GI-Edition

Lecture Notes in Informatics

Gesellschaft für Informatik (Hrsg.)

Informatiktage 2014

Fachwissenschaftlicher Informatik-Kongress

27. und 28. März 2014
Hasso Plattner Institut der Universität Potsdam

Gesellschaft für Informatik (Hrsg.): Informatiktage 2014

13

Seminars

Gesellschaft für Informatik (Hrsg.)

Informatiktage 2014

Big (Data) is beautiful

Fachwissenschaftlicher Informatik-Kongress

27. und 28. März 2014

Hasso-Plattner-Institut der Universität Potsdam

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Seminars
Series of the Gesellschaft für Informatik (GI)

Volume S-13

ISSN 1614-3213
ISBN 978-3-88579-447-9

Volume Editor
Gesellschaft für Informatik e.V.
Ahrstraße 45
53175 Bonn
E-Mail: gs@gi.de
Redaktion: Ludger Porada
E-Mail: ludger.porada@gi.de

printed by Köllen Druck+Verlag GmbH, Bonn

Wissenschaftliche Tagungsleitung
Alfred Zimmermann, Reutlingen University

Programmkomitee
Ehrenvorsitz
Rul Gunzenhäuser – Universität Stuttgart
Otto Spaniol – RWTH Aachen

Sören Auer – Universität Leipzig
Karlheinz Blank – T-Systems Stuttgart
Hermann Engesser – Informatik-Spektrum
Robert Hirschfeld – HPI Potsdam
Walter Hower – Hochschule Albstadt-Sigmaringen
Agnes Koschmider – KIT Karlsruhe
Reinhold Kröger – Hochschule RheinMain
Wolfgang Küchlin - Universität Tübingen
Frank Leymann – Universität Stuttgart
Florian Matthes – TU München
Martin Mähler – IBM Böblingen
Alexander Paar – TWT Science & Innovation Stuttgart
Gunther Piller – Fachhochschule Mainz
Karl Prott – Capgemini Hamburg
Kurt Sandkuhl – Universität Rostock
Rainer Schmidt – Hochschule Aalen
Ulrike Steffens – HAW Hamburg
Gottfried Vossen – Universität Münster
Martin Wollschlaeger – TU Dresden
Alfred Zimmermann – Hochschule Reutlingen (Vorsitz)
Wolf Zimmermann – Universität Halle
Eva Zauke - SAP Walldorf

Vorwort zum Tagungsband 2014

Liebe Teilnehmerinnen und Teilnehmer,

die Informatiktage 2014 werden in diesem Jahr größer, bunter und vielfältiger. Mit dieser inzwischen traditionsreichen Nachwuchsveranstaltung der Gesellschaft für Informatik treffen wir uns erstmals am Hasso-Plattner-Institut an der Universität Potsdam, das hervorragende Bedingungen für unsere Veranstaltung bietet.

Es erwarten Sie Vorträge und Talk-Runden zum Leitmotto der Veranstaltung. Es ist kein Zufall, dass dieses dem Thema Big Data gewidmet ist, welches die Gesellschaft für Informatik derzeit in vielen Veranstaltungen, darunter auch im Rahmen der Jahrestagung INFORMATIK 2014, vorantreibt. Mit Ihnen allen wollen wir die zahlreichen Aspekte von Big Data diskutieren. Ausgewählte Studierende werden ihre Arbeiten im Wissenschafts- und Absolventen-Workshop sowie in der Postersession präsentieren. Darüber hinaus finden Sie Beiträge zum Thema Big Data im vorliegenden Tagungsband.

Der Brückenschlag zwischen Wirtschaft und Informatik-Nachwuchs war stets ein erklärtes Ziel der Informatiktage. Ich bin zuversichtlich, dass es uns auch in diesem Jahr gelingen wird, dies zu erreichen.

Ich freue mich darauf, mit Ihnen gemeinsam an den 14. Informatiktagen teilzunehmen und wünsche Ihnen eine anregende, spannende und informative Veranstaltung.

