinois Press, 1949.

Zeichnen von Begrenzungen ungenau definierter Gebiete

Markus Völker

Universität Karlsruhe
Fakultät für Informatik
mvoelker@ira.uka.de

Art der Arbeit: Arbeit als wissenschaftliche Hilfskraft
Mitarbeiter: Iris Reinbacher, Marc Benkert, Marc van Kreveld, Alexander Wolff
Betreuer: Alexander Wolff

Abstract: Bei geographischen Informationsanfragen wird oftmals nach Regionen ohne klar definierte Begrenzungen gesucht, wie z.B. nach *Süddeutschland*. Unsere Arbeit beschäftigt sich mit der Bestimmung angemessener Begrenzungspolygone derartiger Regionen. Dies führt zu Problemen im Zusammenhang mit rot-blau-Separation und Polygonen mit minimalem Umfang. Wir haben hierzu mehrere Algorithmen entworfen, implementiert und untersucht, von denen einige im Folgenden skizziert werden.

1 Einleitung

Manche Suchmaschinen ermöglichen die Suche nach Objekten in der geographischen Nachbarschaft z.B. eines Ortes. Wird nach *Museum* und *Karlsruhe* gesucht, so sollen auch Seiten von Museen im Umkreis von Karlsruhe, welche die beiden Worte nicht enthalten, angezeigt werden. Um derartige Anfragen bearbeiten zu können, muss die Datenbank eine Verbindung zwischen dem Suchbegriff und einer geographischen Region herstellen können. Für nicht exakt definierte Regionen wie *Süddeutschland* ist es daher von Interesse, *sinnvolle* Begrenzungen zu finden. In [2] wird beschrieben, wie man das World Wide Web nutzen kann, um für größere Städte die Zugehörigkeit zu einer bestimmten Region zu testen. Man erhält auf diese Weise eine Karte mit roten Punkten (Städte, die vermutlich zur Region gehören) und blauen Punkten (Städte, die vermutlich nicht in die Region gehören). Sind die Zuordnungen sicher korrekt, so existiert ein Algorithmus zum Zeichnen einer Begrenzung der Region von Alani et al. [1], der auf Voronoi Diagrammen basiert. Wir haben uns mit dem Fall beschäftigt, dass ein Teil der Zuordnungen fehlerbehaftet ist. Dies führt zum folgenden Problem: Gegeben eine Menge *innerer* Punkte (rot) und eine Menge *äußerer* Punkte (blau), soll ein *sinnvolles* Polygon bestimmt werden, das die beiden Mengen trennt. Hierbei streben wir Polygone an, die einen geringen Umfang, ein hohes Kompaktheitsverhältnis [10, 11] und möglichst wenige falsch zugeordnete Punkte besitzen. In der algorithmischen Geometrie existieren bereits Algorithmen, die eine rot-blau-Trennung mittels einer Linie, Streifen, Sektoren und ähnlichen festen Formen vornehmen [5, 3]. Für die Bestimmung eines Begrenzungspolygons mit minimalem Umfang konnte NP-Härte gezeigt werden [4, 7].

2 Anpassungsmethode

Bei der ersten Methode, die wir untersucht haben, beginnen wir mit einem Polygon, das möglichst viele rote Punkte einschließt und noch blaue Punkte enthalten darf, und versuchen die Gestalt des Polygons so anzupassen, dass die blauen Punkte nicht mehr im Inneren liegen. Einen guten Ausgangspunkt für ein Begrenzungspolygon liefert die sogenannte α -Shape [8, 9], eine Verallgemeinerung der konvexen Hülle. Ist die Menge der roten Punkte hinreichend groß, erhält man mit der α -Shape ein einfaches Polygon, dessen Gestalt durch den Wert von α bestimmt wird und bei dem starke Ausreißer unter den roten Punkten von vornherein ausgegrenzt werden.

Jetzt werden blaue Punkte, die noch im Polygoninneren liegen, durch Anpassung des Polygons nach außen gebracht und zwar dermaßen, dass alle zuvor im Polygon liegenden roten Punkte auch weiterhin im Polygon liegen und der Umfang des erzeugten Polygons minimiert wird. Es bietet sich an, nur solche blauen Punkte nach außen zu bringen, bei denen die Form des Polygons, gemessen am Umfang, nicht allzusehr verändert wird. Hieraus ergeben sich u.a. folgende Probleme, zu denen wir Algorithmen konstruiert haben:

Gegeben ein einfaches Polygon P (rot), eine Menge R roter Punkte im Inneren von P und ein blauer Punkt b in P . Bestimme das Polygon P' , das alle Kanten von P bis auf eine besitzt und in der Weise geschlossen ist, dass b nicht innerhalb von P' liegt, alle Punkte aus R in P' liegen und der Umfang von P' minimal ist. Ist n die Anzahl der roten Punkte auf dem Polygon und im Inneren, so lässt sich das Problem in $O(n \log n)$ lösen.

Für das Problem, bei dem im Inneren statt des einen blauen Punktes m blaue Punkte liegen, haben wir einen Algorithmus mit Zeitkomplexität $O(C^{m \log m} \cdot n)$ gefunden, wobei C eine Konstante ist. Für konstantes m ergibt sich damit eine Laufzeit von $O(n)$ und das Problem ist damit *fixed parameter tractable*. Dies ist von Interesse, da nur eine geringe Anzahl falsch klassifizierter Punkte zu erwarten ist.

3 Umfärbungsmethoden

Unser zweiter Ansatz besteht darin, nicht das Begrenzungspolygon, sondern die Klassifizierung der Punkte zu ändern. Hierbei sollen Punkte, die von Punkten der anderen Farbe umgeben sind und damit mit hoher Wahrscheinlichkeit falsch klassifiziert wurden, umgefärbt werden. Um zu testen, ob ein Punkt von Punkten der anderen Farbe umringt ist, bestimmen wir die Delaunay-Triangulation der Punktmenge. Jetzt kann man für jeden Punkt prüfen, ob in der Triangulation mehrere aufeinanderfolgende Nachbarn des Punktes die andere Farbe besitzen. Bilden die andersfarbigen Nachbarn einen Winkel ϕ , der einen bestimmten Schwellwert $\Phi \geq 180^\circ$ überschreitet, so färben wir den Punkt um. Da die Reihenfolge der Umfärbungen eine Rolle für das Endresultat spielt, haben wir mehrere Regeln, nach denen umgefärbt werden kann, untersucht. Eine der Regeln besteht darin, nur dann umzufärben, wenn dadurch der Umfang des Begrenzungspolygons abnimmt und $\phi \geq \Phi$ gilt. Dies ist naheliegend, da wir einen möglichst kleinen Umfang anstreben. Für den daraus resultierenden Algorithmus konnten wir eine untere Schranke von $\Omega(n^2)$ für

die im worst-case benötigte Anzahl von Umfärbungen und eine obere Grenze von $2^n - 1$ beweisen, wobei n die Anzahl der Punkte ist. Färbt man nur um, wenn $\phi \geq \Phi$ gilt und die Differenz der Anzahl der andersfarbigen und der Anzahl der gleichfarbigen Nachbarn einen bestimmten Wert δ überschreitet, so liegt die Zahl der benötigten Umfärbungen in $O(n)$.

4 Implementierung und Abbildungen

Wir haben die Algorithmen unter Verwendung der Bibliotheken LEDA, CGAL und Qt in C++ implementiert und empirisch analysiert. Als Testdaten haben wir sowohl reale Daten, die nach dem in [2] beschriebenen Verfahren u.a. für Wales erstellt wurden, als auch zufällige Daten verwendet. Auch bei den Verfahren, für die wir keine guten oberen Laufzeitschranken finden konnten, waren die Laufzeiten unkritisch. Abschließend noch ein paar Ausgaben der Programme.

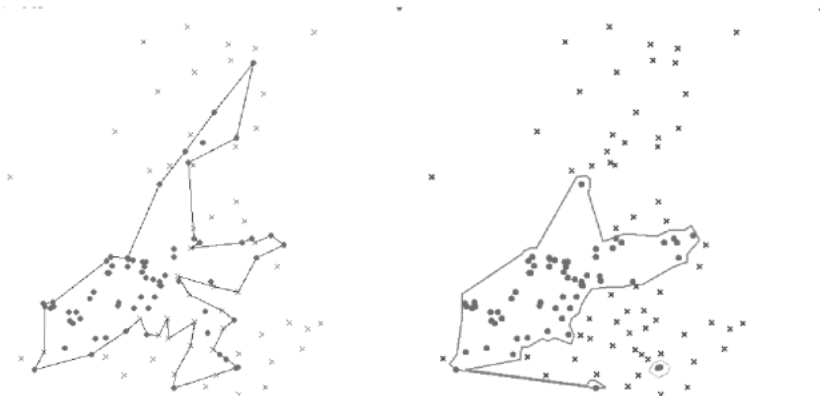


Abbildung 1: Berechnete Begrenzungen für den Datensatz Wales. Im linken Bild wurde die Anpassungsmethode mit $\alpha = 0.05$ verwendet, im rechten Bild die Umfärbungsmethode mit $\Phi = 200^\circ$.

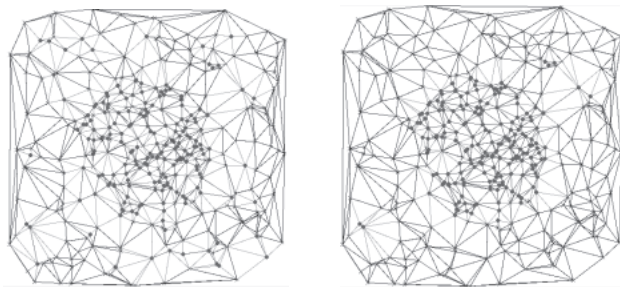


Abbildung 2: Ein- und Ausgabe der Umfärbungsmethode für $\Phi = 210^\circ$ auf einer nach gewissen Regeln zufälligen Punktmenge (abgebildet mit der zugehörigen Delaunay-Triangulation).

Literatur

- [1] H. Alani, C.B. Jones, and D. Tudhope. Voronoi-based region approximation for geographical information retrieval with gazetteers. *Int. J. Geographical Information Science*, 15(4):287–306, 2001.
- [2] A. Arampatzis, M. van Kreveld, I. Reinbacher, C.B. Jones, S. Vaid, P. Clough, H. Joho, and M. Sanderson. Web-based delineation of imprecise regions. manuscript, 2004.
- [3] E. M. Arkin, F. Hurtado, J. S. B. Mitchell, C. Seara, and S. S. Skiena. Some separability problems in the plane. In *Abstracts 16th European Workshop Comput. Geom.*, pages 51–54. Ben-Gurion University of the Negev, 2000.
- [4] Esther M. Arkin, S. Khuller, and Joseph S. B. Mitchell. Geometric knapsack problems. *Algorithmica*, 10:399–427, 1993.
- [5] Timothy M. Chan. Low-dimensional linear programming with violations. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, pages 570–579, 2002.
- [6] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.
- [7] P. Eades and D. Rappaport. The complexity of computing minimum separating polygons. *Pattern Recogn. Lett.*, 14:715–718, 1993.
- [8] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Heidelberg, West Germany, 1987.
- [9] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Trans. Inform. Theory*, IT-29:551–559, 1983.
- [10] G. David Garson and Robert S. Biggs. *Analytic Mapping and Geographic Databases*. Quantitative Applications in the Social Sciences. Sage University, 1992.
- [11] David O’Sullivan and David J. Unwin. *Geographic Information Analysis*. John Wiley & Sons Ltd, 2003.

Rokkatan - ein massiv mehrspielerfähiges Echtzeitspiel

Jan Hendrik Metzen, Alexander Ploss und Maraike Schellmann
Westfälische Wilhelms-Universität Münster
{jmetzen|plotzer|maraikes@uni-muenster.de}

1 Einleitung

Bei Rokkatan handelt es sich um ein *massiv mehrspielerfähiges Echtzeitspiel*, in dem eine große Anzahl Spieler - einige hundert - gleichzeitig an einer Online-Partie über das Internet teilnehmen. Um diese Anzahl von Spielern zu ermöglichen, basiert Rokkatan auf einer skalierbaren multi-Server Netzwerkarchitektur. Bei dem Spiel handelt es sich somit um ein komplexes verteiltes System, bei dessen Entwicklung verschiedene Problemstellungen wie Replikation und Konsistenzhaltung der Spieldaten oder Behandlung von Serverausfällen erfolgreich gelöst wurden.

Rokkatan wurde im Rahmen eines Projektseminars im Sommersemester 2004 von einer Gruppe von zwölf Studenten in Münster entwickelt und in C++ implementiert. Innerhalb des Projektteams gab es drei Teilgruppen, wovon eine für die Entwicklung der graphischen Darstellung des Spiels, die zweite für den Client und die dritte für die Server zuständig war. Im Anschluss an dieses Projektseminar wurde Rokkatan auf freiwilliger Basis weiterentwickelt.

2 Rokkatan: Das Spiel

Rokkatan spielt in einer mittelalterlichen Welt und stellt eine Mischung zwischen Rollen- und Strategiespiel dar. Der Rollenspielaspekt besteht darin, dass jeder Spieler die Rolle einer Spielfigur, seines *Avatars*, übernimmt. Diesen Avatar, der zum Beispiel ein Ritter oder ein Bogenschütze sein kann, steuert er in Echtzeit durch die zweidimensionale Spielwelt. Der Strategieaspekt kommt dadurch zustande, dass die Spieler in Teams organisiert sind und im Spielverlauf versuchen, ihrem Team zum Sieg zu verhelfen. Neben dem direkten Bekämpfen Mitglieder anderer Teams gibt es Flaggen in der Spielwelt, die zu besetzen und gegen andere Teams zu verteidigen sind. Durch besetzte Flaggen erhält ein Team Punkte, das Team mit der höchsten Punktzahl nach Ablauf der Spielzeit gewinnt die Partie.

In Rokkatan gibt es verschiedene Gegenstände, wie zum Beispiel Heiltränke, die aufgesammelt und benutzt werden können, um im Kampf verlorene Lebenspunkte wieder aufzufüllen. Des Weiteren gibt es verschiedene Geländetypen wie Wälder, Wiesen und Seen.

Der Spieler sieht jeweils nur einen Ausschnitt der Spielwelt abhängig von der Position seines Avatars. Auf [WM05] sind neben einer ausführlichen Beschreibung der Spielelemente von Rokkatan auch Bildschirmfotos verfügbar.

3 Spielimplementierung

Client-Server oder Peer-to-Peer Architekturen, welche üblicherweise in mehrspielerfähigen Echtzeitspielen eingesetzt werden, bieten nicht die erforderliche Skalierbarkeit für ein massiv-mehrspielerfähiges Spiel [SKH01]. Aus diesem Grund wurde Rokkatan auf Basis einer *Proxy-Server Architektur* [MFGM04] entwickelt, die wie in Abbildung 1 beispielhaft dargestellt für eine einzelne Spielpartie mehrere Spielserver, die so genannten Proxy-Server, einsetzt.

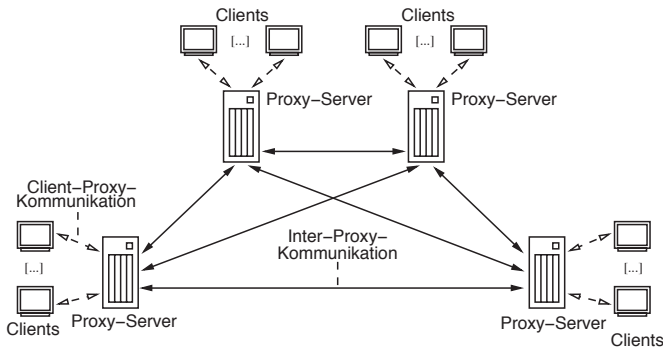


Abbildung 1: Beispielpartie in der Proxy-Server Architektur

In dieser Architektur wird der einzelne Server der Client-Server Architektur durch ein Netzwerk mehrerer Server, den Proxies, ersetzt. Jeder Client ist mit einem dieser Proxies verbunden. Der wesentliche Vorteil bei dieser Architektur besteht darin, dass sich die Kommunikations- und Rechenlast bei dem Betrieb einer Partie auf mehrere Proxies verteilt. Eine steigende Anzahl Clients lässt sich dadurch in einem gewissen Rahmen durch eine Erhöhung der Zahl der Proxies kompensieren.

Da auf jedem Proxy-Server der vollständige Spielzustand repliziert ist, wurde das *Eventual Consistency*-Modell zur Konsistenzhaltung der Spieldaten implementiert. Für jedes Element der Spielwelt wie Avatare, Heiltränke, Flaggen, etc. gibt es genau einen Proxy-Server, der den Zustand verändern darf. Nach einer solchen Änderung teilt dieser Server allen übrigen Proxies über die *Inter-Proxy-Kommunikation* den neuen Zustand des entsprechenden Elementes in einer Synchronisationsnachricht mit. Durch die Latenz in der Netzwerkkommunikation ist der Spielzustand nicht auf allen Proxies zu einem Zeitpunkt derselbe. Trotzdem ist diese Art der Synchronisation gut geeignet, da Zustandsänderungen sehr schnell erfolgen können und somit die dringend erforderliche, hohe Ansprechempfindlichkeit der Echtzeitanwendung gewährleistet ist.

Die Clients dienen dazu, dem jeweiligen Spieler die Spielwelt darzustellen, dessen Eingaben zu interpretieren und in Befehle für den zuständigen Proxy umzusetzen, welche mittels *Client-Proxy-Kommunikation* an diesen übertragen werden. Sämtliche Änderungen am Spielzustand werden von den Proxies entsprechend der von den Clients übermittelten Nutzerbefehle durchgeführt. Vor der Ausführung müssen die Befehle durch den Proxy auf Gültigkeit überprüft werden, da ein böswillig veränderter Client möglicherweise unerlaubte Aktionen sendet. Der entsprechende Spieler würde sich durch dieses *cheating* [Pri00] einen unfairen Vorteil verschaffen. In entgegengesetzter Kommunikationsrichtung reicht es aus, wenn ein Proxy den Clients nur Zustandsaktualisierungen für den für den Spieler sichtbaren Bereich übermittelt. Hierdurch wird die Kommunikationsbandbreite deutlich reduziert.

Ein weiterer Vorteil der Replizierung der Spielwelt ist, dass bei *Ausfall eines Proxies* keine Informationen verloren gehen. Ein in Rokkatan implementierter Mechanismus sorgt dafür, dass die Zuständigkeiten für die Elemente des ausgefallenen Servers auf die anderen Proxies verteilt wird. Lediglich die Spieler, deren Clients mit dem ausgefallenen Proxy verbunden waren, scheiden aus der laufenden Partie aus.

Echtzeitspiele erwecken für Benutzer den Eindruck, dass sich die Zeit in ihnen kontinuierlich fortbewegt. In dem Softwaresystem allerdings ist die Spielsimulation in kurze Runden, den so genannten *Ticks*, diskretisiert. Die Tickrate als Frequenz der Zustandsaktualisierungen ist in Rokkatan einstellbar. Um ein schnelles Ansprechverhalten zu erreichen sind ca. 25 Ticks/Sekunde nötig, was den üblichen Aktualisierungsfrequenzen schneller Actionspiele von kommerziellen Herstellern wie z. B. *Unreal Tournament* [Swe99] entspricht. In Rokkatan musste zudem dafür gesorgt werden, dass die Proxies möglichst *synchron* bleiben, die Zeit sich also auf einem Proxy nicht schneller fortentwickelt als auf einem anderen. Ist ein Proxy mit seinen Aufgaben fertig und die für diesen Tick vorgesehene Zeitspanne noch nicht verstrichen, so schläft der Prozess deswegen bis zum Tickende. Schafft er hingegen seine Aufgaben nicht in der vorgegebenen Zeitspanne, so wird versucht die Zeit in den nächsten Ticks aufzuholen. Falls dies dauerhaft nicht möglich ist, ist der entsprechende Server überlastet und kann die weichen Echtzeitanforderungen des Systems nicht mehr erfüllen, was zu einem ruckligen oder fehlerhaften Spielablauf auf den Clients des entsprechenden Servers führt.

Tabelle 1: Maximale Spielerzahl in Rokkatan

Anzahl Proxy-Server	1	2	3	4
Maximale Clientzahl	80	130	170	210

Stehen die Proxy-Server einer Partie kurz vor der Überlastungsgrenze, ist die maximale Zahl teilnehmender Clients erreicht. Die Proxy-Server Architektur ist jedoch durch Hinzunahme weiterer Server skalierbar. Tabelle 1 zeigt die ersten Testergebnisse der maximal möglichen Spielerzahlen. Wir arbeiten momentan an Tests mit höheren Proxyzahlen an einer Partie und unterschiedlichen, verschieden großen Spielwelten.