Prof. Dr.-Ing. Peter Liggesmeyer
(Präsident der Gesellschaft für Informatik e.V.)

Inhaltsverzeichnis

Big Data – Datenbanken und Informationssysteme

Grundlagen der Informatik

Künstliche Intelligenz

Softwaretechnik

Mensch-Computer-Interaktion

Technische Informatik

Informatik in den Lebenswissenschaften

Graphische Datenverarbeitung

Wirtschaftsinformatik

Informatik und Ausbildung / Didaktik der Informatik

Informatik und Gesellschaft

Multidimensional Process Mining mit dem Process Cube Explorer

Jannik Arndt, Thomas Meents, Bernd Nottbeck
Universität Oldenburg
{jannik.arndt, thomas.meents, bernd.nottbeck}@uni-oldenburg.de

Abstract: In vielen Anwendungsfällen werden Arbeits- und Prozessabläufe in Log-Dateien protokolliert. Das Process Mining findet in diesen Eventlogs das zugrundeliegende Prozessmodell, welches dann zur Kontrolle und Optimierung der Abläufe genutzt werden kann. Der Process Cube Explorer bietet ein Framework für Multidimensional Process Mining, mit dem Eventlogs aus einem Data Warehouse extrahiert und in Prozessmodellen dargestellt werden. Der multidimensionale Ansatz erlaubt es dem Anwender, die Datenbasis anhand der Dimensionen einzugrenzen und so die Auswirkung verschiedener Eigenschaften auf das Prozessmodell zu entdecken.

1 Einleitung

Beim Process Mining handelt es sich um eine Mischung aus Machine Learning, Data Mining und automatisierter Prozessmodellierung. Ziel ist es Wissen über Prozesse aus Eventlogs zu gewinnen [vdA11]. Dabei unterteilt sich das Process Mining in die drei Anwendungsbereiche *Process Discovery* (Finden neuer Prozessmodelle), *Conformance Checking* (Überprüfung von existierenden Prozessmodellen) und *Model Enhancement* (Erweiterung der bestehenden Modelle).

Das in dieser Arbeit vorgestellte Programm *Process Cube Explorer* ist primär im Bereich *Discovery* einzuordnen, in dem aus einer Menge von automatisch oder manuell aufgezeichneten Events das zugrundeliegende Prozessmodell erkannt wird. Hierfür sind die einzelnen Events in chronologischer Abfolge in einem *Eventlog* gespeichert, auf das verschiedene Mining-Algorithmen angewandt werden können. Der multidimensionale Analyseansatz der dieser Arbeit zur Grunde liegt, erleichtert die Analyse großer Datenmengen.

Seit dem Frühjahr 2013 entwickelt eine Projektgruppe aus elf Studenten im Rahmen des Dissertationsvorhabens von Thomas Vogelgesang an der Universität Oldenburg ein Forschungsframework für Multidimensional Process Mining.

2 Herausforderungen

Process Discovery hat in realen Anwendungen zwei große Herausforderungen, die beide aus dem Big-Data-Umfeld stammen: Erstens ist die Menge an automatisch generierten Events

in Eventlogs in der Regel nur sehr schwierig zu überschauen und in ihrer Gesamtheit ohne Vorauswahl oft nicht aussagekräftig. Zweitens steigt der Rechenaufwand der Algorithmen mit der Menge der Events die betrachtet werden deutlich. Es ist also essenziell, dass der Benutzer dieser Anwendung die Möglichkeit hat, seine Daten auf eine sinnvolle Teilmenge zu begrenzen.

An diesem Punkt setzt das Multidimensional Process Mining an: Joel Ribeiro schlägt die Verwaltung vorberechneter Teilergebnisse in einer multidimensionalen Struktur, dem sogenannten "Event Cube", vor [RW11]. Hierzu wird bereits beim ETL-Prozess eine Vielzahl von Aggregationen und weiterer Funktionen ausgeführt. Dieser Ansatz benötigt also eine umfangreiche Vorberechnung aller Daten. Der Ansatz von Thomas Vogelgesang [Vog13] speichert bereits die Eventlogs multidimensional und führt das Process Mining dann auf einer durch den Benutzer eingegrenzten Teilmenge durch. Dafür wird das Konzept des Data Warehousings (DWH) genutzt und die Eigenschaften aus dem Eventlog als Dimensionen dargestellt, anhand derer gezielt zusammengehörige Eventdaten ausgewählt werden können.

Im Programm werden die Abläufe weitestgehend automatisiert und der Benutzer gezielt durch den Prozess geführt. So können alle Schritte im selben Programm vollzogen werden, vom Laden der Daten über die Auswahl durch das bekannte multidimensionale Modell ("Cube"), die Auswahl zwischen verschiedenen Mining-Algorithmen, dem eigentlichen Mining bis hin zur übersichtlichen Darstellung der Daten und weiteren Vergleichs- und Analysemöglichkeiten wie z. B. *Comparing Footprints*, ein Vergleich zwischen dem generierten Prozessmodell und dem zugrundeliegendem Eventlog.

3 Herangehensweise

Die Datenanalyse ist mit allen Eventlogs möglich, die in einem ETL-Prozess in ein DWH überführt werden, das sich grob an das vereinfachte Schema in Abbildung 1 halten muss. Zur Überprüfung der Miningergebnisse wurden Beispieldatensätze erstellt und reale aus der Physionet bereitgestellten *MIMIC II*-Datenbank[1] verwendet. Die *MIMIC II* enthält eine große Menge anonymisierter Krankenhausdaten zu Behandlungsabläufen und verwandten Ereignissen.

Den Kern des DWH bildet eine Faktentabelle *Fact* die auf mehrere Dimensionstabellen verweist. In diesen sind die Eigenschaften der Patienten wie Alter und Geschlecht, der Krankheit und der Diagnosen gespeichert. Die Zellen der Faktentabelle fassen Fälle (*Cases*) zusammen, bei denen alle Dimensionsausprägungen identisch sind. Ein *Case* wiederum fasst alle *Events* eines Prozessablaufs zusammen. Dies entspricht z. B. der Behandlung eines Patienten. Zusätzlich können auch Events Dimensionen besitzen, z. B. der verantwortliche Arzt für einen Behandlungsschritt. Hierdurch werden feinere Analyseeinstellungen möglich.

An das Framework können verschiedene Datenbankentypen generisch angebunden werden, zur Zeit werden u. a. Oracle, MySQL und PostgreSQL unterstützt. Die geladenen Daten

[1] `http://mimic.physionet.org`

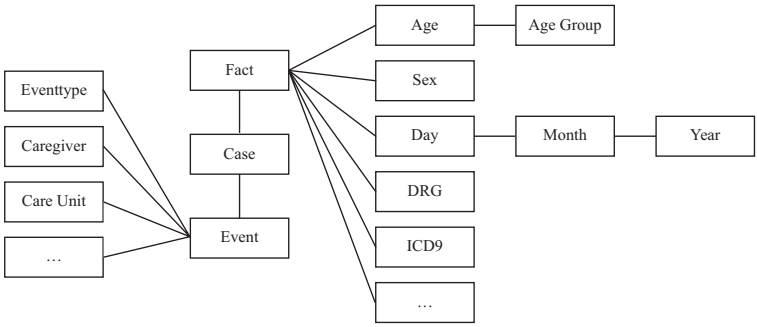


Abbildung 1: Die Datenstruktur des Data Warehouse, aus dem die Eventdaten geladen werden.

können dann im Programm anhand der Dimensionswerte, also der Eigenschaften der Fälle, in verschiedenen Aggregationsstufen sowie durch weitere Filter ausgewählt werden.