Literatur

- [MFGM04] Jens Müller, Stefan Fischer, Sergei Gorlatch und Martin Mauve. A Proxy Server-Network for Real-time Computer Games. In Marco Danelutto, Domenico Laforenza und Marco Vanneschi, Hrsg., *Euro-Par 2004 Parallel Processing*, Jgg. 3149 of *Lecture Notes in Computer Science*, Seiten 606–613, Pisa, Italy, August 2004. Springer-Verlag.
- [Pri00] Matt Pritchard. How to Hurt the Hackers: The Scoop on Internet Cheating and How You Can Combat It. *Game Developer*, June 2000.
- [SKH01] Jouni Smed, Timo Kaukoranta und Harri Hakonen. Aspects of Networking in Multi-player Computer Games. In *Proceedings of International Conference on Application and Development of Computer Games in the 21st Century*, Seiten 74–81, Hong Kong SAR, China, November 2001.
- [Swe99] Tim Sweeney. Unreal Networking Architecture <<http://unreal.epicgames.com/Network.htm>>, July 1999.
- [WM05] AG Parallele und Verteilte Systeme WWU Münster. Rokkatan <<http://wwwmath.uni-muenster.de/pvs/projects/rokkatan>>, Januar 2005.

E-Learning und Datenbanken

Marcel Stiller

HTWK Leipzig
Studiengang: Informatik
mstille@imn.htwk-leipzig.de

Art der Arbeit: praktisches Studiensemester (Bachelorarbeit)

Betreuer: Prof. Dr.-Ing. Thomas Kudraß

Überblick: Diese Arbeit befasst sich mit der Entwicklung des WBT-Systems ADIA (Anbindung von Datenbanken in Internet-Anwendungen), durch welches die Erstellung dynamischer Webseiten vermittelt wird. Das System ist so konzipiert, dass sowohl Wissen präsentiert werden kann, aber auch Aufgaben zum Üben bereit stehen. Diese bestehen aus der Ergänzung eines Quelltextes, welche vom Nutzer durchgeführt wird. Die Aufgaben werden serverseitig kompiliert und der Nutzer bekommt das Ergebnis seiner Arbeit sofort im Browser angezeigt. Hierbei handelt es sich um Webseiten, deren Inhalte laufzeitdynamisch aus einer Datenbank erzeugt werden.

1. Motivation und Idee

Mit der immer stärkeren Verbreitung des Internet und der zunehmenden Anzahl dynamischer Webseiten wird es immer wichtiger diese Technologien einem breiten Kreis von Anwendern und Entwicklern zu vermitteln. Dies ist aber immer mit hohem Aufwand verbunden, da passende Datenbanksoftware client- wie serverseitig installiert sein muss. Außerdem sind solche Anwendungen dann meist nur über ein Intranet z.B. an der Uni nutzbar, nicht aber von zu Hause aus. Zum Üben am heimischen PC bliebe nichts anderes übrig, als eine eigene Datenbank zu installieren. Die Motivation von ADIA besteht darin eine Lernumgebung zu entwerfen, die nicht viel mehr als einen Browser benötigt, aber trotzdem innerhalb der Übungsaufgaben eine Datenbank für Datenabfragen zur Verfügung stellt. Der einzige Aufwand dabei ist, ADIA auf einem Webserver zu installieren und verfügbar zu machen. Zum anderen besteht das Ziel darin, ein dynamisches System zu entwerfen, welches nicht nur für die Ausbildung von Informatikern geeignet ist, sondern auch für andere Fachrichtungen. Das System kann durch einen Administrator mit beliebigen Inhalten gefüllt werden, wodurch ermöglicht wird es inhaltlich individuell anzupassen. Zudem ist ein weiteres Ziel, für die Inhalte so viele Dateitypen wie möglich zu unterstützen. Damit wird erreicht, dass vorhandene Lernobjekte genutzt und integriert werden können. Außerdem muss auch bei den Aufgabentypen ein breites Spektrum abgedeckt werden, wodurch die Vermittlung und Übung unterschiedlichster DB-Schnittstellen zu Programmiersprachen möglich wird. Diese Ideen bilden die Grundlage zur Entwicklung von ADIA.

2. Funktionalität

ADIA ist ein Mehrbenutzersystem, d.h. unterschiedliche Benutzer arbeiten in ihrer eigenen Datenbank. Es besteht aus einer Schnittstelle für den Administrator und einer für den Lernenden. Strukturell besteht ADIA aus folgenden Elementen: Das Wurzelement ist der *Kurs* (courses), welcher wiederum *Lektionen* (lessons) enthalten kann. Innerhalb einer Lektion können nun *Vorlesungen* (lectures), *Aufgaben* (tasks), *Referenzen* (references) und *Fragebögen* (questionnaires) eingebunden werden. Zu einer Aufgabe gehören außerdem noch *Hinweise* (advices), welche dazu dienen, ein DB-Schema zu erläutern, allgemeine Tipps oder Lösungsansätze zu den Aufgaben zu geben. Diese Struktur ist in Abb. 1 kurz dargestellt und kann je nach Bedarf mit dynamischen Inhalten durch den Administrator gefüllt werden (vgl. Abb. 2). Des Weiteren stehen unterschiedliche Aufgabentypen, wie Java Servlets und PSP (PL/SQL Server Pages), zur Verfügung. Zu jedem Aufgabentyp können Aufgaben angelegt werden. Diese bestehen aus einem Quelltext, der Lücken aufweist, welche durch den Lernenden vervollständigt werden sollen. Dabei kann er innerhalb dieser Aufgaben Datenbankabfragen stellen, wodurch die Webseite mit dynamischen Inhalten gefüllt wird. Er ist außerdem in der Lage die Aufgabe serverseitig zu kompilieren und sich das Ergebnis im Browser anzeigen zu lassen. Die Kompilierung ist allerdings nur eine syntaktische Prüfung; ob die Aufgabe logisch richtig ist muss der Lernende selbst entscheiden. Um diese Entscheidung zu treffen kann er seine Lösung mit einer Musterlösung vergleichen, welche als Ausgabe im Browser oder als Quelltext zur Verfügung steht. Die anderen Elemente wie z.B. Vorlesung sind so konzipiert, dass sie schon vorhandene Materialien einbinden können, unterstützt werden Formate wie JPEG, HTML, PDF und andere.

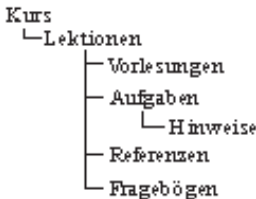


Abbildung 1: Struktur

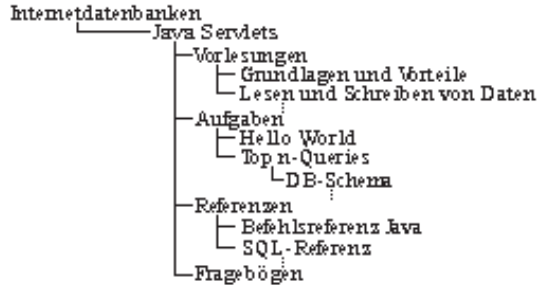


Abbildung 2: Ausschnitt aus einem Beispiel

3. Umsetzung

3.1 Systemvoraussetzungen

ADIA wurde in Flash MX 6.0 entwickelt, an wenigen Stellen wird dies durch Javascript unterstützt. Als Webserver wird Apache genutzt und es liegt eine Oracle 9i Datenbank (Release 1) zu Grunde. Zur Kommunikation werden Java Servlets in Kombination mit XML verwendet. Die Servlets nutzen die Oracle Servlet Engine. Auf der Clientseite wird lediglich ein Browser benötigt und der aktuelle Flashplayer.

3.2 Aufbau und Kommunikation

Um ADIA auch später noch modifizieren zu können, ist es wichtig, Präsentationslogik und Kommunikation strikt zu trennen. Das System besteht aus der Oberfläche, Klassen zum Verarbeiten von XML-Dokumenten, Java Servlets und der Datenbank. Innerhalb der Oberfläche gibt es verschiedene Elemente, die Daten präsentieren können, z.B. Lektionen, Vorlesungen oder Aufgaben. Jedes dieser Oberflächenelemente muss mit Inhalt gefüllt werden, aber auch Inhalte an die Datenbank senden können. Für diese Kommunikation stehen die Kommunikationsklassen zur Verfügung. Jede dieser Klassen ist für die Kommunikation mit einer Tabelle in der Datenbank gedacht. Dazu muss man nur die passende Klasse einbinden und deren Methoden verwenden. Jede dieser Klassen ermöglicht das Lesen, Schreiben, Löschen und Updaten von Datensätzen. Da es keine direkte Kommunikation zwischen Flash und Oracle gibt, wurden hier Java Servlets als Middleware eingesetzt. Diese lesen auf Anfrage Daten aus Tabellen und generieren XML-Dokumente, die von den Kommunikationsklassen weiterverarbeitet werden. Außerdem können Servlets XML-Dokumente erhalten und deren Inhalt zur Modifikation der Datenbank nutzen. Besteht nun der Wunsch, ADIA um neue Elemente zu erweitern, muss man zuerst eine neue Tabelle in der Datenbank hinzufügen. Als nächstes müssen die Servlets erweitert werden und danach eine Klasse zur Kommunikation mit Flash erstellt werden. Ist dies geschehen, kann die Oberfläche ergänzt werden und mit Hilfe der neuen Klasse mit Inhalt gefüllt werden.

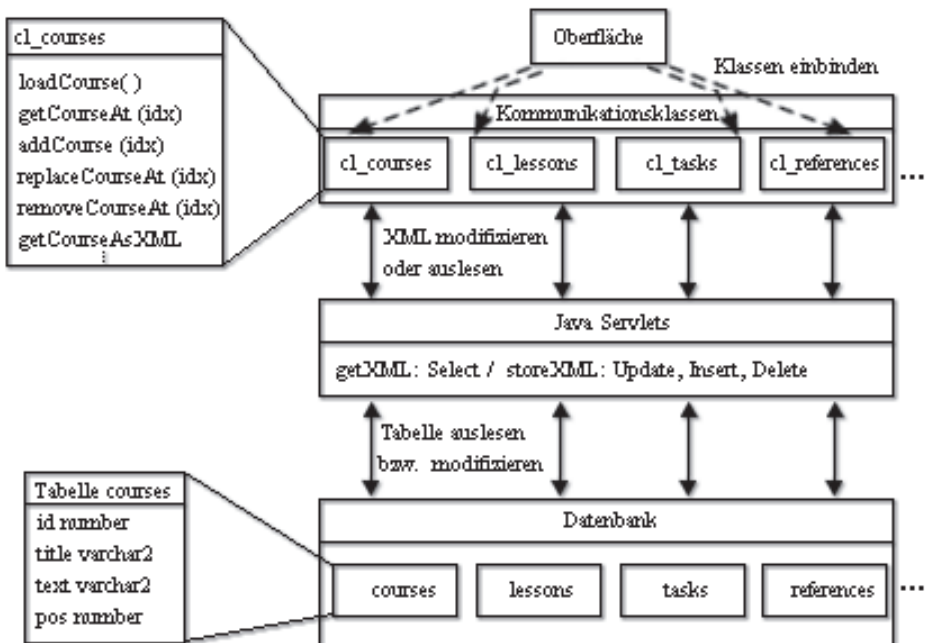


Abbildung 3: Ausschnitt des Aufbaus und der Kommunikation zwischen Flash und Datenbank.

3.3 Kompilierung der Aufgaben

Innerhalb von ADIA legt der Aufgabentyp (Java Servlets, PSP etc.) fest, wie eine Aufgabe kompiliert wird. Um dies durchzuführen, sendet der Client Benutzernamen, Aufgabe und ihren Typ an ein Perl Skript. Dieses Skript ruft ein weiteres auf, welches den entsprechenden Typ verarbeiten kann. Das Skript kompiliert die Aufgabe, lädt sie danach in die Servlet Engine und veröffentlicht diese, damit der Nutzer sie ansehen kann. Treten bei der Kompilierung Fehler auf, werden diese gesammelt, um sie dem Nutzer anzuzeigen. Ist die Abarbeitung der Skripte beendet, wird das dem Client gesendet und das Ergebnis kann nun im Browser angezeigt werden. War die Abarbeitung nicht erfolgreich, wird das Fehlerprotokoll zurück gegeben, was dann vom Nutzer ausgewertet werden kann.

4. Weiterentwicklung

In der Zukunft soll ADIA um neue Aufgabentypen erweitert werden. Schwerpunkte hierbei sind die Einbettung von SQL in XML-Dokumente (XSQL/Oracle), sowie XPath-Ausdrücke (innerhalb von XSLT). Dafür muss der Inhalt einer schon existierenden Vorlesung (XML-Datenbanken) integriert werden. Außerdem wird ADIA um eine Feedbackfunktion bezüglich der Fragebögen erweitert. Diese wird aber nicht im Sinne einer Lernfortschrittskontrolle eingebunden, sondern als Selbstkontrolle für den Lernenden. Es soll dann möglich sein, dem Nutzer anzuzeigen, welche Aufgaben er schon bearbeitet hat und eine Vergleichsmöglichkeit seiner gelösten Multiple Choice Aufgaben bieten. Dadurch wird ein einfacher Überblick über den eigenen Stand möglich und auch die Selbstkontrolle für Multiple Choice realisiert. Somit sollte es künftig gelingen, über den Kreis der Informatik-Studenten hinaus, einen E-Learning-Service auch für Nicht-Informatiker anzubieten, die für ihre Anwendungen Kompetenzen in Web-Datenbanktechnologie erwerben müssen.

Literaturverzeichnis

- [Dic00] E. Dick: Multimediale Lernprogramme und telematische Lernarrangements, Bildung und Wissen, 2000
- [Häh04] S. Hähnel, Eine Lernumgebung für Webdatenbanken Informatiktage 2003, Konradin Verlagsgruppe, 2004
- [HW02] B. Hall, S. Wan, Object-Oriented Programming with Actionscript. New Riders Publishing, 2002
- [Mue00] S. Muench, Oracle XML Applications, O'Reilly, 2000

Entwicklung eines WYSIWYG Editors für das Erstellen von Lehrmaterial im XML Format

Thomas Weise

TU Chemnitz
Fakultät für Informatik
tweise@gmx.de

Art der Arbeit: Projekt
Betreuer/in der Arbeit: Dr. Elke Wällnitz

Abstract: In der Arbeit wird ein WYSIWYG-Editor zum Erstellen von Lehrmaterial für das E-Learning vorgestellt. Es wird ein auf XML basierendes Speicherformat genutzt. Dieses ermöglicht das Auszeichnen der Texte mit semantischen Markierungen, das Anbinden von Multimedia-Objekten und Simulationen. Das erstellte Lehrmaterial kann nach didaktischen Gesichtspunkten umgeordnet und in andere Formate wie XHTML und PDF konvertiert werden.

1 Einleitung

Im Februar 2001 startete das Projekt „Wissenswerkstatt Rechensysteme“, kurz WWR, an dem sich zwölf deutsche Hochschulen beteiligen. Im Rahmen dieses Projekts wird ein internetbasiertes System von multimedialen Lehr- und Lernmodulen zur Unterstützung der Aus- und Weiterbildung im Wissensgebiet Technische Informatik aufgebaut. [wwr]. Um die Verständlichkeit und Anschaulichkeit der neuen Lehrmaterialien zu erhöhen, liegt der Schwerpunkt auf dem Einsatz von Simulationen und Animationen.

Die Speicherung des Inhalts der Lehrinhalte der WWR erfolgt in XML-Dokumenten. XML ist ein Datenformat, welches erst durch die Umwandlung in andere Formate graphisch dargestellt wird. Durch die Trennung von Daten und Darstellung kann jedoch auch kein allgemeiner XML Editor in der Art eines WYSIWYG¹ - Textverarbeitungssystems existieren. Bereits für im Umgang mit XML geschultes Personal ist es zeitraubend, längere XML-Texte „von Hand“ zu verfassen. Der Einsatz von XML allein ist in Gebieten außerhalb der universitären Informatik praktisch nicht realisierbar. Damit ein XML-Dialekt breitflächig zum Einsatz kommen kann, muss er in Verbindung eines speziell zugeschnittenen WYSIWYG-Editors angeboten werden. Das Erstellen eines solchen Editors ist der Gegenstand dieser Arbeit.

¹ What You See Is What You Get - Editoren erlauben es, Dokumente graphisch zu bearbeiten

2 KML

Die Lehrmaterialien der Wissenswerkstatt Rechensysteme werden in dem speziellen XML-Dialekt $\langle ML \rangle^{3,2}$ gespeichert. Durch seinen großen Umfang mit über 150 Sprachelementen [ml3xsd] ist die Entwicklung eines WYSIWYG-Editors für $\langle ML \rangle^3$ im Rahmen dieser Arbeit nicht möglich. Stattdessen wird ein ähnlicher XML-Dialekt, KML (Knowledge Markup Language), entwickelt. Er implementiert ein sinnvolles Subset der $\langle ML \rangle^3$ -Funktionalität. Alle folgenden Eigenschaften von KML sind an $\langle ML \rangle^3$ angelehnt.

semantische Gliederung

Es ist möglich, Teile des Inhalts der KML-Dokumente semantisch zu Klassifizieren. Markierungen wie „Beschreibung“, „Definition“, „Anmerkung“, „Satz“ und Ähnliches können für einzelne Abschnitte gesetzt werden.

logische Gliederung

Wissensgebiete werden in Module eingeteilt. Diese Module können in einer Baumhierarchie in weitere Untereinheiten gegliedert werden.

didaktische Gliederung

Aus der eingegebenen inhaltlichen Materialbasis wird das individuelle Lehrmaterial zusammengestellt. Dies erfordert oftmals eine didaktische Umordnung. KML ermöglicht dies. Damit die ursprüngliche Zusammenstellung des Lehrmaterials erhalten bleibt, wird die didaktische Gliederung getrennt von der der logischen Gliederung gespeichert.

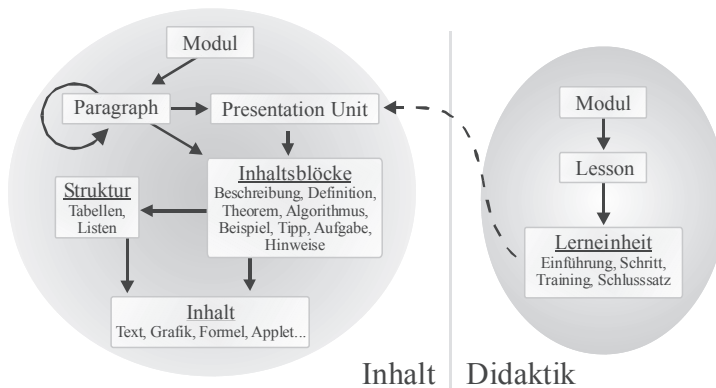


Abbildung 1: logische und didaktische Struktur von KML [kml]

² Multidimensional Learning Objects and Modular Lectures Markup Language

Markierung der Intensität

An Lehreinrichtungen existieren Kurse oftmals in verschiedenen Intensitätsebenen. Ein Kurs „Grundlagen der Informatik“ könnte für Informatik-, Mathematik- und Mediengestaltungstheorie-Studenten angeboten werden. Die drei Kurse haben das gleiche Thema, jedoch wird unterschiedlich tief in die Materie vorgedrungen. Der Lehrinhalt kann mit KML mit den Niveaus „Expert“, „Advanced“ und „Basic“ markiert werden.

Unterscheidung von Material für Lehrer und Schüler

Mit KML können Lehrer und Schüler verschiedenes Unterrichtsmaterial nutzen. Dadurch ist es möglich, spezielle Anmerkungen oder Lösungen von Aufgaben nur für Lehrer sichtbar zu machen.

verschiedene Ausgabeformate

KML-gemäßes XML kann in verschiedene Ausgabeformate konvertiert werden. Folien/Projektionen, Webseiten und Papierskripten stehen als Medien zur Verfügung.

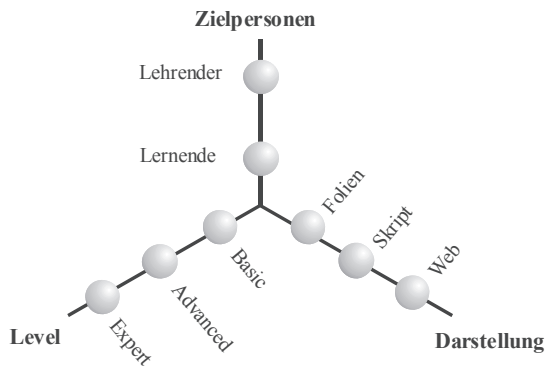


Abbildung 2: das dreidimensionale Modell von KML

3 WYSIWYG-Editor

Mit Hilfe der Programmiersprache Java wird ein auf Swing [javb] basierender Editor für den XML-Dialekt KML entwickelt. Dabei wird besonders auf folgende Entwurfsmerkmale Wert gelegt: WYSIWYG-Fähigkeit, Ausrichtung auf die Lehre, Speicherung der Daten im XML-Format, Anbindung von Filtern und Transformatoren zur Umwandlung der Daten in webfähige, visuelle Formate, Kosteneffizienz, Einfachheit, Verfügbarkeit, Erweiterbarkeit, Wartbarkeit und Wiederverwendbarkeit.

Literaturverzeichnis

- [javb] Guido Krüger: Handbuch der Java-Programmierung, 3. Auflage
HTML-Ausgabe 3.0.1 · © 1998-2003 Guido Krüger
Addison-Wesley, 2002, ISBN 3-8273-1949-8
www.javabuch.de, www.gkrueger.com, hjp3@gkrueger.com
- [kml] Dr. Elke Wällnitz: Der plattformunabhängige KML-Editor als Werkzeug zur
Entwicklung von E-Learning-Modulen auf der Basis von XML
- [ml3xsd] XSD-Schemata von <ML>³
<http://www.ml-3.org/>
<http://www.ml-3.org/ML3/1.2/Didactic>,
<http://www.ml-3.org/ML3/1.2/Content>
- [wwr] Prof. Dr.-Ing. habil. Tavangarian, Djamshid: Wissenswerkstatt Rechensysteme
Ein bundesweites Vorhaben zur Erstellung eines Baukastensystems von multimedialen
Lehr- und Lern-Modulen im Bereich Technische Informatik
<http://www.wwr-project.de>

Entwicklung eines Softwareframeworks für audiovisuelle digitale Medien im Internet

Thomas Sporer, Tino Jahnke, Anton Köstlbacher

Projekt Knowledgebay
Universität Regensburg
c/o Lehrstuhl für Informationswissenschaft
93040 Regensburg
{thomas.sporer, tino.jahnke, anton.koestlbacher}@knowledgebay.de

Abstract: Die beschriebene Software stellt die Basis für den Betrieb des Informationsdienstes von Knowledgebay (www.knowledgebay.de) dar. Sie wurde für die Veröffentlichung, Darbietung und Verbreitung von audiovisuellen digitalen Medieninhalten über das Internet spezialisiert. Der vorliegende Beitrag stellt zunächst den Prozess der Prototypenentwicklung des Softwareframeworks vor. Dann wird das entstandene Softwareprodukt näher beschrieben.

1. Einleitung

Mit dem Projekt Knowledgebay wurde von einer interdisziplinär zusammengesetzten Gruppe von Studierenden ein Informationsdienst entwickelt, der audiovisuelle Wissensmedien im Internet anbietet. Ziel des Projekts ist die Erweiterung des Zugangs zum Wissen von Hochschulen durch die Nutzung moderner Informations- und Kommunikationstechnologien [SJW03]. Beim hier vorgestellten Softwareframework handelt es sich um die informationstechnische Basis des Projekts.

2. Entwicklungsprozess des Softwareframeworks

Bei der Realisierung des Softwareframeworks handelt es sich um ein studienbegleitendes Entwicklungsprojekt, das über mehrere Semester hinweg realisiert wurde. Als Ergebnis der Pilotphase des Projekts lässt sich ein Softwareprototyp festhalten, der sich seit Sommersemester 2003 im Echtbetrieb befindet und bereits evaluiert wurde.

Das Softwaresystem entstand nach einem evolutionären Entwicklungsmodell [PB02] und hat sich aus den spezifischen Anforderungen des Projekts heraus entwickelt. So konnte bereits in einer sehr frühen Phase des Projekts ein lauffähiges Teilsystem genutzt und in der Praxis getestet werden. Die Basisfunktionalitäten der Software wurden dann entsprechend dem Bedarf der (Test-)Nutzer des Systems angepasst und um weitere Systemkomponenten erweitert.

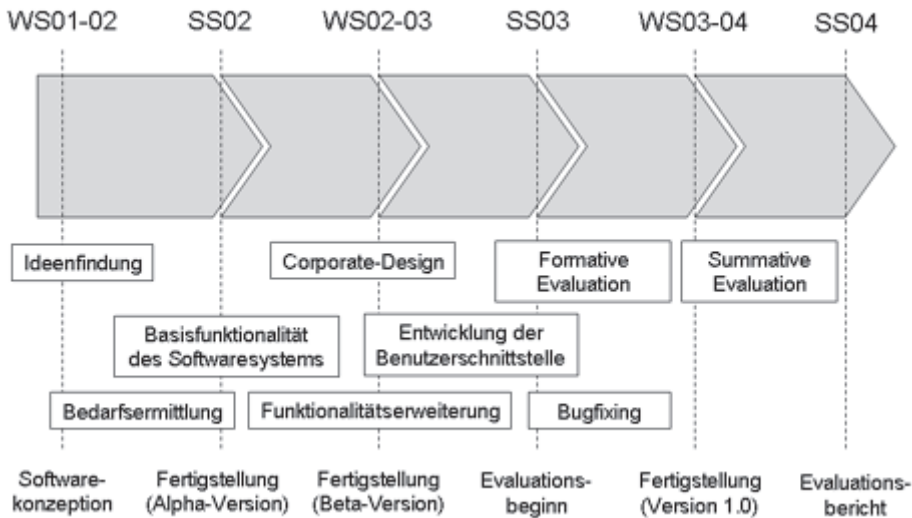


Abbildung 1: Darstellung des Entwicklungsprozesses

Während der gesamten Prototypenphase fand ein intensiver Austausch zwischen den Nutzern und den Entwicklern des Systems statt, so dass die Software bereits im Entwicklungsprozess formativ evaluiert sowie in einem iterativen Prozess fortlaufend optimiert und weiterentwickelt wurde [BR99]. Im Verlauf dieses sequentiellen Software-Life-Cycle [PB02] wurde im Herbst 2004 ein zusammenfassender Evaluationsbericht fertig gestellt.

3. Produktbeschreibung des Softwareframeworks

Das Softwareframework kann als so genanntes Community-Content-Collaboration-Management-System (C3MS) verstanden werden [BK04]. Ein mögliches Einsatzszenario lässt sich nach dem Dienstemodell einer Notebook-Universität [Ker04] beschreiben. Das Softwaresystem stellt im NBU-Szenario einen spezifischen Informationsdienst für audiovisuelle Lehr-Lernmedien auf einem digitalen Campus bereit.

Die Software unterstützt dabei die Prozesse der verteilten, orts- und zeitunabhängigen Produktion und Nutzung der auf dem Internetportal des Projekts Knowledgebay veröffentlichten Medieninhalte. Das Backendsystem stellt dabei die Basis für die Inhaltsproduktion dar, während das Frontendsystem den Fokus auf Nutzung der Medien des Archivs legt.

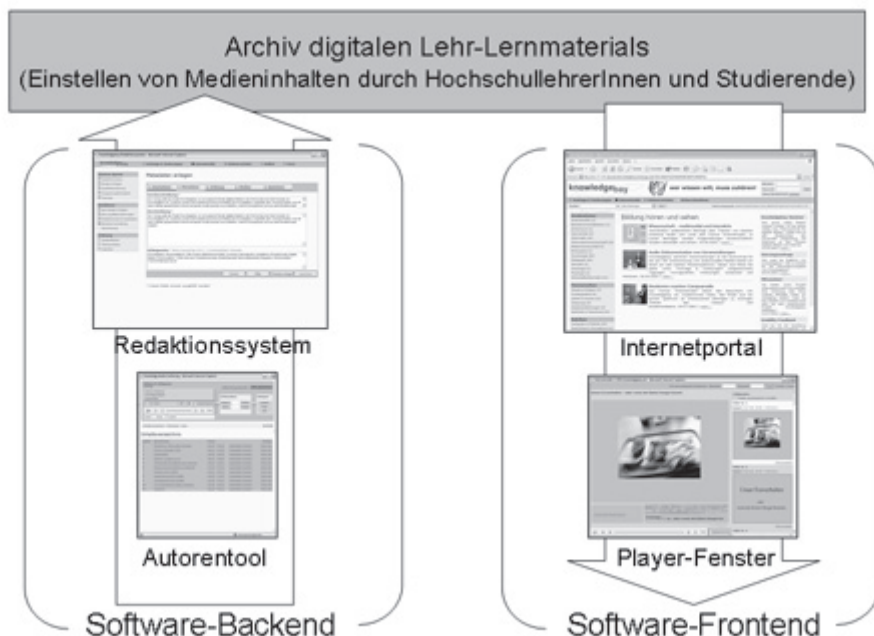


Abbildung 2: Zusammenhang der einzelnen Systemkomponenten

Das Softwareframework besteht aus fünf sich ergänzenden Systemkomponenten:

- Online-Autorentool: Produktion von inhaltlich-multimedial aufbereiteten Streaming-Dokumenten als Rich-Media-Applikation
- Webbasiertes Redaktionssystem: Anlegen von neuen Medieninhalten und Verwalten bestehender Medien im Online-Archiv
- Digitales Medienarchiv: Repository der im Rahmen des Projekts produzierten und mit Metadaten ausgezeichneten Medien
- Internetportal: Schnittstelle zum einfachen und strukturierten Zugriff der Portalbenutzer auf die Inhalte des Medienarchivs
- Player-Fenster: Wiedergabe der Inhalte als Rich-Media-Applikation mit Möglichkeiten zur Interaktion der Benutzer

Das Softwaresystem basiert auf einer Kombination von Web-, Datenbank- und Streaming-Technologien. Es kommen nachstehende technische Standards zum Einsatz:

- HTML/SMIL/CSS: Gestaltung der grafischen Benutzerschnittstelle des Portals und Darbietung der audiovisuellen Inhalte als Rich-Media-Applikation
- PHP4/SQL92: Dynamische Generierung der Portalseiten aus der Datenbank des Inhaltsarchivs und serverseitig automatische Erzeugung der SMIL-Dateien
- Javascript/ActiveX: Interaktive Steuerung und Automatisierung der Medienwiedergabe sowie zur Aufbereitung der Medien im Authoringtool
- RDF/RSS: Einbindung externer Newsfeeds auf der Startseite des Portals und zur Dissemination der Archivinhalte an Dritte

Die Basis des Systems besteht aus verschiedenen OpenSource-Softwareprodukten (z.B. Apache, MySQL), die in der beschriebenen Softwarelösung integriert sind.

4. Weiterentwicklung und Ausblick

Künftig soll die Software entsprechend den Evaluationserkenntnissen neu implementiert und weiterentwickelt werden. Die Zielsetzung der Weiterentwicklung ist insbesondere Schnittstellen zu einer Reihe weit verbreiteter Learning-Management-Systeme zu schaffen, um die Software kompatibel zu bereits an vielen Hochschulen vorhandenen eLearning-Lösungen zu machen [DVE02].

Literaturverzeichnis

- [AG89] Aschersleben, G.; Gstalter, H.; Kaiser, F.; Strube, V.; Zang-Scheucher, B. (1989). Prototyping als Verfahren zur Software-Entwicklung. *Zeitschrift für Arbeitswissenschaft*, 43, 42-47.
- [BK04] Baumgartner, P.; M. Kalz (2004). Content-Management-Systeme aus bildungstechnologischer Sicht. In: Content Management Systeme in e-Education. Auswahl, Potenziale und Einsatzmöglichkeiten. Innsbruck-Wien, StudienVerlag.
- [BR99] Bauer, D. ; Röckmann, D. (1999). *Methoden der Bewertung von Software*. Heidelberg: dpunkt-Verlag.
- [DVE02] Doberkat, E.-E.; Veltmann, C.; Engels, G.; Hausmann, J.H.; Lohmann, M. (2002): Anforderungen an eine eLearning-Plattform. Innovation und Integration. URL: <http://eldorado.uni-dortmund.de:8080/FB4/ls10/memos/122/122.pdf> [1.3.2005]
- [Heg03] Hegner, M.: *Methoden zur Evaluation von Software*. IZ-Arbeitsbericht Nr. 29, Bonn: InformationsZentrum Sozialwissenschaften (ASI).
- [SJW03] Sporer T.; Jahnke T.; Walter G.; Erbacher C.; Köstlbacher A.; Riecks A.: Knowledgebay – Projektvorstellung auf der Campus Innovation 2003. URL: www.knowledgebay.org/publikationen/campusinnovation2003.pdf [1.3.2005]
- [Ker04] Kerres, M. (2004). Zur Integration digitaler Wissenswerkzeuge in die Hochschule. In: E. Kruse, U. Küchler, M. Kuhl (Hrsg.): Unbegrenzt Lernen – Lernen über Grenzen? Generierung und Verteilung von Wissen in der Hochschulentwicklung. Münster: LIT-Verlag
- [PB02] Pomberger, G., Blaschek, G. (2002). Software Engineering – Prototyping und objektorientierte Software-entwicklung. München, Wien. Hanser Fachbuch Verlag

Konzeption und Entwicklung von O.S.I.R.I.S. (Open source basiertes integriertes regionales Informationssystem für Virtuelle Rathäuser)

Hinzen, M.; Meesters, S.; Müller, T.; Niemeyer, G.; Robert, S.

Westfälische Wilhelms-Universität Münster
Lehrstuhl für Wirtschaftsinformatik und Informationsmanagement

[mhinzen | meesters | mullerto | niemeyer | srobert]@uni-muenster.de

Art der Arbeit: Projektseminar mit Praxispartnern
Betreuer der Arbeit: Prof. Dr. J. Becker

Abstract: Das Projektseminar „ProService“ des Lehrstuhls für Wirtschaftsinformatik und Informationsmanagement der Westfälischen Wilhelms-Universität Münster hat in Zusammenarbeit mit dem Kreis Warendorf und dem Kommunalrechenzentrum citeq eine umfassende Anforderungsanalyse mit anschließender Konzeption eines Virtuellen Rathauses durchgeführt. Um zu zeigen, dass die ermittelten Anforderungen Ziel führend und umsetzbar sind, wurde ein Prototyp des Virtuellen Rathaus unter dem Namen O.S.I.R.I.S. (Open Source basiertes Integriertes Regionales InformationsSystem für Virtuelle Rathäuser) entwickelt.

1 Motivation und Zielsetzung

Kommunalverwaltungen bedienen sich des Mediums Internet, um Bürgern und Unternehmen zu jeder Zeit und an jedem Ort die Möglichkeit zu bieten, Informations-, Kommunikations- und Transaktionsdienstleistungen der Verwaltung in Anspruch nehmen zu können. Bevor Verwaltungsdienstleistungen vollständig digital und medienbruchfrei sowohl bei der externen Inanspruchnahme als auch bei der internen Bearbeitung durchgeführt werden können, sollten den Anspruchsgruppen zunächst in adäquater Weise qualitativ hochwertige Informationen und Kommunikationsmöglichkeiten zu den Verwaltungsdienstleistungen zur Verfügung gestellt werden. Das „Virtuelle Rathaus“ ist probates Mittel zur Erreichung dieser Zielsetzung.

Die Studie „Virtuelles Rathaus Münsterland 2004“ hat gezeigt, dass bei den derzeitigen Realisierungen im Münsterland einige Leistungslücken erkennbar sind [Be04]. Neben einem umfangreicheren Angebot onlinezugreifbarer Dienstleistungen, fehlt es vor allem an einer strukturierten und einheitlichen Beschreibung dieser, sowie an einheitlichen und intuitiv verständlichen Navigationskonzepten. Weiterhin wurde festgestellt, dass gerade

kleine und mittlere Verwaltungen über keine ausgereiften Konzepte, Ressourcen und informationstechnische Unterstützung für die Realisierung eines Virtuellen Rathauses verfügen.

Das Projektseminar „ProService“ des Lehrstuhls für Wirtschaftsinformatik und Informationsmanagement der Westfälischen Wilhelms-Universität Münster hat daraufhin in Zusammenarbeit mit dem Kreis Warendorf und dem Kommunalrechenzentrum citeq im Rahmen des Projektes Regio@KomM eine umfassende Anforderungsanalyse in Bezug auf das Virtuelle Rathaus durchgeführt [Al04]. Als zentrales Anliegen auf Seiten der Kommunen kristallisierte sich dabei heraus, dass gerade kleine und mittlere Kommunen dem immensen Pflegeaufwand nicht gewachsen sind und die entstehenden Kosten vorhandener kommerzieller Systeme nicht zu tragen bereit sind. Sämtliche Anforderungen wurden in einem umfangreichen Anforderungsdokument strukturiert zusammengetragen. Als konzeptionelle Basis der Implementierung eines Virtuellen Rathauses wurde ein produktunabhängiges Fachkonzept erstellt.¹ Auf dessen Grundlage wurde anschließend O.S.I.R.I.S. (Open Source basiertes Integriertes Regionales Informationssystem für Virtuelle Rathäuser) entwickelt.

2 Funktionalitäten von O.S.I.R.I.S.

Als domänenspezifische, umfangreiche Erweiterung des Content Management Systems (CMS) Typo3 beruht O.S.I.R.I.S. vollkommen auf frei und kostenlos verfügbare Technologien, wie der Datenbank MySQL und der Skriptsprache PHP und stellt somit eine kostengünstige Möglichkeit zum Aufbau und Betrieb eines Virtuellen Rathauses dar. Typo3 als Grundlage bietet alle Vorzüge eines modernen CMS. Dazu zählen beispielsweise die konsequente Trennung von Inhalt, Layout und Struktur sowie eine webbasierte, dezentrale Inhaltspflege. O.S.I.R.I.S. richtet die Inhaltspflege konsequent auf die Domäne der Verwaltung aus, indem passende Contenttypen, wie Mitarbeiter, Dienstleistung oder Gebäude, direkt mitgeliefert werden und somit der Customizingaufwand auf ein Minimum reduziert wird.

Das Virtuelle Rathaus O.S.I.R.I.S. lässt sich in den für den Bürger sichtbaren Front-End-Bereich und in den verwaltungsinternen Back-End-Bereich untergliedern. Der Front-End-Bereich stellt Bürgern und Unternehmen Informationen über die Verwaltungsdienstleistungen barrierefrei zur Verfügung. Dabei können die Inhalte über verschiedene Navigationskonzepte erreicht werden. Neben einer alphabetischen Auflistung, werden Dienstleistungen nach Nutzergruppen und Lebenslagen gruppiert bzw. nach der Häufigkeit der bisherigen Nachfrage aufgelistet. Im Back-End-Bereich werden die Dienstleistungen gepflegt, in die bestehenden Strukturen eingebunden sowie Dienste für Verwaltungsmitarbeiter und Administratoren zur internen Nutzung und Pflege angeboten.

¹ Beide stehen interessierten Lesern als Arbeitsberichte (Nr. 108 und 109) des Instituts für Wirtschaftsinformatik der WWU Münster kostenlos unter <http://www.regio-komm.de> zur Verfügung.

Auf Grund des Ziels durch O.S.I.R.I.S. vor allem kleinen und mittleren Kommunen das Angebot an Bürgerservices in Form eines Virtuellen Rathauses zu erleichtern, wurde großer Wert auf die Möglichkeit eines zentralen Hostings durch Rechenzentren gelegt. In diesem Rahmen wurde mit der Erweiterung zu einer mandantenfähigen Lösung das Hosting mehrerer Kommunen auf einer Instanz ermöglicht. Somit werden Verwaltungen vom technischen Betrieb entbunden und können sich auf die reine Pflege der Inhalte konzentrieren.

Diese Integration von mehreren Kommunen in einem System schafft die Voraussetzung für eine interkommunale Zusammenarbeit der Verwaltungen. Allgemeine Beschreibungstexte, notwendige Unterlagen, Gebühreninformationen, Gesetzesgrundlagen etc. sind bei identischen Dienstleistungen verschiedener Verwaltungen gleich. Um die mehrfache Pflege gleichartiger Informationen zu vermeiden, können diese von einer Kommune in einem so genannten Musteranliegen zentral hinterlegt und anderen Verwaltungen zur Verfügung gestellt werden. Diese arbeitsteilige Erfassung und Pflege von Musteranliegen ermöglicht es Kommunen erhebliche Arbeitersparnis zu realisieren. O.S.I.R.I.S. unterstützt diesen Prozess durch eine systemseitig integrierte Workflowkomponente, die eine mandantenübergreifende Pflege gemeinsam genutzter Inhalte gestattet und sich somit von anderen Softwarelösungen für Virtuelle Rathäuser abhebt.