Im nächsten Schritt wählt der Benutzer einen Mining-Algorithmus aus. Bereits implementiert sind drei Varianten des bekannten Alpha-Algorithmus [dMvDvdA04], eine erweiterte Version des HeuristicMiners [WvdAdM06] und eine Implementierung des kürzlich veröffentlichten "Infrequent Miner - inductive" [LFvdA13]. Weitere Algorithmen können leicht über eine Schnittstelle in das Framework integriert werden.

Für jede Dimensionsausprägung, also jede gewählte Zelle des Würfels, erstellen die Mining-Algorithmen ein Prozessmodell, welches als Petrinetz dargestellt wird (siehe Abbildung 2). Diese Modelle können exportiert, gedruckt und miteinander verglichen werden. Die Software bietet nun die Möglichkeit, die Qualität der Modelle zu beurteilen. Mittels *Conformance Checking* kann überprüft werden, auf wie viele der Cases ein generiertes Prozessmodell zutrifft. Außerdem werden dem Benutzer für jedes Prozessmodell weitere Qualitätskennzahlen, z. B. der Anteil der berücksichtigten Events, angezeigt.

4 Ergebnisse

In der Zeit vom April 2013 bis März 2014 wurde ein Framework entwickelt, das zum einen den Benutzer beim kompletten *Process Discovery*-Prozess auf multidimensionalen Daten unterstützt, und zum anderen sehr einfach zu erweitern ist. Der Quellcode des Projekts wird zum Ende der Laufzeit im März 2014 veröffentlicht, sodass insbesondere die Analysemöglichkeiten weiter ausgebaut und weitere Algorithmen integriert werden können. Mit diesem Tool können Forscher und Wissenschaftler, bspw. im Health Care-Bereich, die Software einsetzen um bestehende Prozessmodelle unter Berücksichtigung der individuellen Eigenschaften der Patienten darzustellen, zu analysieren und zu verbessern.

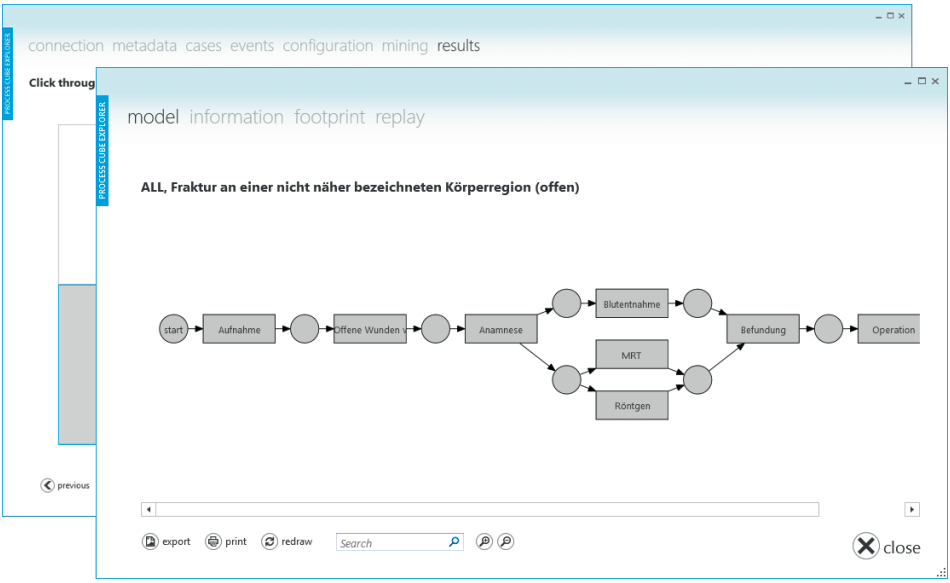


Abbildung 2: Die generierten Prozessmodelle werden dem Nutzer übersichtlich angezeigt, außerdem kann er Modelle vergleichen und einen *Comparing Footprint* erstellen.

Literatur

[dMvDvdA04] A.K. Alves de Medeiros, B.F. van Dongen und Wil M. P. van der Aalst. Process Mining : Extending the α-algorithm to Mine Short Loops. *BETA Working Paper Series, Eindhoven University of Technology*, 2004.