3 Einsatz von O.S.I.R.I.S.

Im Sinne des Open Source Gedankens steht O.S.I.R.I.S. allen Kommunalverwaltungen, Rechenzentren und anderen Interessierten zusammen mit einer umfassenden Anwendungs- und Entwicklungsdokumentation als Typo3-Extension „civserv“ zur freien Verfügung.² Schon kurz nach der Fertigstellung konnte O.S.I.R.I.S. deutschlandweit auf großes Interesse stoßen und einige Kommunen derart von seinen Vorzügen überzeugen, dass sie bisher bestehende kommerzielle Systeme ablösen. Allein im Münsterland wird O.S.I.R.I.S. bereits in 16 Verwaltungen genutzt bzw. steht kurz vor dem Einsatz.³

Als zentrale Stelle für die Kontrolle und das Management der Weiterentwicklungsbestrebungen wird die Extension neben anderen auf Sourceforge.net unter dem Projekt TYPO3 Extension Development Platform gehostet.⁴ Anfragen aus Dänemark und Brasilien haben gezeigt, dass die Extension auch international auf nachhaltiges Interesse stößt und sich bereits Entwicklerteams zusammengefunden haben, die sich der Anpassungen auf lokale Gegebenheiten annehmen. Durch eine durchgängige Ausrichtung zur Mehrsprachigkeit kann O.S.I.R.I.S. leicht in andere Sprachen übersetzt werden.

² Vgl. <http://typo3.org/extensions/repository/search/civserv/>.

³ Stellvertretend sei hier die Internetpräsenz des Kreises Warendorf genannt: <http://www.kreis-warendorf.de/1571.0.html>.

⁴ Vgl. <http://sourceforge.net/projects/typo3xdev/>.

Literaturverzeichnis

- [Al04a] Algermissen, L. et al: Anforderungen an Virtuelle Rathäuser – Ein Leitfaden für die herstellerunabhängige Softwareauswahl. Arbeitsbericht Nr. 108 des Institutes für Wirtschaftsinformatik. Westfälische Wilhelms-Universität Münster, Münster, 2004.
- [Al04b] Algermissen, L. et al: Fachkonzeptionelle Spezifikation von Virtuellen Rathäusern - Ein Konzept zur Unterstützung der Implementierung. Arbeitsbericht Nr. 109 des Institutes für Wirtschaftsinformatik. Westfälische Wilhelms-Universität Münster, Münster, 2004.
- [Be04] Becker, J. et al: Virtuelles Rathaus Münsterland 2004 – Status Quo und Entwicklungsperspektiven. Westfälische Wilhelms-Universität Münster, Münster, 2004.

Mobiler Client für die E-Learning Plattform „CLIX“

Florian Fieber, Dipl.-Inform., M. Sc.

Fachhochschule Pforzheim
Master of Science in Information Systems / Wirtschaftsinformatik
florian@fieber.org

Art der Arbeit: Semesterarbeit im Fach „Advanced Software Engineering Project“
Betreuer der Arbeit: Prof. Dr. Karl-Heinz Rau, karl-heinz.rau@fh-pforzheim.de

Abstract: In der Semesterarbeit im Fach “Advanced Software Engineering Project” unter Leitung von Prof. Dr. Karl-Heinz Rau entwickelte ich eine Anwendung für mobile Endgeräte mit J2ME. Dabei erstellte ich für die in der Fachhochschule eingesetzte E-Learning Plattform “CLIX” einen mobilen Client. Der Hersteller der Plattform, IMC Saarbrücken, war bereit mich in diesem Projekt zu unterstützen und die notwendigen Schnittstellen zu schaffen und bereit zu stellen. Mit dem während des Projektes entwickelten Prototyp ist es beispielsweise möglich, News zu gebuchten Kursen abzurufen sowie neue News einzustellen. Der Prototyp wurde als Machbarkeitsstudie von IMC auf der Learntec 2004 in Karlsruhe vorgestellt.

1 Ausgangssituation

Die folgenden Schwerpunkte standen bei der Machbarkeitsstudie im Vordergrund:

- Evaluierung der Java 2 Micro Edition (J2ME), dabei speziell Mobile Information Device Profile (MIDP 2.0) und Connected Limited Device Configuration (CLDC)
- Benutzerführung auf mobilen Geräten, Testen des User-Interfaces
- Kommunikationsmöglichkeiten mit anderen Komponenten

Ziel des Projektes war die Entwicklung eines mobilen Clients für die E-Learning-Plattform „CLIX Campus“, welche an der Fachhochschule Pforzheim eingesetzt wird. Über den Client sollen verschiedene Funktionalitäten der Plattform nutzbar sein. Ein Benutzer (Student oder Tutor) kann sich über den Client an der Plattform authentifizieren und Aktionen ausführen. Diese sind zum Beispiel das Lesen von News zu unterschiedlichen Vorlesungen sowie das Erstellen von neuen News Beiträgen durch einen Tutor. Der mobile Client stellt somit eine Alternative zum webbasierten Frontend dar und bietet dem Benutzer einen mobilen Zugang zu seinen Kursen. Über ein Konfigurationsmenü kann der Benutzer seine persönlichen Einstellungen vornehmen.

Die benötigten Funktionalitäten der E-Learning-Plattform wurden als Add-On-Komponenten entwickelt und über WebServices bereit gestellt. Ein weiterer Server („Mobile Server“) greift auf diese Dienste zu, bereitet sie auf und stellt sie dem Client zur Verfügung. Der Client kommuniziert nur mit dem Mobile Server und nicht direkt mit dem CLIX Server.

Die Anwendung teilt sich somit in drei Komponenten auf. Die E-Learning Plattform CLIX läuft auf einem eigenen Server im Netz der Fachhochschule Pforzheim. Der „Mobile Server“ dient als Vermittler zwischen CLIX und dem mobilen Client. Er greift auf CLIX zu, bereitet die Daten auf und stellt sie dem Client zur Verfügung. Der mobile Client läuft auf mobilen Endgeräten und kommuniziert mit dem Mobile Server. Durch die Aufteilung in die drei Komponenten gewinnt man eine Abstraktionsschicht zwischen CLIX und mobilem Client. Hauptgrund für die Trennung von Mobile Server und CLIX Server ist demnach der Wunsch, CLIX im Intranet zu betreiben aber auch einen Zugriff des Mobile Clients aus dem Internet zu gewährleisten. Weiterhin können so WebServices von anderen Clients genutzt werden und eine Release-Entkopplung des Clients vom CLIX Server ist umsetzbar.

2 Implementierung

2.1 Use Cases

In der Applikation wurden die Anwendungsfälle „Authentifizieren“, „Ändern der Einstellungen“, „Lesen der News“ und „Erstellen der News“ umgesetzt. Diese stehen mit Ausnahme des letzten Anwendungsfalls für die Akteure „Student“ und „Tutor“ zur Verfügung. Ein Tutor kann im Gegensatz zu einem Studenten auch neue News erstellen.

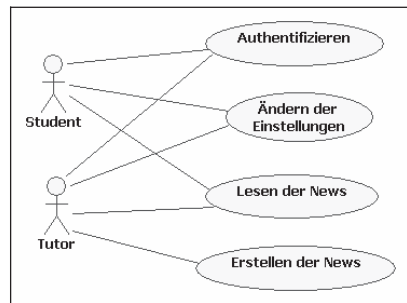


Abb. 1: Use Cases

2.2 Mobile Client

Der Client wurde entsprechend dem MVC-Muster aufgebaut, ein zentraler Controller nimmt alle Benutzeraktionen der User-Interfaces entgegen und ruft die Funktionalitäten von unterschiedlichen Models auf. Diese stehen als Handler für unterschiedliche Aktivitäten in einer verketteten Liste zur Verfügung. Kann ein Handler eine Anfrage nicht beantworten, leitet er sie weiter an den nächsten Handler. Ist ein Handler in der Lage, eine Anfrage zu bearbeiten, gibt er das Ergebnis zurück an den Controller. Dieser übergibt nach der Abarbeitung der Benutzeraktion die Kontrolle an das nächste User-Interface auf (z.B. Statusmeldung).



Abb. 2: Login



Abb. 3: Kursauswahl



Abb. 4: Anzeige der News

2.3 Mobile Server

Der Mobile Server dient als Schnittstelle zwischen dem Client und dem CLIX Server. Er nimmt die Anfragen des Client entgegen und bereitet sie für den CLIX Server auf. Außerdem ist er vor allem für das Session-Handling verantwortlich: bei der ersten Anfrage des Client wird eine Session mit dem Mobile Server aufgebaut, welche bei jeder weiteren Anfrage des Client genutzt wird. Dagegen ist die Beziehung zwischen dem Mobile Server und dem CLIX Server zustandslos.

3 Technologien

In dem Projekt sollte die Java 2 Mobile Edition (J2ME) an Hand einer Machbarkeitsstudie evaluiert werden. Die J2ME wurde speziell für mobile Geräte entwickelt und stellt im Gegensatz zu J2SE oder J2EE ganz andere Anforderungen an den Entwickler (beispielsweise stehen viel weniger Ressourcen zur Verfügung).

Die E-Learning Plattform "CLIX" des Herstellers IMC in Saarbrücken basiert auf einer J2EE Architektur. Für das vorliegende Projekt entwickelte IMC eine Webservice Schnittstelle zu CLIX über die der Mobile Server auf die Inhalte der Plattform zugreifen kann. Außerdem stellte IMC für das Projekt ein eigenes Logo bereit.

Die WebServices Schnittstelle wurde durch die SOAP Implementierung Axis von der Apache Software Foundation realisiert. Durch die Bereitstellung der WSDL-Beschreibung der WebServices konnte die Erzeugung der entsprechenden Clients (bzw. Stubs) automatisiert werden.

Auf dem Mobile Server läuft das Web Application Framework Struts, ebenfalls von der Apache Software Foundation .

Für die Entwicklung des Projektes wurde die Entwicklungsumgebung Eclipse eingesetzt. Für den Build Prozess wurde Ant verwendet, hierfür gibt es spezielle Tasks zum Verifizieren, Obfusieren und Packen von mobilen Anwendungen. Außerdem wurde für die Entwicklung das J2ME Wireless Toolkit von Sun (Version 2.0) verwendet. Dieses bietet unter anderem einen Emulator für Mobiltelefone.

4 Zusammenfassung

Man erhält mit J2ME und MIDP 2.0 eine mächtige Plattform zur Entwicklung von mobilen Anwendungen mit Java, vergleichbar zu J2SE und J2EE. Dennoch muss der Entwickler immer im Auge haben, dass die mobilen Geräte deutlich beschränkter sind in Bezug auf Prozessorleistung, Speicherplatz und Übertragungsgeschwindigkeiten. Es gilt, performant zu programmieren und nicht unbedacht eine Vielzahl von Objekten zu erzeugen, welche eigentlich nicht gebraucht werden. Ebenso stehen in der J2ME nicht so viele Hilfsklassen zur Verfügung wie in J2SE, diese gilt es durch eigene Implementierungen zu ersetzen. Weiterhin bedarf es auf jeden Fall eines effektiven Buildprozesses da die Ressourcen auf den Client und den Server verteilt werden müssen und beim Client nicht unbegrenzter Speicherplatz zur Verfügung steht.

Nicht nur aus Performancegründen an sich, sondern auch wegen der unterschiedlichen Implementierung von MIDP 2.0 der Hersteller, muss regelmäßig auf unterschiedlichen Zielgeräten getestet werden. Das Java-Paradigma „Write once, run everywhere“ kommt nur bedingt zum Tragen, da es sich zeigt, dass die Applikationen nicht bei jedem Hersteller problemlos laufen.

Wie in dem Projekt gezeigt, eignet sich J2ME durch seine GUI-Möglichkeiten sehr gut, um mobile Clients für (bestehende) Enterprise Anwendungen zu entwickeln.



Abb. 5: Ansicht des Splashscreen im Emulator

JDBC Datenbank Beobachter

Oliver Brüntje, Andre Glüpker, Frank Mertens, Karsten Wolke

FH Oldenburg/Ostfriesland/Wilhelmshaven
Fachbereich Technik , INK
Constantiaplatz 4
26723 Emden

De Montfort University,
Software Technology Research Laboratory
The Gateway
Leicester LE1 9BH, UK

oliver.bruentje@t-online.de,
andre_gl@web.de,
f_mertens@arcor.de,
mail@karsten-wolke.de

Art der Arbeit: Studienarbeit
Betreuer/Betreuerin der Arbeit: Prof. Dr. Karl Hayo Siemens

Abstract: Ein JDBC Datenbank Beobachter analysiert Informationsflüsse zwischen Client und Datenbank. Die Art der Information entscheidet über die Benachrichtigung interessierter Stellen. Diese Stellen melden bei Bedarf dynamisch ihr Interesse an. Mit Hilfe eines Datenbank Beobachters ist man in der Lage, mehrere Datenbestände on-the-fly aktuell zu halten, ohne regelmäßige, lang andauernde Datenabgleiche durchzuführen. Wegen des modularen Aufbaus des Datenbank Beobachters können weitere Komponenten implementiert werden, beispielsweise um datenbankspezifische Sprachelemente zu unterstützen und auszuwerten.

Einleitung

In der heutigen Zeit steigt die Bedeutung der Aktualität von Datenbeständen. Zur Lösung dieses Problems kann ein modularer JDBC Datenbank Beobachter (engl. JDBC-Database Observer, kurz JDBO) herangezogen werden. Der Datenfluß von und zu einer Datenbank wird überwacht, und ausgewählte Clients werden bei Auftreten bestimmter Ereignisse benachrichtigt. Diese Clients melden sich bei einem Server an, um ihr Interesse für ein bestimmtes Datenbank-Ereignis zu signalisieren. Weitere Komponenten sind dafür zuständig, die Anfragen an die Datenbank zu analysieren, zu kategorisieren und die Benachrichtigung der interessierten Clients zu veranlassen. Der JDBO basiert auf dem Listener Konzept [1, 2] und wendet es auf Datenbanksysteme an. Der JDBO bildet die Ereignisquelle.

Ein Beispiel für den Einsatz dieses Signalisierungssystems ist ein Produktionsstandort, an dem alle beteiligten Komponenten ihre Zustände in einer zentralen Datenbank speichern. Diese Änderungen können Benachrichtigungen interessierter Komponenten zur Folge haben. Weiterhin kann ein JDBO für die Datenbankreplikation zur Laufzeit eingesetzt werden. Durch einen JDBO werden Daten nur aktualisiert, wenn sich diese ändern. Zeitaufwendiges und Performance minderndes Polling entfällt.

Eigenschaften des Datenbank Beobachters

Anhand der oben genannten Beispiele ist zu erkennen, dass das Signalisieren einer Änderung an einen Interessenten weniger Aufwand bedeutet, als wenn dieser seine Informationen durch zyklisches Abfragen (polling) des zentralen Datenbestandes aktuell hält. Es ist geschickter, das Signalisieren einem zentralen System statt einem einzelnen Informationsverwerter zu überlassen. Überträgt man dies auf die Kommunikation von einzelnen Programmen mit einer Datenbank, so folgt, dass ohne das vorgestellte Konzept jeder Client die Datenbank zyklisch abfragen muss, um seine Daten auf dem aktuellen Stand zu halten (polling). Mit dem JDBO wird die Kommunikation im Netzwerk reduziert und die Auslastung des Clients und des Datenbank- Servers verringert. Insgesamt werden somit weniger Ressourcen verbraucht.

Der JDBO basiert auf der JDBC (Java Database Connectivity) Spezifikation [3]. Um den JDBO im Verbund mit einer Datenbank zu verwenden, wird ein JDBC-Treiber für das Datenbanksystem (DBS) vorausgesetzt. JDBC ist ähnlich aufgebaut wie ODBC (Open Database Connectivity) und bietet Java Anwendungen einen einheitlichen Zugriff auf Datenbanken. Die Kommunikation zwischen Anwendung und Datenbank geschieht via SQL. Alle JDBC Treiber unterstützen mindestens den Standard SQL92. Mittlerweile gibt es für nahezu alle relationalen Datenbanksysteme einen JDBC Treiber. Der JDBO kann somit zusammen mit nahezu jeder relationalen Datenbank eingesetzt werden.

Die Nutzung des JDBO ist für Clients transparent, da der JDBO selbst als JDBC Treiber fungiert. Er wird vor den herstellerspezifischen JDBC Treiber geschaltet (siehe Abbildung 1). Der JDBO kann somit in existierenden Systemen eingesetzt werden. Manipulationen am Quellcode bestehender Applikationen sind in der Regel nicht notwendig.

Der JDBO ist modular aufgebaut und bildet eine verteilte Anwendung im Netzwerk. Dadurch ist eine redundante und hoch verfügbare Bereitstellung der Komponenten möglich [4].

Einsatzgebiete des Datenbank Beobachters

Die Einsatzgebiete des JDBO sind vielfältig. Einer der größten Stärken des JDBO besteht darin, dass er im Zusammenhang mit nahezu allen relationalen Datenbanksystemen eingesetzt werden kann. Wie in den vorherigen Abschnitten beschrieben, kann er beispielsweise eingesetzt werden, wenn eine Java Komponente

über bestimmte Informationen in einer oder mehreren Datenbanken informiert werden möchte. Die Datenbanken werden nicht ständig abgefragt (polling), sondern nur, wenn der JDBO der Datenbank signalisiert, dass sich sein Datenbestand an einer bestimmten Stelle geändert hat.

Um aktuelle Datenbestände zu gewährleisten, werden häufig zeitintensive, unter Umständen nächtliche Datenabgleiche (Backups) nötig. Beim Ausfall der Datenbank kann nur der Zustand der letzten Sicherung wieder hergestellt werden. Zur Lösung dieses Problems, und um kurze Antwortzeiten der Datenbanken zu erhalten, können verteilte Datenbanken eingesetzt werden. Durch die Aktualisierung der Datenbestände zur Laufzeit (on-the-fly) mit dem JDBO sind alle Datenbestände stets konsistent. Dabei werden Verfahren wie das 2- bzw. 3- Phasen Commit Protokoll [5] eingesetzt. Der Ausfall einer Datenbank hat dann keine Auswirkungen, da andere Datenbanken die Aufgaben übernehmen können. Zudem können mit Hilfe des JDBO zur Laufzeit (*on the fly*) Datenbanken verschiedener Hersteller repliziert werden. Eine Replikation zwischen verschiedenen Datenbanksystemen wird somit auch unterstützt. Weiterhin können bei Bedarf auch nur bestimmte Informationen zwischen mehreren Datenbanken mit Hilfe des JDBO konsistent und synchron gehalten werden.

Ferner lässt sich ein JDBO auch zum Loggen von Datenbankzugriffen verwenden. Es können beispielsweise alle SQL Ausdrücke an eine Datenbank gesammelt werden. Die gesammelten Daten können beispielsweise zur Fehleranalyse von SQL Ausdrücken herangezogen werden.

Architektur des Datenbank Beobachters

Der JDBO besteht aus mehreren, untereinander kommunizierenden Komponenten (siehe Abbildung 4). Die zentralen Aufgaben des JDBO und deren Komponenten werden im Folgenden erläutert.

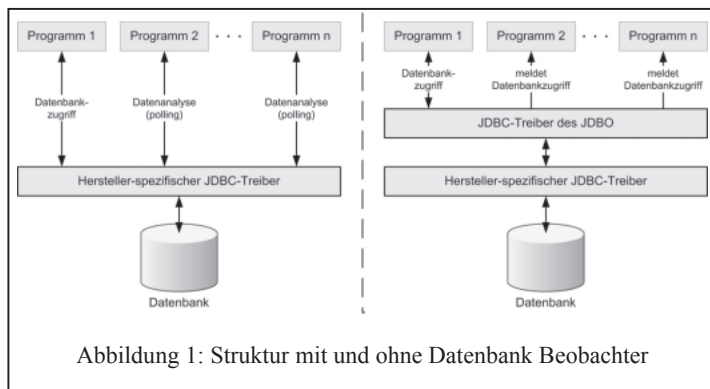


Abbildung 1: Struktur mit und ohne Datenbank Beobachter

Ein Teil des JDBO wird zwischen Client und dem DBS-spezifischen JDBC-Treiber der Datenbank eingeschoben. Damit registriert der JDBO den gesamten Datenstrom, der zwischen dem Client und der Datenbank ausgetauscht wird, und kann entsprechend darauf reagieren.

Der JDBO verhält sich für die Clients wie der herstellereigene JDBC-Treiber, ist jedoch selbst DBS-unabhängig. Er kann daher mit jeder relationalen Datenbank eingesetzt werden, ohne dass der Client Veränderungen

an der Datenbankschnittstelle bemerkt. Alle Interaktionen mit der Datenbank werden von dem JDBO 1:1 zum herstellerspezifischen JDBC-Treiber durchgereicht (siehe Abbildung 1).

Greift ein Client auf eine Datenbank zu, durchlaufen die Zugriffe in Form von SQL-Anweisungen den JDBC-Treiber des JDBO. Diese Aktionen werden ohne Änderung an den darunter liegenden herstellerspezifischen JDBC-Treiber und somit an die Datenbank weitergereicht. Zusätzlich werden die SQL-Anweisungen einem Analyser zur Auswertung übergeben.

Der Analyser hat die Aufgabe, die Art des Zugriffs sowie betroffene Tabellennamen, Zeilen, Spalten und Zellen zu ermitteln. Diese Informationen werden vom Analyser in einem Event-Objekt gespeichert und im nächsten Schritt einge-

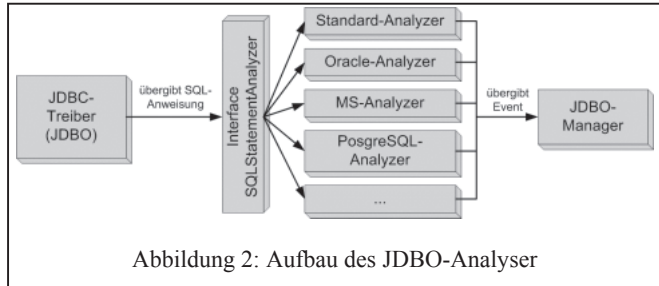


Abbildung 2: Aufbau des JDBO-Analyser

gesetzt, um gezielt Benachrichtigungen auszulösen. Der Analyser selbst ist ebenfalls modular aufgebaut (siehe Abbildung 2). Er stellt einen Standard-Analyser bereit, der SQL-Anweisungen analysiert, die dem Standard SQL92 entsprechen. DBS-spezifische SQL-Anweisungen werden mit angepassten DBS-Analysern ausgewertet.

Nachdem ein Event-Objekt vom Analyser angelegt wurde, wird es über die RMI-Schnittstelle an einen JDBO-Manager übertragen. Der Analyser ermittelt die Referenz des JDBO-Managers mit Hilfe eines Namensdienstes, in dem sich zuvor der JDBO-Manager mit seiner Adresse registriert hat. Ein JDBO-Manager ist für das Benachrichtigen der Clients zuständig. Clients können bei einem JDBO-Manager zu jeder Zeit dynamisch Interesse an Datenbankereignissen anmelden. Ein JDBO-Manager ist somit eine Art Verteiler zwischen dem Analyser und den angemeldeten, an Events interessierten Clients (siehe Abbildung 3). Durch diesen Aufbau und den Einsatz von RMI [6, 7] lassen sich mehrere JDBO-Manager einsetzen und diese entfernt auf separaten Maschinen betreiben.

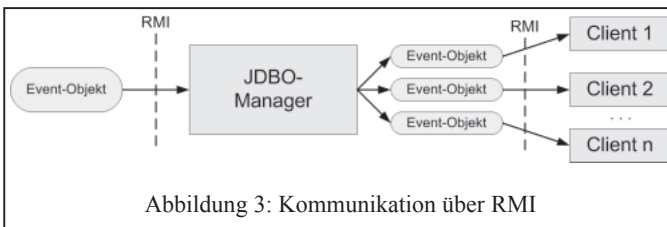


Abbildung 3: Kommunikation über RMI

Während der vorausgegangenen Anmeldung der Clients an einem JDBO-Manager teilen diese mit, über welche Art von Events (Tabellendaten geändert, Spalte X in Tabelle

Y geändert etc.) sie benachrichtigt werden möchten. Das Anmelden von Interesse geschieht über ein entferntes Listener-Objekt [1, 2]. So werden Clients, die sich auf anderen Maschinen befinden, über Ereignisse benachrichtigt. Diese Funktionalität wird

mit Hilfe von RMI [6, 7] erreicht. Zu jedem Event-Typ hält der Server eine Liste von zu benachrichtigenden Clients. Der Server verfügt somit über die Informationen für Benachrichtigungen.

Erhält der JDBO-Manager ein Event-Objekt, ruft er die zugehörige Liste bzw. Listen mit den interessierten Clients auf. Mit Hilfe der entfernten Listener-Objekte werden dann die passenden Clients benachrichtigt. So wird ein Event-Objekt an alle interessierten Clients verteilt. Die Clients sind dann über den Datenbankzugriff informiert und können gegebenenfalls darauf reagieren. Die Kommunikation insgesamt bzw. der Ablauf ist in Abbildung 4 dargestellt. Eine genauere Beschreibung des JDBO-Frameworks ist unter [8] zu finden.

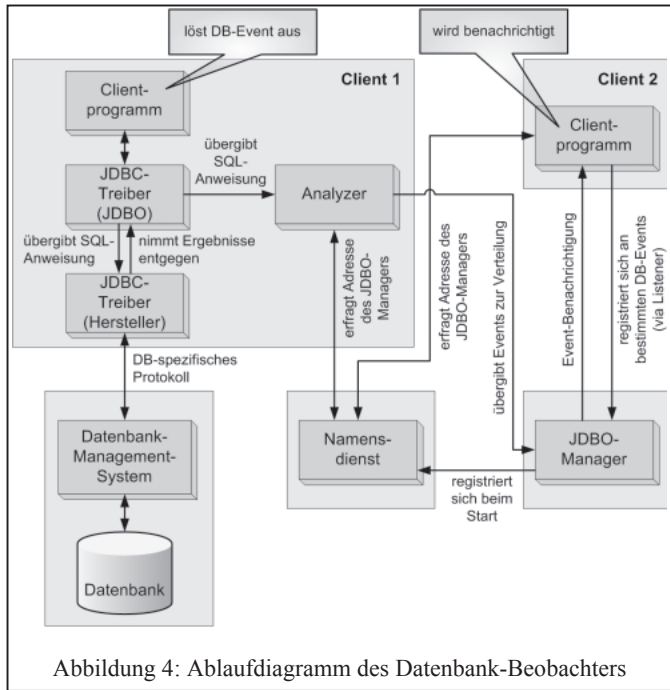


Abbildung 4: Ablaufdiagramm des Datenbank-Beobachters

Eine genauere Beschreibung des JDBO-Frameworks ist unter [8] zu finden.

Literaturverzeichnis

- [1] Gamma, E.; Helm, R.; Johnson, R.; Vlissides J.: Entwurfsmuster, Elemente wieder- verwendbarer objektorientierter Software, Addison Wesley, Reading, Massachusetts. 2001.
- [2] Wolke, K.; Janicke, H.; de Witt N.: Listener Concept and Java RMI. Tagungsband der Informatiktage 2002, Fachwissenschaftlicher Informatik-Kongress, S. 112-117, Konradin Verlag GmbH, Leinfelden, 2003.
- [3] Sun Microsystems: JDBC Technology. <http://java.sun.com/products/jdbc/>, 2004.
- [4] Wolke K.: RMI und Enterprise Java Beans, Hochverfügbarkeit und erweiterte Funktionalität. Diplomarbeit an der Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven, <http://www.karsten-wolke.de/Diplomarbeit/doc/Diplomarbeit.pdf>, 2003.
- [5] Garcia-Molina, H.; Ullman, J.: Database Systems, The Complete Book. Prentice Hall, 2003.
- [6] Sun Microsystems: Java™ Remote Method Invocation (RMI), Tutorial. <http://java.sun.com/j2se/1.4.2/docs/guide/rmi/>, 2003.
- [7] Sun Microsystems. RMI Architecture and Functional Specification. <http://java.sun.com/j2se/1.4.2/docs/guide/rmi/spec/rmiTOC.html>, 2003.
- [8] Brüntje, O.; Glüpker, A.; Mertens, F.; Wolke, K.: JDBC Database Observer, <http://www.karsten-wolke.de/public/doc/JDBO/JDBO.zip>

Visualisierung von Distanzfunktionen insbesondere der Earth Movers Distance

Till Schulte-Coerne

RWTH Aachen
Lehrstuhl für Informatik IX
schulte-coerne@informatik.rwth-aachen.de

Art der Arbeit: Semesterarbeit
Betreuerin der Arbeit: Ira Assent

Abstract: In dieser Arbeit wird die Earth Movers Distance als Distanzfunktion für das Content-Based-Image-Retrieval und ein neu entwickeltes Visualisierungstool zur Unterstützung der Forschungsarbeit an dieser Distanz erläutert.

1 Einführung

Im Rahmen der hier beschriebenen Semesterarbeit am Lehrstuhl für Informatik IX (Datenmanagement und Exploration, Prof. Dr. rer. nat. Thomas Seidl) an der RWTH Aachen wird ein Visualisierungstool für Distanzfunktionen entwickelt. Das Projekt ist eingebettet in den Forschungsbereich *Content-Based-Image-Retrieval*, also der Suche nach ähnlichen Bildern anhand ihres Inhalts in Multimedia-Datenbanken. Um Bilder vergleichen zu können, werden hierbei automatisch Merkmale (*Features*) extrahiert, die einer geeigneten Distanzfunktion unterworfen werden, um einen Wert für gegenseitige Unähnlichkeit zu gewinnen. Gesucht werden dafür Distanzfunktionen, die der menschlichen Wahrnehmung möglichst nahe kommen. Die Konzeptionierung und Validierung verschiedener Distanzfunktionen soll durch ein Visualisierungstool unterstützt werden. Dazu wird die Visualisierung aller Punkte, die eine bestimmte Distanz zu einem Mittelpunkt haben (das so genannte *ISO-Level*) gewählt. Um z.B. Eigenschaften einer unteren Schranke der *Earth Movers Distance* [RTG98] (kurz: EMD) für die mehrstufige Anfragebearbeitung zu illustrieren, ist es hilfreich, die ISO-Levels der EMD und einer möglichen Hypothese in einer 2- bzw. 3-dimensionalen Darstellung anzuzeigen. Im folgenden wird die 3-dimensionale Visualisierung vorgestellt, die 2-dimensionale ist einfacher aufgebaut und wird deshalb hier nicht betrachtet. Da die EMD bislang nur in prozeduraler Form vorlag, umfasst die Aufgabenstellung auch eine objektorientierte Implementierung in Java.

2 Distanzfunktionen für Content-Based-Image-Retrieval

Die *Minkovski-Distanz* als Maß für die Ähnlichkeit von Bildern bzw. den zugehörigen Farbhistogrammen berechnet die Distanz anhand der unterschiedlichen Farbhäufigkeiten:

$d_p(o, q) = \sqrt[p]{\sum_{i=1}^d |o_i - q_i|^p}$. Dabei wird etwa ein rotes Bild nicht als ähnlicher zu einem violetten als zu einem gelben bewertet. Dieses Problem lässt sich durch die Verwendung der *Quadratischen Form* lösen, die Histogrammeinteilungen in einer Matrix A bzgl. ihrer Ähnlichkeit gewichtet: $d_A(o, q) = \sqrt{(o - q)^t A (o - q)}$. Rubner und Tomasi [RT01] untersuchen verschiedene Distanzfunktionen im Bereich Computer Vision. Sie stellen fest, dass die Quadratische Form manchmal unterschiedlich strukturierte Bilder als zu ähnlich einstuft und schlagen als Lösung die EMD vor.

2.1 Earth Movers Distance

Die Berechnung der EMD basiert auf dem Transportproblem aus der Linearen Programmierung [HL90] und kann daher intuitiv als Transport von Erdmasse verstanden werden. Das Histogramm eines Bildes wird als eine Ansammlung von Erdhaufen betrachtet und das eines anderen Bildes als Ansammlung von Erdlöchern. Alle Haufen haben zu allen Löchern den Abstand, der aus den Abständen der Bins (also dem Farb-Abstand) zueinander resultiert. Gesucht ist nun die beste Aufteilung, sodass alle Erdlöcher komplett gefüllt, alle Erdhaufen komplett abgetragen und möglichst kurze Wege zurück gelegt werden müssen. Das bedeutet, man sucht nach der besten Aufteilung der Binwerte des einen Bildes zu den Binwerten des anderen Bildes bezüglich einer Kostenmatrix, die sich aus der Ähnlichkeit der Features ergibt. Die Kostenfunktion, die es unter den oben erwähnten Nebenbedingungen zu minimieren gilt, lautet: $\sum_{i=1}^n \sum_{j=1}^m (f_{ij} c_{ij})$. Die Matrix $F = (f_{ij})$ stellt die Aufteilung der Werte der Bins und $C = (c_{ij})$ die Kosten für die jeweiligen Transporte dar.

3 Konzept zur Visualisierung von Distanzfunktionen

Während man für 2D-Darstellungen pixelweise vorgehen kann, ist dies den meisten 3D-Engines auf Grund des nicht-diskreten Raumes schwer möglich. Da die ISO-Levels der hier betrachteten Distanzfunktionen geschlossene Oberflächen sind, bietet sich zur Lösung dieses Problems die Methode der *Marching-Cubes* [LC87] an. Hierbei verwenden wir ein 3-dimensionales Gitter, für dessen Gitterpunkte die jeweiligen Distanzwerte ausgerechnet werden. Dann wird anhand eines Vergleiches der Werte zweier nebeneinander liegender Gitterpunkte festgestellt, ob die Oberfläche zwischen den beiden Punkten liegt. Betrachtet man dann die Boxen, die sich aus jeweils 8 Gitterpunkten ergeben, so definieren die Schnittpunkte mit deren Kanten einen Teil der Oberfläche und alle Boxen somit die gesamte Oberflächenstruktur.

Will man nun die ISO-Levels verschiedener Distanzfunktionen darstellen, so nimmt man

als Eingabe die Koordinaten der Gitterpunkte und eines Mittelpunktes. Da wir hier nur auf den metrischen Fall der EMD eingehen wollen, muss gewährleistet sein, dass die Summe der Binwerte in beiden Histogrammen gleich ist. Das bedeutet, man benötigt ggf. zusätzlich zu den Koordinaten eine vierte Komponente, die die Summengleichheit sicherstellt. Man berechnet also 4-dimensionale Distanzen im Teilraum der summengleichen Punkte und projiziert diese dann in den 3-dimensionalen Raum. Allerdings ist es auch möglich, nur die Punkte im 3D zu betrachten, die summengleich sind. Daraus ergeben sich dann 2-dimensionale Objekte in einer Ebene.

Das Programm bietet nun die Möglichkeit, über eine Benutzerschnittstelle die Art der Visualisierung auszuwählen. Visualisiert man den 3D-Fall, so ergeben sich Bilder der Form wie in Abbildung 1a gezeigt. Der 4D-Fall führt zu Resultaten wie in Abbildung 1b. Des

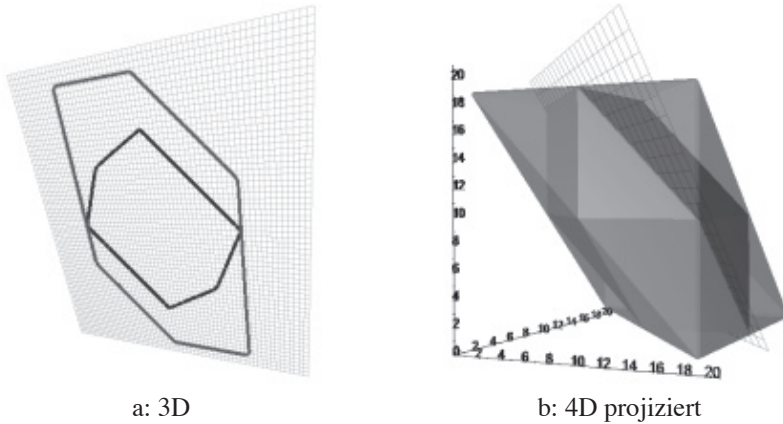


Abbildung 1: Gewichte Maximums-Norm (außen) als untere Schranke der EMD (innen)

weiteren bietet das Programm die Möglichkeit, die EMD zu parametrisieren und andere Distanzen (die *Minkowski-Distanzen* und die *Quadratische Form*) auszuwählen und ebenfalls zu parametrisieren.

4 Zusammenfassung

Mit Hilfe des hier vorgestellten Visualisierungstools ist es jetzt möglich, die EMD mit anderen Distanzen zu vergleichen. Das ist gerade deswegen von Nutzen, da hiermit die Suche nach unteren Schranken für die EMD erleichtert wird. Damit wird es möglich, die gute Annäherung der EMD an die menschliche Wahrnehmung mittels mehrstufiger Anfragebearbeitung effizient zu nutzen.

Literatur

- [HL90] Frederick Hillier und Gerald Lieberman. *Introduction to Linear Programming*. McGraw-Hill, 1990.
- [LC87] William E. Lorensen und Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, Seiten 163–169. ACM Press, 1987.
- [RT01] Yossi Rubner und Carlo Tomasi. *Perceptual Metrics for Image Database Navigation*. Kluwer Academic Publishers, 2001.
- [RTG98] Yossi Rubner, Carlo Tomasi und Leonidas J. Guibas. The Earth Mover's Distance, Multi-Dimensional Scaling, and Color Based Image Retrieval. In *Proceedings of the ARPA Image Understanding Workshop*, Seiten 661–668, 1998.

Predicting Customer Behavior using Naive Bayes and Maximum Entropy – Winning the Data-Mining-Cup 2004 –

Arne Mauser, Ilja Bezrukov, Thomas Deselaers, Daniel Keysers
Lehrstuhl für Informatik VI, Computer Science Department
RWTH Aachen University, D-52056 Aachen, Germany
{mauser, bezrukov, deselaers, keysers}@i6.informatik.rwth-aachen.de

Abstract: In this work we describe combinations of classifiers using Naive Bayes, Maximum Entropy, Neural Networks and Logistic Regression for classification of customer records. Performance of these approaches is confirmed by the 1st, 3rd, and 5th rank in the Data-Mining-Cup 2004.

1 Introduction

With the increasing possibility of collecting data in business applications there is a rising demand to utilize the available information. For financial institutions it may be the detection of fraudulent transactions or prediction of a company's liquidity situation. In sales it might be the prediction of any aspect of customer behavior.

A major expense factor in mail-order business is the cost of returns. In 2004, prediction of returning behavior of customers of a major German mail-order company was the task of the Data-Mining-Cup competition. The Data-Mining-Cup is an annual student competition organized by the Chemnitz University of Technology and Prudsys AG.

In this paper, we are going to present the approaches, that obtained the 1st, 3rd, and 5th rank out of 97 participants. A Maximum Entropy combination of Naive Bayes classifiers won the Data-Mining-Cup in 2004. Combining Logistic Regression, Neural Networks, and Maximum Entropy obtained the 3rd rank.

2 Task description & approach

The data, provided by a German mail-order company was split into a test set and a training set. Each consisted of 20147 customer records with 65 attributes. The attributes contained information about the customer's ordering behavior (e.g. "value of ordered goods in period C"), as well as geographical and statistical information (e.g. "fraction of households with children in the customer's ZIP-Code area").

Customer records were separated into 3 different classes, each class representing a different returning behavior. Customers who returned less than 18% of ordered goods were classified as “low returners” (L), those customers returning more than 40% were classified as “high returners” (H), those in between were labelled as “indefinite” (I).

For the test data the class labels were unknown and had to be predicted. A cost matrix for weighting classification errors between classes was used within the evaluation. Correct predictions of high and low returners were assessed with +1 points, wrong predictions with -1 points. With indefinite returners, correct classifications were rewarded with +0.5 points while errors were neglected. Each classification was assessed according to this measure and the result was summed over all records resulting in a global score.

3 Data preprocessing

Real-world data usually suffers from deficiencies that make classification tasks difficult: missing values, outliers, and noisy distributions affect the performance of classification algorithms. Furthermore, for many classifiers it is necessary to adjust feature values to a common interval. Generalization and noise reduction through histograms may also help to improve classification.

In our approach, the majority of features were transformed into “equi-depth” histograms. Bin borders were adjusted so that each bin contained approximately the same number of elements. Each individual feature value was then replaced by the center of the bin it belonged to. Using 5-fold crossvalidation, we determined that a 10-bin histogram performed best on the training data.

For features that contained a percentage of returned goods, we additionally generated two binary features for zero and missing values. The remaining features, like the ZIP-Code or the title of the customer were replaced with binary features. The decisions on what transformation to apply to the individual features were taken manually, based on the feature description.