[LFvdA13] Sander J. J. Leemans, Dirk Fahland und Wil M. P. van der Aalst. Discovering Block-Structured Process Models From Event Logs Containing Infrequent Behaviour. In *34th International Conference, PETRI NETS 2013*, BPI Workshop. Springer Berlin Heidelberg, 2013.

[RW11] J.T.S. Ribeiro und A.J.M.M. Weijters. Event Cube: Another Perspective on Business Processes. In Meersman et al., Hrsg., *On the Move to Meaningful Internet Systems: OTM 2011*, Jgg. 7044 of *Lecture Notes in Computer Science*, Seiten 274–283. Springer Berlin Heidelberg, 2011.

[vdA11] Wil M. P. van der Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes.* Springer, 2011.

[Vog13] Thomas Vogelgesang. Multidimensional Process Mining A flexible analysis approach for health services research. In *EDBT/ICDT 2013*, Genoa, Italy, 2013.

[WvdAdM06] A.J.M.M. Weijters, W.M.P. van der Aalst und A.K. Alves de Medeiros. Process mining with the heuristics miner-algorithm. In *Technical ReportWP 166, BETA Working Paper Series*. Eindhoven University of Technology, 2006.

Protokollierung von CAN-Nachrichten mithilfe eines integrierten Data Warehouse

Martin Beckmann

TU Berlin
Fachgebiet Softwaretechnik, DCAITI
martin.beckmann@mailbox.tu-berlin.de

Art der Arbeit: Bachelorarbeit
Betreuer der Arbeit: Thomas Noack

Abstract: Diese Arbeit beschäftigt sich mit der Entwicklung eines Werkzeuges zur Verarbeitung von CAN-Nachrichten von Steuergeräten. Die Nachrichten werden anschließend protokolliert, um eine Auswertung im Rahmen von Softwaretests zu ermöglichen. Dazu wurde eine Hardwarelösung entwickelt und eine passende Software implementiert.

1 Motivation

Für die vielfältigen und komplexen Aufgaben in Automobilen werden zunehmend Steuergeräte zur Bewältigung dieser Aufgaben genutzt [Saa03]. Es ist notwendig, dass diese spezialisierten Steuergeräte dabei untereinander kommunizieren können. Als Beispiel sei genannt, dass die Lautstärke eines Entertainmentsystems in Abhängigkeit von der Geschwindigkeit des Fahrzeuges angepasst wird. Daran sind Komponenten der Fahrdynamik, der Nutzerinteraktion und des Infotainmentsystems beteiligt. Ermöglicht wird diese Kommunikation durch die Verwendung des CAN-Bus [Alb04]. Bei dem Controller Area Network (CAN) handelt es sich um ein serielles Bussystem, welches speziell auf die Anforderungen von Steuergeräten in Automobilen ausgelegt ist [11806].

Um Fehlfunktionen bei dieser Komplexität erkennen und beheben zu können, werden die Komponenten Integrationstests unterzogen. Bei diesen Tests werden voneinander abhängige Komponenten kombiniert und getestet. In diesem Zusammenhang geht es um die Kommunikation auf dem CAN-Bus zwischen den Steuergeräten in einem Verbund. Die Integrationstests sind unvermeidbar, um Qualitätseinbußen und auch Fehlfunktionen sicherheitsrelevanter Funktionen zu vermeiden [AS12]. Die Kommunikation der Steuergeräte ist dabei sowohl während der Integration in ein Teilsystem als auch in das Gesamtsystem von besonderer Bedeutung.

Der Einsatz von vielen Komponenten erzeugt ein hohes Datenaufkommen auf dem Bus. Um die Wechselwirkungen unter den Steuergeräten und deren Funktion kontrollieren zu können, ist eine Aufzeichnung der CAN-Nachrichten notwendig. Dies ist weiterhin erforderlich, da eine große Anzahl an relevanten Testfällen existiert und somit die Möglichkeit der späteren Diagnose besteht. Um das Verhalten reproduzierbar zu überprüfen, muss die Buskommunikation über einen längeren Zeitraum festgehalten werden. Dadurch wird im Nachhinein eine korrekte Testdurchführung nachweisbar.