4 Data classification

In order to estimate the performance of classification algorithms on unseen data and to find the most suitable approach, we evaluated several methods on the training data using 5-fold crossvalidation.

The four classification algorithms we tested performed similarly well: Logistic Regression [HK00], Neural Networks [Bis96], Naive Bayes [HK00] and Maximum Entropy [BPP96]. In the following paragraphs, we will give a brief description of the latter two methods.

Naive Bayes classifiers rely on Bayes’ decision rule with the assumption that the probability of each attribute value is independent from the values of other attributes. This assumption is not valid in general, but provides the advantage of computational simplic-

Table 1: Results of classification methods obtained using 5-fold crossvalidation on training data.

Method	Error rate	Score
Logistic Regression	27.2	10539.0
Neural Network	27.3	10496.5
Maximum Entropy	27.0	10610.0
Naive Bayes	39.0	5774.0
Logistic Regression + Neural Network + Maximum Entropy	26.9	10664.0
Naive Bayes + Maximum Entropy	27.0	10613.0

ity and often leads to very good results. Inspired by this approach we used the posterior distributions of the class given each feature and combined these using the sum rule.

Maximum Entropy classifiers model the marginal distributions of the training data while trying to be as general as possible. Among the set of possible models the one with the highest entropy is chosen in training. We used the Maximum Entropy approach to estimate weights for the posterior distributions of the different features in the combination.

Combining different classifiers often improves the results over using a single classifier as disadvantages of one method might be compensated by others.

We used two methods for classifier combination. In our first approach we combined separately trained classifiers using the sum rule. That is, we summed up the a posteriori probabilities of the classifiers and chose the class having the maximum sum. Using this method, we combined Logistic Regression, Neural Network and Maximum Entropy classifiers.

Our second approach to classifier combination used a Naive Bayes classifier. No assumption was made regarding the distribution of the values. Probabilities were estimated by relative frequencies. Unseen values in test were replaced by their nearest neighbors. Instead of taking the product of the models estimated for the individual features, we used a Maximum Entropy approach for weighting the individual distributions.

For convenience, we reduced the number of classes to two, neglecting the class of “indefinite” returners, as this class was strongly underrepresented. Sharing the probability mass of the neglected class equally between the remaining two other classes, we used the probabilities of the Naive Bayes distributions for class “L” as feature functions. For training the feature function weights we used the Generalized Iterative Scaling algorithm as described in [KON02].

5 Results

In Table 1, we summarize the results for the described classification methods using 5-fold crossvalidation on training data. For the competition, parameter estimation was performed using all available training data. Table 2 shows the final scores of the top 5 submissions on the unseen test data as determined by the competition organizers.

Table 2: Top 5 submissions in the competition.

	University	Methods used	Score
1.	RWTH-Aachen	Naive Bayes + Maximum Entropy	10511.0
2.	Warsaw University	Support Vector Machine	10491.0
3.	RWTH-Aachen	Log. Reg. + Neural N. + Max.Ent	10490.0
4.	Handelshochschule Leipzig	Unknown	10459.0
5.	RWTH-Aachen	Maximum Entropy	10455.0

The combination of Naive Bayes and Maximum Entropy scored best on the test set, although it did not perform as well on the training set. This can be explained by a slight over-fitting to the training data of our first combination approach.

6 Conclusion

Using histogramization as data preprocessing steps and various classifier combinations we obtained excellent results in the Data-Mining-Cup 2004.

It can be seen that our data transformation was a good choice for the given task, as different classifiers perform equally well on this data. A Maximum Entropy combination of a set of Naive Bayes classifiers performed best on the unseen test data. A combination of Logistic Regression, Neural Networks, and Maximum Entropy using the sum rule performed almost as well. This shows that, given a set of well transformed data, the selection of the individual classifier may be less important.

Furthermore we observe that it is essential to handle the issue of over-fitting with care.

References

- [Bis96] C. M. Bishop. *Neural networks for pattern recognition*. Clarendon Press, 1996.
- [BPP96] A. L. Berger, S. Della Pietra, and V. J. Della Pietra. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71, 1996.
- [HK00] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 1st edition, 2000.
- [KON02] D. Keysers, F. J. Och, and H. Ney. Maximum Entropy and Gaussian Models for Image Object Recognition. In *Pattern Recognition, 24th DAGM Symposium, Zürich, Switzerland*, volume LNCS 2449 of *Lecture Notes in Computer Science*, pages 498–506. Springer-Verlag, 2002.

Interaktive Muskelmodellierung für virtuelle Umgebungen

Sebastian Ullrich
sebastian.ullrich@rwth-aachen.de
Lehrstuhl für Hochleistungsrechnen
Rheinisch-Westfälische Technische Hochschule Aachen

Abstract: Diese Arbeit befasst sich mit immersiver Modellierung. Hintergrund ist die Darstellung von virtuellen Menschmodellen mit realistischer Hautdeformation durch eine mehrschichtige anatomische Muskulaturesimulation. Dazu ist ein passendes Modellierungswerkzeug zur Erzeugung und Modifikation von Daten für die Simulation gesucht worden. Das aus dieser Anforderung entstandene *MuscleAssemblyLab* wird hier vorgestellt und durch ein vergleichendes Experiment evaluiert.

1 Einleitung

Zur Darstellung von künstlichen Menschen in virtueller Umgebung gibt es eine Fülle von Visualisierungstechniken. Aus der künstlerischen Anatomie ist die Idee hervorgegangen Kenntnisse über nicht sichtbare Schichten – hauptsächlich Muskulatur – bei der Darstellung und Animation von Haut zu berücksichtigen. Bisher wurde dies jedoch hauptsächlich innerhalb der Filmindustrie bei rechenintensiven Spezialeffekten eingesetzt. Ansätze zur allgemeinen echtzeitfähigen Verwendung in der virtuellen Realität finden sich beispielsweise in [AT01] und [NTHF02].

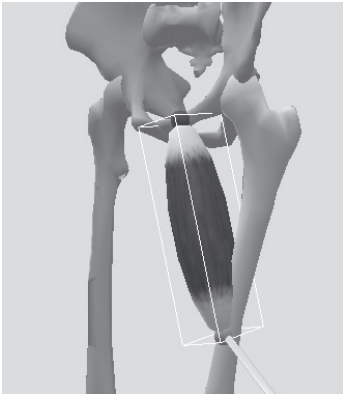
Um eine stärkere Nutzung zu etablieren sind Mittel zur einfachen Anwendung der Algorithmen wichtig. Zuerst wurden vorhandene Werkzeuge in Betracht gezogen, die eine Modellierung und Simulation von Muskulatur unterstützen: *Absolute Character Tools* (Plugin für 3D-Studio MAX), *CATMuscle* (Plugin für 3D-Studio MAX) und *cgmuscle* (Plugin für Maya). Die Verwendung der Muskulaturdaten in eigenen Algorithmen erweist sich als problematisch, weil die Daten in einem proprietären und auf die jeweils eigene Simulation abgestimmten Format abgespeichert werden. Außerdem ist die Einschränkung der Programme auf 2D-Desktopsysteme eine zusätzliche Barriere.

Als fundamentale Verbesserung werden immersive Modellierungswerkzeuge angesehen, da entsprechende Ein- und Ausgabegeräte (Eingabegeräte mit sechs Freiheitsgraden, stereoskopische Ausgabe) direkte Interaktionen in dem 3D-Modellierungsraum ermöglichen. Hierzu wurde die Funktionalität und Umsetzung der folgenden Forschungsprojekte zum Teil als Vorbild genommen: *3dm* [BDHO92], *HoloSketch* [Dee95] und *LEMAN* [TG98].

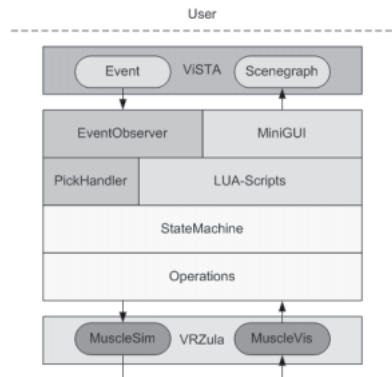
Ziel dieser Arbeit ist durch ein neues Werkzeug Muskelmodellierung mit immersiver Modellierung zu kombinieren und durch einen Benutzertest Vorteile gegenüber Desktopsystemen zu evaluieren.

2 MuscleAssemblyLab

Die Hauptaufgabe von *MuscleAssemblyLab* ist das Erstellen und Modifizieren von Parametern für eine Muskelsimulation. Um dies sowohl an Desktopsystemen, als auch in immersiven CAVE-artigen Umgebungen zu ermöglichen, wird diese Funktionalität durch direkte Manipulation realisiert. Das heißt das direkt an Objekten, in diesem Fall an der grafischen Darstellung von Muskeln mittels geeigneter Eingabemetaphern, Änderungen durchgeführt werden können (vgl. mit Abbildung 1(a)).



(a) direkte Manipulation: Verschiebung des grün markierten Muskelendes mit einem Zeigestrah



(b) Architektur

Abbildung 1: Funktionsweise und Architektur von MuscleAssemblyLab

2.1 Architektur

Das Modellierungswerkzeug ist an ein VR-Toolkit (ViSTA¹) und die Simulation- und Visualisierung (VRZula²) (vgl. Abbildung 1(b)) angebunden. Der Entwurf zur Architektur der Benutzerschnittstelle basiert auf einer von Tanriverdi et al. vorgeschlagenen Methode zur Entwicklung von VR-Schnittstellen [TJ01]. Dieser Ansatz wurde gewählt, weil zwischen einer Interaktions- und einer Verhaltenskomponente unterschieden wird, was zur Erhöhung der Flexibilität bei der Anpassung an verschiedene Plattformen beiträgt.

In der Interaktionskomponente befindet sich der *EventObserver*, der die Daten der Eingabegeräte vorverarbeitet. Danach können die Daten von dem *PickHandler* oder von *LUA-Skripten* weiterverarbeitet werden und *Operationen* (z.B. Muskel erstellen, löschen, verschieben, ...) auslösen. Bevor jedoch Operationen gestartet werden, erfolgt in der *State-Machine* eine Prüfung und Änderung von Systemzuständen.

¹http://www.rz.rwth-aachen.de/vr/research/vista/vista_e.php

²<http://www.vrzula.org>

Die Operationen und die StateMachine gehören zur Verhaltenskomponente. Weiterhin stellen die Operationen die Schnittstelle zur Simulation und Visualisierung der Muskulatur dar. Außerdem gibt es eine graphische Komponente, durch die der Benutzer Feedback über seine Interaktionen erhält. Dazu gehört hauptsächlich die *MiniGUI*, durch die ein Menüsystem dargestellt wird, aber auch kleinere Funktionen, wie z.B. das farbliche Markieren von Objekten.

3 Experiment

Zur Evaluierung der Vorteile einer immersiven Modellierungsumgebung ist das nachfolgend beschriebene Experiment durchgeführt worden. Dazu wurden zwei Hypothesen aufgestellt und untersucht: „*In einer immersiven Umgebung wird aufgrund natürlicher Interaktionsformen schneller modelliert.*“ und „*In einer immersiven Umgebung wird die Modellierung präziser durch die zusätzlichen Freiheitsgrade der Eingabe und Ausgabe.*“

Für den Versuch wurden 23 freiwillige Probanden jeweils zuerst an einem Desktopsystem und zwei Wochen später in einer CAVE-artigen Umgebung getestet. Dabei bestand die Aufgabe in der Modellierung von Muskulatur an einem rechten Oberarm unter Zuhilfenahme anatomischer Abbildungen. Zum Vergleich wurde die Zeit zur Lösung der Aufgabe gemessen und durch ein euklidisches Abstandsmass Abweichungen von einem Referenzmodell ermittelt.

Durch multivariate Analysen ist eine Signifikanz der Zeit bezüglich der beiden Plattformen nachgewiesen worden, die sich auch im Vergleich der Mittelwerte der Normalverteilungen von 14,3 min am Desktop und 10,43 min in der CAVE durch eine deutliche Verbesserung widerspiegelt. Dagegen konnte bezüglich der Genauigkeit, bei einem Vergleich zwischen den Systemen keine Signifikanz nachgewiesen werden. So beträgt der mittlere Fehler pro Muskelende am Desktop circa 3,4 cm und in der CAVE circa 3,0 cm, d.h. es hat nur sehr geringfügige Verbesserungen gegeben.

Mit diesen Ergebnissen erfolgt die Schlussfolgerung, dass sich der Einsatz von immersiver Modellierung zur Minimierung des Zeitaufwands lohnt. Eine erhoffte Verbesserung der Präzision konnte nicht belegt werden.

4 Zusammenfassung und Ausblick

Mit dem beschriebenen MuscleAssemblyLab ist ein neues Werkzeug zur intuitiven Modellierung von Muskulatur geschaffen worden. Durch die gewählte Architektur und damit verbundene Aufteilung in Interaktions- und Verhaltenskomponente ist es sowohl an Desktopsystemen, als auch in immersiven Umgebungen einsetzbar (siehe Abbildung 2).

Ein geplanter Ansatzpunkt für Verbesserungen und neue Experimente ist die Modellierungspräzision. Während des Experiments wurde beobachtet, dass die meisten Probanden in der CAVE auf größere Distanz zum Objekt (vgl. untere Hälfte von Abbildung 2(b))

modellieren. Daher liegt die Vermutung nahe, dass eine Forcierung eines geringeren Modellierungsabstands die Genauigkeit bei der Modellierung erhöht.

Weiterhin kann wahrscheinlich durch spezialisierte Eingabegeräte auch an Desktopsystemen die Modellierung vereinfacht bzw. verbessert werden.



(a) monoskopischer Desktop

(b) immersive CAVE-Umgebung

Abbildung 2: Beispiel für die Verwendung von MuscleAssemblyLab mit verschiedenen Ein- und Ausgabesystemen

Literatur

- [AT01] Amaury Aubel und Daniel Thalmann. Interactive Modeling of Human Musculature. In *Computer Animation*. Seoul, Korea, 2001.
- [BDHO92] Jeff Butterworth, Andrew Davidson, Stephen Hench und Marc. T. Olano. 3DM: A Three Dimensional Modeler Using a Head-Mounted Display. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, Seiten 135–138. Cambridge, Massachusetts, United States, ACM Press, 1992.
- [Dee95] Michael F. Deering. HoloSketch: A Virtual Reality Sketching/Animation Tool. *ACM Transactions on Computer-Human Interaction*, 2(3):220–238, 1995.
- [NTHF02] Victor Ng-Thow-Hing und Eugene Fiume. Application-Specific Muscle Representations. In *Proceedings of Graphics Interface*, Seiten 107–116, 2002.
- [TG98] Russell Turner und Enrico Gobbetti. Interactive Construction and Animation of Layered Elastically Deformable Characters. *Computer Graphics Forum*, 17(2):135–152, 1998.
- [TJ01] Vildan Tanriverdi und Robert J.K. Jacob. VRID: A Design Model and Methodology for Developing Virtual Reality Interfaces. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, Seiten 175–182. ACM Press, 2001.

An Open and Extensible Framework for Visualization

Christoph Bastuck, Timo Hambuerger, Thorsten Hof, Maik Keller,
Peter Kohlmann, John Mehnert, Stefanie Nowak

christoph@jeremys-fate.com, timo.hambuerger@gmx.de, thof@cityweb.de,
kellermaik@gmx.de, Peter.Kohlmann@gmx.de,
john_mehnert@web.de, Stefanie.Nowak@t-online.de

Supervisor: Dr. Chistof Rezk-Salama, Prof. Dr. Andreas Kolb

University: University of Siegen

Departement: Computer Graphics and Multimedia Systems

Area of operation: Practical term, Student research project

Abstract: The paper presents a uniform framework for scientific visualization, that comprises a variety of different filtering, mapping and rendering techniques. We offer architectural concepts that implement a novel approach to extensibility on different levels. We have developed an open and expandable model for intuitive user interaction, which can be adapted to meet the demands of different scientific communities.

1 Motivation

The visualization of scientific data is of great importance in many scientific disciplines, such as medicine, natural and computational science and engineering. In recent years a huge variety of sophisticated visualization techniques have been developed by numerous research groups all over the world. Simultaneously, computer graphics applications have moved from expensive supercomputers to low-cost desktop computers, due to an increase in computational power of graphics processor which outperforms Moore's law.

To account for the high complexity inherent in scientific data fields, visualization has become a creative and investigative task, which requires a high degree of user interaction. In an interactive process, the user analyzes the data field, reveals and examines different aspects and the interrelationships which are hidden within the large amount of data.

2 Architectural considerations

Our aim was to create a uniform visualization framework which does not restricted itself to a designated application area or a specific data origin. After the general framework is implemented, specific applications can be build on top of it. In consequence, we focus on three main aspects, the visualization framework must fulfill:

1. Open architectural design with easily expandable functional range
2. Extensibility to arbitrary multidimensional and multivariate data
3. Intuitive usability for non-expert users

The task of creating images from abstract data fields can be described using several levels of abstraction. On the highest level, the process of image generation is considered a three-tier approach, frequently quoted as the *visualization pipeline*.

The first tier is a *preprocessing* step, where filters are applied that manipulate the data. In the second tier, the *mapping* step, the output of the preceding step is mapped to geometric shapes and visual attributes. In the third tier, the *rendering* step, images are created as the result of the visualization.

The first step contains several pipeline elements which can be arranged in a *chain*. The user must have full control over the configuration chain at runtime. Simultaneously, the system must assure consistency among its elements. Additionally, the architecture allows the user to implement new elements such as filters and converters. The same idea was applied in the second tier. The user can write his own mapping procedures, which extends the functional range of the framework. In order to meet all the requirements of current and future applications, the framework has to support multidimensional datasets (e.g. 2D and 3D) with store multiple data fields (e.g. scalar, vector or multivariate data) that are discretized on a common grid (e.g. uniform, curvilinear or unstructured).

As a result of the open architecture, arbitrary visualization tasks can be achieved with respect to usability. The configuration of the visualization pipeline is decisive for the results of the visualization process. Hence, its graphical representation is the interface for user interaction, element insertion and deletion as well as configuration of individual elements. Its handling should be intuitive and mostly independent of the user's previous knowledge. Therefore, it is important to provide sufficient abstraction from the pure technical view of the pipeline [BKM91]. In addition to the flexible pipeline configuration, a multi-view concept allows the user to choose at runtime from a set of possible views onto the pipeline. This set can be expanded by implementing custom views. As a result, we can adapt the usability to different approaches of user interaction.

3 Implementation

We used the concepts of object oriented programming to achieve openness. Abstract base classes and polymorphism are used to define basic interfaces, that allow the framework to be extended at any time. An abstract class is built for all pipeline elements that includes basic pipeline functionality, a runtime type recognition system and functionality concerning data flow and data manipulation. Especially efficient operations for execution of linear filters and an effective mechanism for allocation and deallocation of system memory, during traversal of the pipeline are implemented. An abstract base class for data conversion extends the set of pipeline elements. Converters change the type of data by executing a characteristic function. As an example, we implemented 3D line integral convolution [RSHT99] as a converter that generates a 3D scalar field out of a

3D vector field. Currently implemented mapping procedures comprise particle traces, 2D and 3D line integral convolution [CL93], stream surfaces and ribbons for vector field visualization [PVH⁺02] as well as texture based volume rendering [RS02] and *Marching Cubes* [LC87] for 3D scalar fields, only to mention a few.

We decided to implement manager classes, which assure compatibility among pipeline elements with respect to data type and the output of the preceding unit. Factory classes are used to achieve an open set of pipeline elements and mapping procedures, for an extensible framework that supports new pipeline elements and mapping procedures. To manage the variety of different data, we developed separate abstract classes for the grid and the variables, which provide base functionality and define a programming interface. Factory classes are used for instantiation. Since *multivariate* datasets can have multiple values at each grid point, we decided to create an (optionally compressed) archive that contains a raw data file for each variable. The grid structure is stored in a separate file. For type information, we added a meta file that holds information of the data using XML. The file format can be extended for future needs.

Since visualization requires creative and investigative skills, we designed the *TreeView*,

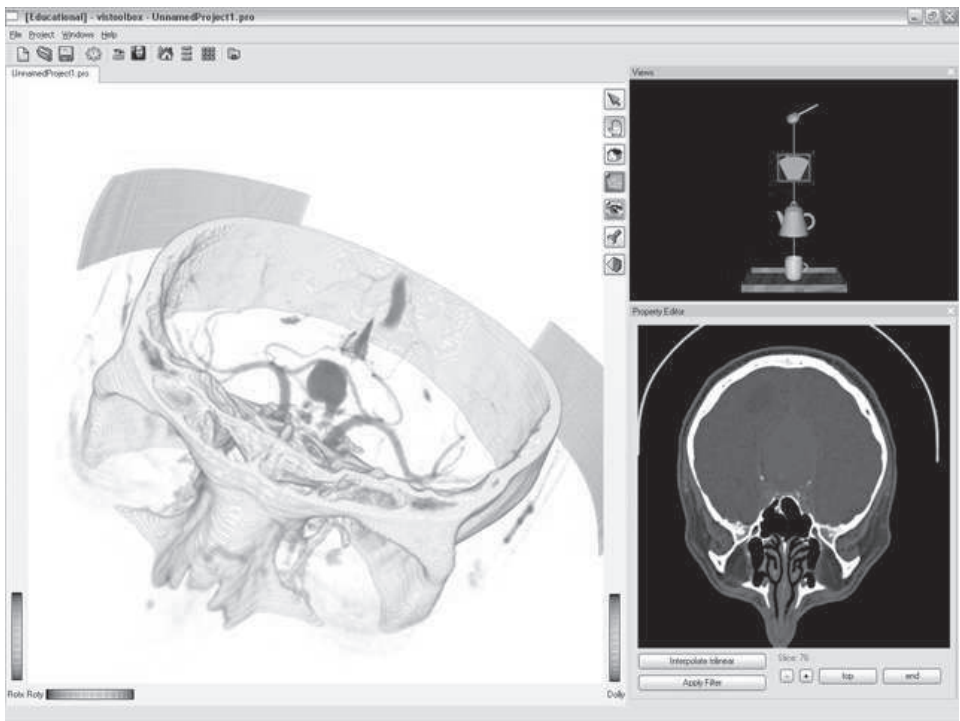


Figure 1: Example of the visualization framework: Main window with CT angiography of the human brain visualized by direct volume rendering, tree view (*upper right*) and the selection filter for volumetric data (*lower right*)

which represents an abstraction of the visualization pipeline. Icons, analogous to entities in everyday life, describe its elements and enable the user to configure the pipeline. Analogies are used to create a common understanding of the functionality by choosing entities we all can refer to. This approach offers a playful access and contributes to an intuitive handling. As different graphical representations of the visualization pipeline can be intuitive for different people, we designed a separate interface for user interaction to configure the pipeline. Through inheritance, new views can be implemented and easily integrated in the framework. At runtime, the user can select the view that meets his requirements.

4 Conclusion and Future Work

We presented a visualization framework that focuses on an open and extensible architecture, which we implemented in a one year project at the University of Siegen. The result of our work is available as *vistoolbox* under GPL open source license. By providing access to everyone we emphasize the extensible character of the framework. The open architecture and detailed documentation (doxygen) empowers users to contribute new functionality to the framework. The software is based on Trolltech Qt and SIM Coin 2.0 and runs on different platforms (Linux, WindowsXP, MacOS).

For future work we are planning to add new mapping procedures and other pipeline elements, especially converters that help to find coherences and thus create understanding of natural phenomena.

References

- [BKM91] N. Bevana, J. Kirakowskib, and J. Maissela. What is Usability? In *Proceedings of the 4th International Conference on HCI*, Stuttgart, 1991.
- [CL93] B. Cabral and L. Leedom. Imaging Vector Fields Using Line Integral Convolution. In *Proc. SIGGRAPH*, 1993.
- [LC87] W. Lorensen and H. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Comp. Graphics*, 21(4):163–169, 1987.
- [PVH⁺02] F. Post, B. Vrolijk, H. Hauser, R. Laramée, and H. Doleisch. Feature Extraction and Visualization of Flow Fields. In *State-of-the-Art Proceedings of EUROGRAPHICS*, pages 69–100, Saarbrücken, Germany, September 2002.
- [RS02] C. Rezk-Salama. *Volume Rendering Techniques for General Purpose Graphics Hardware*. Dissertation, University of Erlangen-Nuremberg, Germany, September 2002. Arbeitsberichte des Instituts für Informatik, Band 35, Nummer 5.
- [RSHT99] C. Rezk-Salama, P. Hastreiter, C. Teitzel, and T. Ertl. Interactive Exploration of Volume Line Integral Convolution Based on 3D–Texture Mapping. In *Proc. IEEE Visualization*, 1999.

Erweiterung eines holistischen statistischen Bilderkenners zur Verwendung von mehreren Merkmalen

David Rybach, Daniel Keysers, Hermann Ney
david.rybach@rwth-aachen.de
{keyzers,ney}@cs.rwth-aachen.de

Lehrstuhl für Informatik VI, Computer Science Department
RWTH Aachen University, D-52056 Aachen, Germany

Art der Arbeit: Studienarbeit
Betreuer der Arbeit: Dipl.-Inform. D. Keysers, Prof. Dr.-Ing H. Ney

Abstract: Statistische Bilderkenner, die mit einem holistischen Ansatz arbeiten, verwenden Wahrscheinlichkeitsverteilungen als Modelle für Objekte und deren Hintergründe. Die Modelle werden aus ortsabhängigen Merkmalen der Trainingsbilder berechnet, z.B. aus den Grauwerten. In dieser Arbeit wird ein Verfahren vorgestellt, das mehrere Merkmale in die statistischen Modelle integriert. Dadurch konnten die Fehleraten auf zwei verschiedenen Datensätzen verringert werden.

1 Einleitung

Der erste Schritt in den meisten Verfahren zur Bilderkennung ist die Berechnung von Merkmalen aus den Bilddaten. Dies können ganz einfache Merkmale sein, zum Beispiel die Grau- oder Farbwerte des Bildes, oder Daten, die aus Berechnungen auf dem Bildmaterial hervorgehen. Die Merkmalsdaten werden analysiert und vom Klassifikator benutzt.

Bilderkenner, die mit einem holistischen statistischen Verfahren arbeiten, können komplexe Szenen analysieren und die abgebildeten Objekte klassifizieren. Sie „erklären“ das gesamte Bild mit statistischen Modellen. Es ist also notwendig, Modelle für Objekte und Hintergründe zu verwenden. Die Bestimmung der Position und der Größe des Objekts und dessen Klassifizierung geschehen in einem Schritt. Viele andere Verfahren müssen vor der Klassifikation eine Segmentierung des Bildes durchführen, die alleine schon fehleranfällig ist. Auch für das automatische Objekttraining werden unsegmentierte Daten benutzt, für die nur die Klasse des abgebildeten Objekts bekannt ist. Daher müssen die Methoden, die zur Klassifikation genutzt werden, auch für das Training angewendet werden. Die Verwendung von unsegmentierten Daten ist wünschenswert, um die manuelle Arbeit an den Daten zu minimieren.

In dieser Arbeit wird die Erweiterung eines holistischen Bilderkenners vorgestellt, die es ermöglicht, mehrere Merkmale eines Bildes zu verarbeiten. Um die Merkmale eines Bildes zu kombinieren, wird ein neues „Bild“ berechnet, das aus mehreren Schichten besteht. Die einzelnen Schichten repräsentieren Merkmale oder Teile eines Merkmals.

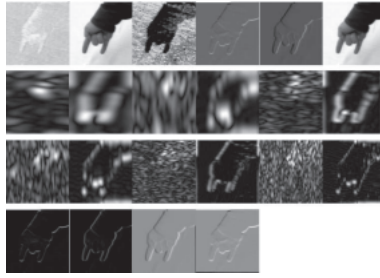


Abbildung 1: Features aus einem Bild: Farbe mit Komponenten S, V, H (3), Ableitung (2), Grauwert (1), Gabor-Transformation mit 3 Frequenzen und 2 Orientierungen jeweils für Farbe und Helligkeit getrennt (12), absolutwertige Ableitung (2), Sobel-Filter (2)

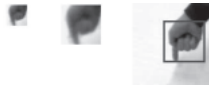


Abbildung 2: Matching eines Referenzmodells und einer Beobachtung: Die Referenz (links) wird skaliert (mitte) und die beste Position innerhalb der Beobachtung wird bestimmt (rechts).

2 Verfahren

Das Verfahren basiert auf einem statistischen Ansatz zur Objekterkennung mit einem ganzheitlichen Modell. Dabei werden statistische Modelle für das Objekt und den Hintergrund verwendet, um alle Pixel eines Bildes zu erklären. Das Objekt wird als quadratischer Ausschnitt des Bildes angenommen. Als Modell für Objekte wird eine Gauß'sche Mischverteilung verwendet. Die Menge von Pixeln, die nicht zum Objekt gehören, werden mit einer univariaten Gaußverteilung, dem Hintergrundmodell, beschrieben.

Neben den Grauwerten eines Bildes können auch andere Merkmale (*Features*) verwendet werden. Alle Features eines Bild werden dann zu einem Bild aus mehreren Schichten (*Layern*) zusammengefasst. In dieser Arbeit wurden neben einfachen Grau- und Farbwert-Features auch die Ableitung, ein Sobel-Filter und die Gabor-Transformation verwendet. Beispiele für alle Features sind in Abbildung 1 dargestellt.

Im Matching wird für eine Sammlung von Features eines Bildes, im folgenden Beobachtung genannt, die wahrscheinlichste Position und Skalierung des gegebenen Referenzmodells gesucht. Dazu werden aus dem Referenzmodell unterschiedlich skalierte Templates erzeugt und auf mehreren Positionen deren Distanz zur Beobachtung berechnet. Abbildung 2 zeigt, wie ein Objektmodell auf eine Beobachtung projiziert wird.

Im Training werden aus einer Menge von Trainingsbildern mit bekannten Klassen die Gauß'schen Mischverteilungen für die Objektmodelle und Gaußverteilungen für die Hintergrundmodelle berechnet. Das Training wird hier zunächst nicht diskriminativ durchgeführt, sondern für jede Klasse separat. Ausgehend von einem initialen Modell, das aus einer gegebenen Sammlung von Features und einer gegebenen Varianz berechnet werden kann, werden iterativ immer bessere Verteilungen berechnet, die die Trainingsdaten erklären. In jeder Iteration wird, nachdem für alle Trainingsdaten das Matching durchgeführt wurde, aus den besten berechneten Hypothesen ein neues Objektmodell berechnet. Die berechneten Verteilungen werden als Modell der jeweiligen Klasse für die Objektklassifikation verwendet. Abbildung 3 zeigt einen Teil der Trainingsbilder für eine Klasse und das daraus berechnete Objektmodell.

Zur Klassifikation eines Bildes wird die aus dem Bild berechnete Sammlung von Featu-

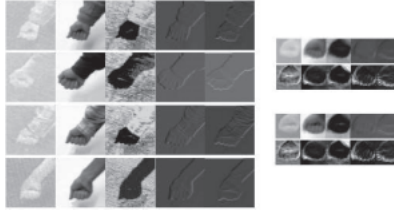


Abbildung 3: 4 von 18 Trainingsdaten (links) und das daraus berechnete Modell (rechts) mit 2 Dichten (jeweils oben der Mittelwert und darunter die Varianz).

Tabelle 1: Fehlerraten der Experimente auf der Bochum-Gestures Datenbank in Prozent. Angegeben sind die mittleren Fehlerraten der drei Testdatensätze.

Features	Fehlerrate
Grauwerte	8,6
Grauwerte, Sobel-Filter	10,3
Grauwerte, Gabor-Filter (3 Frequenzen, 2 Orientierungen)	4,7

res verwendet. Diese Featuresammlung wird mit den Modellen gematcht, die im Training für die einzelnen Klassen erstellt wurden. Der Klassifikator ordnet dann dem zu klassifizierenden Bild diejenige Klasse zu, deren Modell die geringste Distanz zum Bild hat. Die minimale Distanz eines Modells zum Bild impliziert die maximale Wahrscheinlichkeit, dass Objekt und Hintergrund vom Modell erzeugt werden [KMDN03].

3 Ergebnisse

Die Tests wurden auf der Bochum-Gestures- und der CALTECH-Faces-Datenbank durchgeführt. Die Bochum-Gestures Bilddatenbank besteht aus insgesamt 1036 Fotos (128×128 Pixel) von 12 Handgesten. Jede Geste liegt von unterschiedlichen Personen jeweils vor hellem, dunklem und komplexem Hintergrund vor. Die Bilddatenbank wird in [TvdM01] vorgestellt. Mit dem dort beschriebenen Elastic Graph Matching erzielten die Autoren Fehlerraten von 7,1% für Bilder mit einfachem Hintergrund und 14,2% für Bilder mit komplexem Hintergrund. Diese Ergebnisse sind nicht mit den in dieser Arbeit durchgeführten Experimenten zu vergleichen, da die Bilder dort manuell für das Training bearbeitet wurden. In den Experimenten wurden die 345 Bilder mit hellem Hintergrund verwendet, die in 3 Testdatensätze aufgeteilt wurden. Für die Startmodelle wurde aus allen Trainingsbildern einer Klasse der Mittelwert berechnet und daraus Feature-Sammlungen erstellt. Das Hintergrundmodell wurde als Gleichverteilung mit geringem Gewicht in die Distanzberechnung einbezogen. In Tabelle 1 sind ausgewählte Ergebnisse der Experimente dargestellt.

Die CALTECH-Faces-Datenbank des California Institute of Technology besteht aus Grauwert-Bildern von Gesichtern vor komplexen und vor einfachen Hintergründen [BLP96], von denen in den Experimenten 534 verwendet wurden. Da die CALTECH-Faces-Datenbank keine Klasseninformationen beinhaltet, wurden nur Modelle für eine Klasse „Gesicht“ trainiert. Im Test wurden Bilder mit und ohne Gesicht klassifiziert. Anschließend wurde ein Grenzwert für die ermittelten Distanzen berechnet, mit dem entschieden werden kann, ob ein Bild ein Gesicht enthält oder nicht. Die erzielten Fehlerraten zeigt Tabelle 2. Mit speziellen Verfahren zur Gesichtserkennung können deutlich bessere Fehlerraten erzielt werden. Der hier vorgestellte Ansatz macht jedoch keinerlei Annahmen zur Struktur der erkannten Objekte.

Tabelle 2: Fehlerraten auf der CALTECH-Faces-Datenbank in Prozent.

Features	Fehlerrate
Grauwerte	47,7
Grauwerte, Ableitung	32,3
Grauwerte, Gabor-Filter (2 Frequenzen, 1 Orientierung)	42,2

Aus den Fehlerraten in Tabelle 1 ist ersichtlich, dass die Informationen des Gabor-Features die Erkennung verbessern. Die Integration der anderen Features kann die Fehlerrate bei Experimenten auf der Bochum-Gestures-Datenbank allerdings nicht senken. In den Experimenten auf den Gesichts-Bildern senkt die Verwendung der Ableitung die Fehlerrate. Die Hinzunahme des Gabor-Features lässt die Fehlerrate wieder ansteigen, bleibt allerdings unter der Fehlerrate des Experiments, in dem nur Grauwerte benutzt werden.

4 Fazit und Ausblick

Das vorgestellte System ermöglicht die Integration mehrerer Features in einen holistischen Bilderkenner. Die erwartete Verbesserung der Erkennungsleistung bei steigender Anzahl von Features bestätigt sich, tritt aber nicht in allen Experimenten auf. Weitere Experimente sollen noch andere Kombinationen von Features untersuchen.

Offenbar ist der Einfluss der Features auf die Fehlerrate abhängig von der Art der untersuchten Bilder bzw. der Bilddatenbank. In der Bochum-Gestures Datenbank sind zum einen einige Gesten untereinander sehr ähnlich und somit schwer zu differenzieren. Zum anderen ist die Trainingsdatenmenge sehr klein. Trotzdem erzielt das Verfahren niedrige Fehlerraten. Man könnte die Trainingsdaten vergrößern, indem Variationen durch geringe Rotation und Skalierung erstellt werden.

Abhilfe für das Problem der teilweise fehlerhaften Segmentierung beziehungsweise der Abhängigkeit vom Hintergrund könnten nicht-quadratische Prototypen schaffen. Dazu können zunächst quadratische Prototypen berechnet und in einem weiteren Schritt verfeinert werden.

Bislang berücksichtigt das Verfahren noch keine Rotation der zu erkennenden Objekte. Bereits bei der Berechnung der Feature-Sammlungen, also im Vorverarbeitungsschritt, könnten für die Trainingsdaten Features aus rotierten Bildern berechnet werden. Eine Rotation der Features ist nicht direkt möglich, da einige Features von der Ausrichtung der Originaldaten abhängen.

Literatur

- [BLP96] M.C. Burl, T.K. Leung und P. Perona. Face Localization via Shape Statistics. In *Proc. Int. Workshop on Automatic Face and Gesture Recognition*, Seiten 154–159, Juni 1996.
- [KMDN03] D. Keysers, M. Motter, T. Deselaers und H. Ney. Training and Recognition of Complex Scenes using a Holistic Statistical Model. In *Proceedings of the 25th DAGM-Symposium Pattern Recognition*, Lecture Notes in Computer Science 2781, Seiten 52–59. Springer Verlag, September 2003.
- [TvdM01] J. Triesch und C. von der Malsburg. A System for Person-Independent Hand Posture Recognition Against Complex Backgrounds. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 23(11):1449–1453, November 2001.

Simulation dynamischer Prozesse und Kollisionserkennung im Dreidimensionalen mit einfachen geometrischen Objekten

Chavdar Papazov, Daniel Förder

FH Brandenburg
Fachbereich Informatik und Medien
papazov@fh-brandenburg.de
foerder@fh-brandenburg.de

Art der Arbeit: Semesterbegleitende Projektarbeit
Betreuer/in der Arbeit: Prof. Dr. sc. techn. Harald Loose

Abstract: Das Programm *physics studio* wurde im Sommersemester 2004 im Rahmen eines Studienprojektes des 6. Semesters erstellt. Es ermöglicht die Simulation der Dynamik von Kugeln in Wechselwirkung mit beliebigen statischen Objekten.

1 Kurzbeschreibung des Programms

Das Programm ermöglicht es, eine Szene bestehend aus Kugeln und statischen Objekten aufzubauen. Statische Objekte sind Würfel, Zylinder oder Flächen, die aus Dreiecken zusammengesetzt werden. Intern wird also mit Dreiecken gerechnet.

Kugeln können sich in der Größe und Masse unterscheiden. Beliebig viele Kugeln und Objekte können in eine Szene eingesetzt werden. Mit dem Szeneneditor können Szenen modifiziert, die Position, Größe und Masse der Objekte nachträglich verändert oder Objekte gelöscht werden. Durch Verschiebung der Kamera kann die Szene aus verschiedenen Blickwinkeln betrachtet werden.

Komplexe Szenen können in einem speziellen Format als Binärdaten gespeichert und jederzeit wieder geladen werden. Das Programm basiert auf C++, OpenGL und GLUI. Es ist unter Windows, unter Linux oder unter UNIX-Systemen lauffähig.

2 Die GUI

Die GUI wurde mit der GLUI-Bibliothek umgesetzt. GLUI ist eine open source C++ Klassenbibliothek zum Erstellen von graphischen Benutzeroberflächen. GLUI ist speziell auf OpenGL-basierte Anwendungen zugeschnitten. Sie ist plattformunabhängig und ermöglicht eine relativ einfache Verarbeitung der Benutzereingaben. In Abbildung 1

sieht man einen Screenshot aus *physics studio*, nachdem eine Beispielszene geladen wurde:

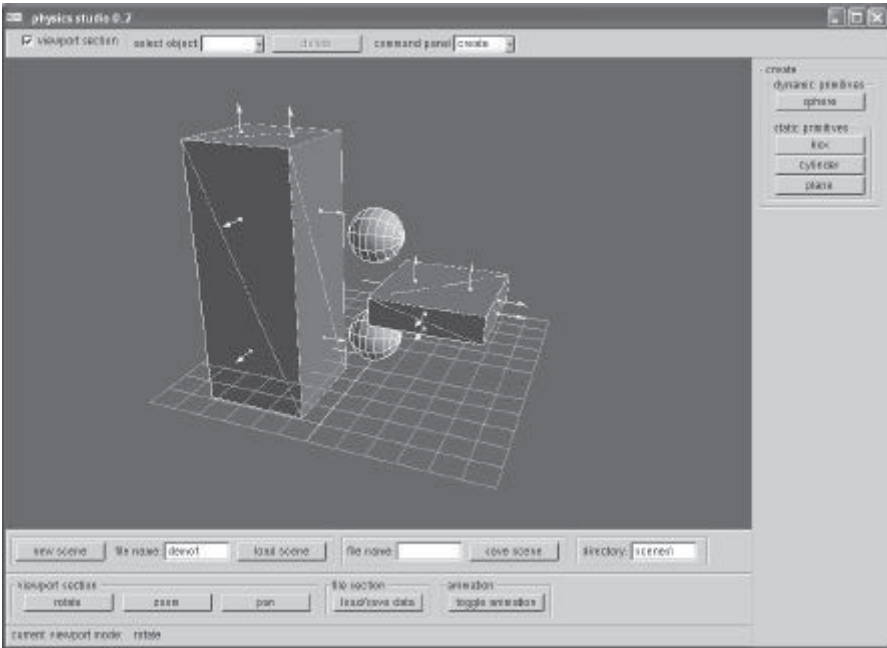


Abbildung 1: Die GUI nach dem Laden einer Beispielszene

Man sieht die verschiedenen Panels zum Auswählen und Erstellen von Objekten, zum Laden und Speichern von Szenen und zum Ändern der Kameraposition. Außerdem sieht man hier die Aufteilung der Quader in Dreiecke und die Normalenvektoren. Bei laufender Simulation kann auch der Impuls der Kugel in Form eines Pfeils dargestellt werden.