Ausgehend von dieser Situation wurde eine Hardware entworfen und umgesetzt, welche an ein CAN-Netzwerk angeschlossen werden kann. Für diese Hardware sollte anschließend eine Software implementiert werden, die eine Überwachung im Betrieb ermöglicht, Daten dauerhaft speichert und einen Export zum späteren Zeitpunkt erlaubt. Die Lösung wurde im Rahmen einer Bachelorarbeit in Kooperation mit der *P3 systems GmbH* umgesetzt. Es folgt die Beschreibung der Implementierung.

2 Implementierung

Um eine Lösung zur Protokollierung in eine integrierte Datenbasis von CAN-Bus Nachrichten zu erhalten, ist es notwendig eine Kombination aus Hardware und Software zu verwenden. Die Architektur der Lösung ist in Abbildung 1 dargestellt.

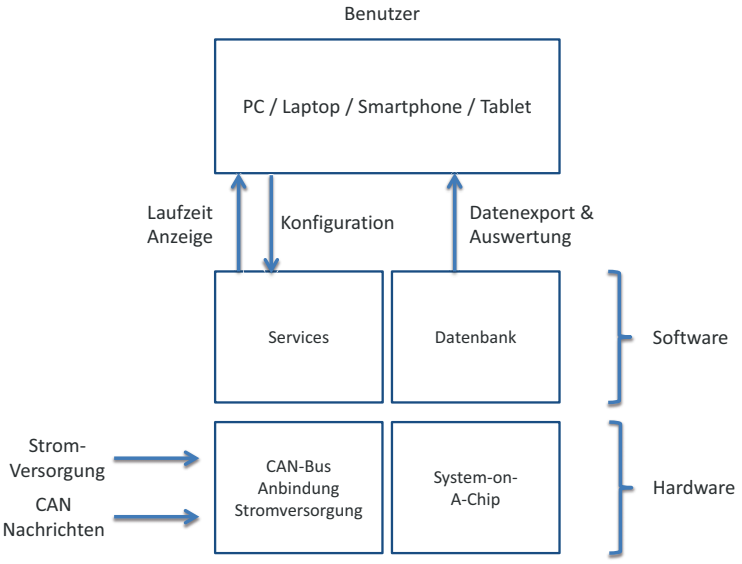


Abbildung 1: Architektur der Lösung

2.1 Hardware

Als Grundstein wird ein weitverbreitetes System-on-a-Chip (Raspberry Pi) verwendet [Fou14]. Wegen seiner Kompaktheit ist es flexibel sowohl im Labor als auch im Fahrzeug einsetzbar. Auf der Hardware wird ein Linux-Betriebssystem ausgeführt, welches u.a. dazu dient eine Datenbank zur Protokollierung zu betreiben.

Zur Anbindung an einen CAN-Bus wurde der Raspberry Pi um selbstentwickelte Hardware erweitert, welche aus zwei Platinen besteht. Diese Platinen werden aufgesteckt. Die eine Platine stellt hierbei die Stromversorgung bereit. Darauf aufgesteckt wird die zweite, ebenfalls in Eigenentwicklung entstandene Platine zur Verarbeitung der CAN-Nachrichten.

2.2 Software

Zum sinnvollen Betrieb der Hardware ist außerdem noch eine Reihe an Softwarebestandteilen entwickelt worden. Auf unterster Ebene ist dies ein Programm, welches die CAN-Nachrichten von der CAN-Hardware einliest. Diese CAN-Nachrichten werden anschließend in ein für Menschen lesbares Format umgewandelt und in ihre jeweiligen Bestandteile zerlegt. Das Zerlegen der Nachricht dient dazu selbige in einer Datenbank zu speichern.

Des Weiteren wurde eine graphische Benutzeroberfläche entwickelt. Darin eingebettet sind Funktionalitäten zur Konfiguration des Gerätes. Dies reicht von Einstellungen bezüglich des CAN-Busses zum Beispiel der Symbolrate bis zu Einstellungen der Datenorganisation.

Der Datenbestand, welcher zur weiteren Auswertung genutzt wird, ergänzt außerdem elementare Informationen wie Sitzungsdauer und die Summe der protokollierten Nachrichten zu gespeicherten Testläufen.

Die Datenbank selbst speichert nicht nur Informationen über die CAN-Nachrichten, wie Bezeichner, Daten der Nachricht, usw. sondern auch essentielle Informationen wie den Zeitstempel der Nachricht oder auf welchem CAN-Bus diese gesendet wurde.

Dabei werden wichtige Eigenschaften eines Data Warehouse erfüllt, die notwendig sind, um einen Softwaretest erfolgreich und belastbar durchführen zu können. Am wichtigsten sind dabei die nicht flüchtige Datenbasis, welche die Unveränderlichkeit der Daten garantiert sowie die historische Datenbasis, welche für Vergleiche herangezogen werden kann.

Neben der eigens entwickelten Software wurden auch bereits etablierte Werkzeuge eingesetzt. Sämtliche eingesetzte Software steht frei zu kommerziellen Nutzung zur Verfügung und konnte daher ohne Weiteres verwendet werden.

3 Fazit und Ausblick

Die erstellte Lösung ist flexibel, klein und mobil genug, um die Ansprüche, die im Labor und im realen Testfahrzeug vorliegen, erfüllen zu können. Neben der reinen Protokol-

lierung der Kommunikationsdaten, kann im Betrieb der Nachrichtenverlauf nachvollzogen werden. Aufgrund der Verwendung von etablierten Bauteilen und freien Werkzeugen, konnte dies kostengünstig realisiert werden.

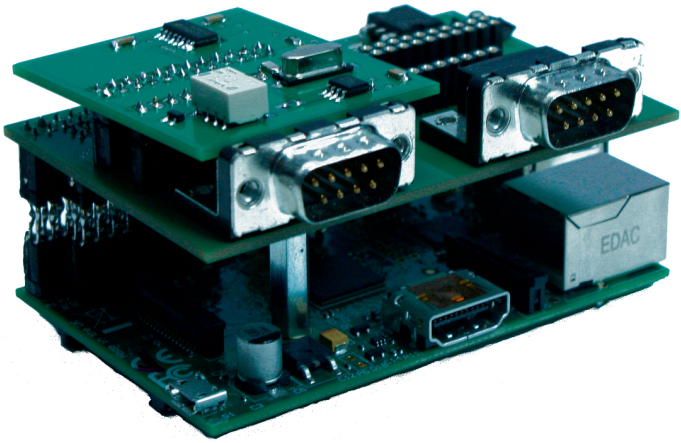


Abbildung 2: Komplett aufgebaute Hardware

Ein Einsatz der Lösung ist möglich ohne weitere Geräte zu nutzen. Deren Einsatz bietet jedoch zusätzliche Funktionalität. Die Art des Gerätes (Laptops, Smartphone, Tablet) ist hierbei frei wählbar, da die Lösung komplett plattformunabhängig arbeitet.

Eine Erweiterung der Arbeit ist sowohl auf der Hard- als auch der Softwareebene möglich. Da das Hardwarekonzept sehr flexibel gestaltet wurde, ist das Hinzufügen von Elektronik zum Protokollieren von anderen Bussystemen (z.B. FlexRay) oder von Messelektronik denkbar. Auf Seiten der Software sind als weitere Schritte die Dateninterpretation und Datenanalyse möglich.

Literatur

[11806] ISO 11898-1. *ISO 11898-1:2003 Road vehicles Controller area network Part 1: Data link layer and physical signalling*, 2006.

[Alb04] Amos Albert. Comparison of event-triggered and time-triggered concepts with regard to distributed control systems. *Embedded World*, 2004:235–252, 2004.