3 Die Dynamik

Die physikalischen Zusammenhänge werden durch die Newton'schen Axiome, Reibungsgesetze und Annahmen über den nichtelastischen Stoß in Form von algebraischen Beziehungen und Differentialgleichungen 2. Ordnung beschrieben. Mit einem Runge-Kutta-Verfahren 4. Ordnung werden die Bewegungsgleichungen integriert und so Position und Geschwindigkeit der Kugeln fehlerarm und effizient berechnet. Die Berechnung des Bewegungszustands der Kugeln sowie das Eintreten der Kollisionsbedingungen werden für jedes Frame, also alle 15 ms, neu ermittelt. Bei langen Simulationen und großer Dynamik kann diese Vorgehensweise zu größeren Abweichungen führen.

Kollidieren Kugeln miteinander oder mit einem Objekt (oder dem Boden), so werden Impuls- und Energieerhaltungssatz für einen Stoß angewandt. Dadurch können Kugeln durch die Szene und über verschiedene Objekte springen. Durch den Bewegungsenergieverlust bei einem nichtelastischen Stoß und die Rollreibung fangen Kugeln irgendwann zu rollen an und bleiben letztendlich stehen.

Die Rotation der Kugeln wird bei jedem Aufprall auf ein Objekt neu berechnet. Wenn eine Kugel also beim Herabfallen immer wieder zwischen zwei Wänden hin- und herprallt, so ändert sich ihre Rotation auf realistische Weise. Das geschieht beim Springen über den Boden natürlich auch, wodurch ein sanfter Übergang zwischen Springen und Rollen entsteht.

Kugeln können auch schiefe Ebenen hinauf rollen, was (falls die Ebene lang genug ist) dazu führt, dass sie langsamer werden und letztendlich wieder zurückrollen. Unten angekommen gehen sie, wenn dort ein geeignetes Objekt platziert wurde, ggf. wieder ins Springen über. Ist die Ebene nicht lang genug, so rollen sie über die Kante ab und fallen dann nach unten. Rollen funktioniert also nicht nur auf Ebenen, sondern auch auf Kanten und Ecken.

4 Die Kollisionserkennung

Der Kollisionserkennungsalgorithmus basiert vollständig auf eigenen Überlegungen. Es wurden keine Teile von anderen Algorithmen übernommen oder modifiziert.

Da die statischen Objekte einer Szene (Würfel, Zylinder und Flächen) ihre Position im Laufe der Zeit nicht ändern, wird eine Kollisionserkennung nur aus Sicht der Kugeln (den dynamischen Objekten) durchgeführt. Folglich werden statische und dynamische Objekte intern unterschiedlich repräsentiert und behandelt. Statische Objekte werden als eine Sammlung von Dreiecken interpretiert, die wohl definierte Position und Grenzen im Raum haben. Kugeln dagegen werden als ein Punkt mit dem dazugehörigen Radius betrachtet.

In jedem Frame wird überprüft, ob eine Kollision vorliegt oder nicht, d.h. ob eine Kugel sich innerhalb einer anderen Kugel befindet oder ob ein Dreieck eines statischen Objektes innerhalb einer Kugel ist.

Die Überprüfung im Fall von einer "Kugel – Kugel" Kollision ist einfach: es wird die Distanz beider Kugeln voneinander mit der Summe beider Radien verglichen.

Die "Kugel – Dreieck" Kollisionserkennung ist aufwändiger. Es werden drei Bedingungen der Reihe nach geprüft:

1. Befindet sich die Fläche eines Dreiecks in einer Kugel?
2. Liegt eine der drei Kanten eines Dreiecks in einer Kugel?
3. Ist eine der drei Spitzen eines Dreiecks in einer Kugel?

Ist eine der Bedingungen erfüllt, liegt eine Kollision vor.

Weil eine Kollision gemeldet wird, nur wenn sie vorliegt, und weil eine Überprüfung nur jedem Frame durchgeführt wird, kann es vorkommen, dass Kugeln Objekte der Szene “durchdringen“. Wenn die Geschwindigkeit einer Kugel so hoch ist, dass sie von einem Frame zum nächsten das Objekt passiert hat, wird die Kollision einfach nicht “gesehen“, weil das Durchdringen zwischen zwei Frames geschieht.

5 Zusammenfassung

Es wurde einen Überblick über den Funktionsumfang und die Arbeitsweise des Programms “physics studio“ gegeben. Die Simulation physikalischer Phänomene und die Kollisionserkennung bilden den Schwerpunkt der Funktionalität der Software. Eine denkbare Erweiterung des Programms, wäre das “Dynamisieren“ aller Objekte einer Szene, was das Berücksichtigen einer Reihe weiteren mechanischen Gesetzmäßigkeiten und Entwicklung neuer Kollisionserkennungsmechanismen zur Folge haben würde.

Literaturverzeichnis und Quellenangabe

- [En98] Engelmann, L.: Formeln und Tabellen für die Sekundarstufen I und II, Paetec, Berlin, 1998
- [H K01] Heinemann, H.; Krämer, H.: Kleine Formelsammlung Physik, Fachbuchverlag Leipzig, 2001
- [Ki96] Kilgard, M. J.: The OpenGL Utility Toolkit (GLUT) Programming Interface, Silicon Graphics, <http://www.opengl.org/documentation/specs/glut/glut-3.spec.pdf>, 1996
- [M W91] Merziger, G.; Wirth, T.: Repetitorium der höheren Mathematik, Binomi, 1991
- [N D97] Neider, J.; Davis, T.: OpenGL Programming Guide Second Edition, Addison-Wesley, http://www.opengl.org/documentation/red_book_1.0/, 1997
- [Ra99] Rademacher, P.: GLUI – A GLUT-Based User Interface Library, <http://sourceforge.net/projects/glui/>, 1999

sowie

Wikipedia – de.wikipedia.org, en.wikipedia.org

Nebula Engine – <http://sourceforge.net/projects/nebuladevice/>

Learn:line NRW – <http://www.learn-line.nrw.de/angebote/modell/runge.htm>

Normalization and Parameterization of Motion-Captured Data

Sandip Sar Dessai

sandip@gmx.de

Abstract: Motion Capturing has become a well-established technique for digitizing human movements. If only movements are to be recorded, the setup for the capturing process can be quite simplistic. If the generated data has to be anatomically exact, like required in gait analysis, the preparations get complicated and time-consuming. We propose a novel approach which does not require a complicated setup but still manages to generate data suitable for a meaningful analysis. This is done by introducing a two-step processing pipeline. In a first step, the data is normalized to create a common baseline for the recordings. Then, the data is parameterized, reducing the required data for specifying a pose to a set of only few parameters. In both steps, special emphasis is put on anatomical relevance of the resulting data.

1 Introduction

During the last decade, optical motion capturing has become a major technique to digitize human movements. A human is equipped with retro-reflexive markers, tracked by a set of optical cameras. Based on the markers' positions in the respective camera images a software reconstructs their 3D-positions, allowing to obtain the limbs' trajectories.

Two of the main application areas of motion capturing are the animation of virtual characters for movie productions, and gait/motion analysis in medical applications.

For movie productions, the main issue is to have a simple recording setup which allows to create realistically looking digital motions. Normally, the whole body is equipped with markers attached near joints. Based on their trajectories the movements can be copied to a stick-figure model. However, in most cases significant post-processing is required to overcome faulty poses caused by occlusions and data noise. This has to be done manually and results in a time-consuming and unintuitive procedure.

In biomedical applications, great care must be taken to obtain the proband's exact joint positions and axes to find motion patterns and perform valuable analysis. The proband is normally equipped with groups of at least three markers ("*marker bodies*") to compute positions and orientations. Since the experiments should be reproducible and comparable, the markers must be attached to specific positions, resulting in tedious preparations.

We propose a method to combine both aspects – a simple recording setup and generation of both valuable and intuitively editable motion data – by introducing two semi-automatic

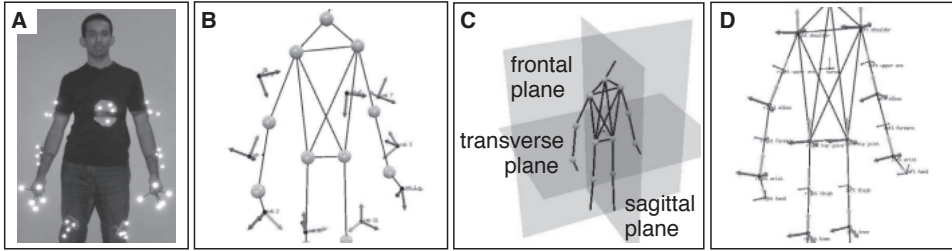


Figure 1: Skeleton Normalization. (a) Recording setup. (b) Motion capture raw data (based on a). The coordinate systems reflect positions and orientations of the marker bodies. (c) Body planes. (d) Normalized data. After the normalization the coordinate systems are aligned with the limbs.

processing steps between the recording and the analysis stage.

The stages are based on an uncalibrated recording in which the human has been equipped with one arbitrarily placed marker body per limb, consisting of four rigidly connected markers (Fig. 1a). Such a recording requires only a few minutes for preparations.

Our work is based on [HSK05], the pipeline of which results in reliably computed coordinate systems for the marker bodies (Fig. 1b). Note though that these bodies do not match the orientations of the actual limbs.

2 Skeleton Normalization

The first step of the processing pipeline, which we call *normalization*, first extracts the body planes of the human. Body planes (Fig. 1c) are the planes separating the front/back (*frontal plane*), left/right (*sagittal plane*) and top/bottom (*transverse plane*) of the body, and are a well-known basis to define limb coordinate systems [HvSLD99].

The body planes are defined for a *default position* [HvSLD99, Ani04], in which the actor stands upright and has the arms hanging down alongside the body. This position is assumed to appear somewhere in the recording, and is located automatically by considering features of the poses, like the shape of the bounding box. Once the position is found, the global body coordinate system, based on the body plane normals, is built, and serves as a global reference for all limbs.

The limb coordinate systems, yet based on the positions and orientations of the marker bodies (Fig. 1b), are then repositioned and reoriented to match the body planes and the orientations of the actual limbs. The recalculation is again based on the default position frame, and the transformations from current to the new coordinate systems in that frame are propagated to all other frames (Fig. 1d).

The axes of the limbs are also used to define the coordinate systems of the attached joints. Since the limb coordinate systems are closely connected to the body planes, this choice is well suited.

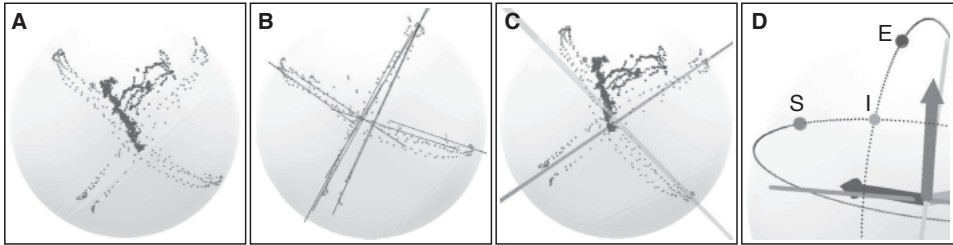


Figure 2: Skeleton Parameterization. The sphere centered at the joint center characterizes all valid positions of the attached limb, the samples on it illustrating the positions encountered in the recording. (a) A wrist joint distribution. Samples recorded during movement are drawn lighter, stationary samples are darker. (b) Direction Joining. Initial direction vectors. (c) Axis planes found with Direction Joining; the axes are orthogonal to them. (d) Finding Rotation Angles. Start point S, intermediate point I, end point E.

3 Skeleton Parameterization

The second step, *parameterization*, defines anatomically meaningful joint axes, and parameterizes the motion with angles around these axes. In most medical and biomechanical scenarios, the choice of these axes is recommended to be based on bony landmarks [HvSLD99, ISB02, ISB04], which are impossible to find in practical recording sessions.

In practical motion analysis, angle parameterizations are just based on the axes of the joint coordinate system. These coordinate systems are most often either those of the corresponding limbs (for simplicity reasons), or explicitly constructed using knowledge about the placement of the markers (which is clearly not general enough).

The main idea behind our approach is to infer the recommended anatomical axes by investigating the movements of the human actor. By asking the person to move the arm along the anatomical axes in a special recording phase, appropriate axes can be found by analyzing the generated set of limb positions within the joint coordinate system (Fig. 2a).

Before analyzing the distribution, the samples are classified into movement and stationary samples (Fig. 2a). Since the algorithms extracting the main directions work on movement samples only, stationary ones perturb the results substantially, and their elimination is of great use. A given sample is classified moving if the distance to the previous sample exceeds a threshold computed based on the distribution of speeds during the recording.

To overcome difficulties with different moving speeds, which lead to a nonuniform sample density, the samples are stratified. Stratification is a process in which the joint sphere, i.e. all possible sample positions, is partitioned into approximately uniform cells, and then one representative is chosen per cell. The cell size is chosen based on the samples' speed distribution such that neither oversampling nor undersampling occurs. We modified the algorithm to only merge samples as long as they continuously fall into a certain cell; if the cell is revisited later, another representative is computed. This keeps the temporal order of the traversal.

Standard extraction techniques, like the *Principal Component Analysis* (PCA), unfortunately fail to find the axes in most cases, albeit classification and stratification. The main problem here is that the axes extracted by the PCA are always orthogonal, a seldom case in reality. *Expectation Maximization*, an iterative optimization technique, leads to better results, but still often fails. The most probable reason is that there are too many local minima along the way, and good initial estimates are hard to find.

Thus we designed a specialized algorithm, *Direction Joining*, capable of finding main movements by investigation of the movement vectors between successive samples.

In a first step of *Direction Joining*, consecutive moving samples are grouped together if their movement vectors do not deviate too much. For each of the groups a general moving vector is computed by averaging the individual movement vectors (Fig. 2b). Now we try to join pairs of groups. Two groups are joined, if the angle between one movement vector and the projection of the second movement vector into the tangent plane of the first one is not too large, and their centers are not too far apart. In the end, only few groups remain. Among those, groups representing the main movement directions can be identified and extracted easily, because they contain far more samples than the other groups. The rotation axes are orthogonal on the planes spanned by the joint center and the main movement directions ("axis planes", Fig. 2c). In the experiments performed so far, *Direction Joining* lead to the extraction of very good movement axes.

Once the two main axes of a joint have been extracted, the joint motion needs to be parameterized according to these axes, i.e. any limb position must be converted into angles around them. The two rotations are based on a common starting point, called the *rest position* of the joint, which reflects the position in which the limb stayed still the longest time during recording (this corresponds to the center of the largest cluster within the stationary samples).

The task hence is to find the two angles around two given axes, with start point S and end point E of the combined known rotation. Our approach is based on the observation that both rotations describe circles, one including S around the first axis, one including E around the second axis (Fig. 2d). Their intersections are candidates for the intermediate point I, being end point of the first and start point of the second rotation. The computation of this intersection is based on [Ebe99], and leads to the required angles.

4 Conclusion

We have presented a two-step processing pipeline for the normalization and parameterization of arbitrary motion captured data, allowing a meaningful data analysis.

The normalization stage finds well aligned coordinate systems for an arbitrary recording setup based on local limb information only (and the global body planes). The method does not use any extra marker information, nor does it explicitly build coordinate systems based on knowledge about the global positioning of markers on the body, and thus is not limited to a certain recording setup. Both simple and fast, the normalization has two main advantages. First, different recording sessions can now easily be compared, as they

lead to the same coordinate systems. Second, tasks which are based on limbs are greatly simplified (e.g. equipping the skeleton with mesh surfaces for a better visual appearance).

The parameterization stage basically allows to define user-defined axes. The axes are calculated based on specific, prescribed movements of the actor. In this way, anatomical axes can be identified and used for parameterizing the movements. Although the axes will not be as exact as those found using mechanical instruments and invasive experiments, they should be exact enough for general motion analysis.

In summary, the two computation steps greatly simplify the procedure that had yet been necessary to generate motion parameters for analysis. These parameters can be used, for example, to identify motion patterns (like running, jumping etc.) based on a sample set of movements by simply comparing the resulting parameters. Other areas of application include the computation of joint limits and limit dependencies between adjacent joints. The computed axes also simplify character animation, since natural movements can now be described by changing a single angle.

We are also currently evaluating the medical accuracy of the computed axes by checking how close they are to the real anatomical axes which can be found in joints.

References

- [Ani04] H-Anim Humanoid Animation Working Group. H-Anim 200x specification ISO/IEC FCD 19774:200x. <http://www.h-anim.org>, February 2004.
- [Ebe99] David Eberly. Distance Between Two Circles in 3D. Magic Software, Inc., 1999.
- [HSK05] Alexander Hornung, Sandip Sar-Dessai, and Leif Kobbelt. Self-Calibrating Optical Motion Tracking for Articulated Bodies. To appear on IEEE VR 2005, March 2005.
- [HvSLD99] I. Hilal, S. van Sint Jan, A. Leardini, and U. Della Croce. VAKHUM-Project: Data Collection Procedure ANNEX I. Technical report, IST Programme, 1999.
- [ISB02] ISB Standardization and Terminology Committee. ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion – part I: ankle, hip, and spine. *Journal of Biomechanics*, 35(4):543–548, 2002.
- [ISB04] ISB Standardization and Terminology Committee. ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion – part II: shoulder, elbow, wrist and hand. *Journal of Biomechanics*, September 2004. In Press.

GI-Edition Lecture Notes in Informatics

Seminars

Vol. S-1: Informatics and Student Assessment, Dagstuhl-Seminar

Vol. S-2: Informatiktage 2005 – Fachwissenschaftlicher Informatik-Kongress

- P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.
- P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.
- P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Language to Information Systems, NLDB'2001.
- P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.
- P-5 Andy Schürr (Hg.): OMER - Object-Oriented Modeling of Embedded Real-Time Systems.
- P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.
- P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods - Countering or Integrating the extremists, pUML'2001.
- P-8 Reinhard Keil-Slawik, Johannes Magenheimer (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.
- P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.
- P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.
- P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.
- P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.
- P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahnič (Hrsg.): The Changing Universities – The Role of Technology.
- P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.
- P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.
- P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.
- P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.
- P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.
- P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3.Okt. 2002 in Dortmund.
- P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3.Okt. 2002 in Dortmund (Ergänzungsband).
- P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.

- P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.
- P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 - Fortschritt durch Beständigkeit
- P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen
- P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003
- P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web
- P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin
- P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement - Erfahrungen und Visionen
- P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems
- P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications
- P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometric and Electronic Signatures
- P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003
- P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft
- P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)
- P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)
- P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik
- P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik
- P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003
- P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003
- P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004
- P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste
- P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistence, Scalability, Transactions – Database Mechanisms for Mobile Applications
- P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning. E-Services
- P-45 Bernhard Rumpe, Wolfgang Hesse (Hrsg.): Modellierung 2004
- P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment
- P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society
- P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications
- P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Daten-

- sicherheit – Anforderungen, Konflikte und Perspektiven
- P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik
- P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004
- P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration
- P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Ad-hoc-Netze Medienintegration
- P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 - Informationssysteme im E-Business und E-Government
- P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen
- P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality
- P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments
- P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen
- P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolffried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web

The titles can be purchased at:

Köllen Druck + Verlag GmbH
 Ernst-Robert-Curtius-Str. 14
 53117 Bonn
 Fax: +49 (0)228/9898222
 E-Mail: druckverlag@koellen.de