[AS12] Tilo Linz Andreas Spillner. *Basiswissen Softwaretest*. dpunkt.verlag GmbH, 5. Auflage, 2012.

[Fou14] The Raspberry Pi Foundation. A birthday present from Broadcom, 2014. Available online at http://www.raspberrypi.org/archives/6299; visited on March 1th 2014.

[Saa03] Alexandre Saad. Das Automobil als Anwendungsgebiet der Informatik-ein Auto ohne Informatik, geht das? In *INFOS*, Seiten 37–40, 2003.

Supporting Big Data Applications using Hybrid Architectures

Michael Bromberger *

Chair for Computer Architecture and Parallel Processing
Karlsruhe Institute of Technology
76131 Karlsruhe, Germany
bromberger@kit.edu

1 Introduction

In our modern information society a lot of data are generated every second. On-line transactions, Emails and Social Networks produce so called 'Big Data'. If we think about the Internet-of-Things, the world-wide increasing number of sensors connected to the Internet generates a lot of these data. For current State-of-the-Art information systems, it is almost impossible to keep up with this massive amount of generated data. Not only storing these data is a problem, also getting valuable informations from inside this 'Big Data' in a adequate time is a huge problem. The great challenge is to find these parts of the data that include useful and valuable informations for people, researchers or companies at the correct time. Getting predictions about earthquakes at the right time using underground microphones can save lives of people. Hybrid architectures are good candidates to rule with this plenty of data. So, our focus is to adapt hybrid architectures for different 'Big Data' applications to get a huge performance gain for these applications.

2 Hybrid Architectures

Many different kinds of hybrid or heterogeneous architectures exists [BDH+10]. A standard approach for such a system uses a General Purpose GPU (GPGPUs), a Field Programmable Gate Array (FPGA) or both connected to an x86 host processor [IBS12]. Hybrid clusters are suitable platforms to achieve a huge performance gain for data-intensive applications [TL10, Awa09]. The Convey HC-1 is a hybrid system including four user-programmable FPGAs connected to a dual-core Intel Xeon 5138 processor via Front Side

*Scholarship holder at Heidelberg Institute for Theoretical Studies (HITS) gGmbH, Schloss-Wolfsbrunnenweg 35, D-69118 Heidelberg . Member of the Data Mining and Uncertainty Quantification Group (DMQ) of Prof. Vincent Heuveline.

Bus (FSB). A memory bandwidth of 76.8 GB/s is provided for an FPGA by eight memory controllers. Using the tools from Convey Computer, a user can (re)configure the included FPGAs for their needs. To reduce the application development time, we designed the concept using so called building blocks [NBK14]. A domain specialist can use these building blocks through an assembler code and benefits from an easily adaptable and programmable high-performance hybrid system. So, the domain specialist is freed of using error-prone hardware description language (HDL).

3 Big Data Applications from the Computational Biology

A lot of problems in biology are solved by computational biology. Bioinformatics solves various kinds of problems using methods from computer science. One major research topic is analyzing protein or genome sequences. For this task, a lot of biological data have to be processed. The tool HHblits compares a query protein sequence against a huge clustered database to find homologous sequences [RBHS12]. Each entry in this database is represented as a Hidden Markov Model (HMM). The alignment between the query protein sequence and this HMM is done by a Viterbi algorithm. This algorithm is very time-consuming, so prefiltering the database using a two-level approach is a good choice to reduce execution time. But the first-level prefilter is still the most time-consuming part in HHblits. We ported this part of the tool [Far07] to the FPGA-based coprocessor (see Figure 1) of the Convey HC-1 and achieve a performance gain for HHblits [BN13, NBSK13].

Through further optimizations, the first-level prefilter on the FPGA-based coprocessor is $3.98\times$ faster than the prefilter on a dual-threaded Intel Xeon 5138 processor using Streaming SIMD Extension (SSE). Exploiting the hybrid system using a task-parallel execution of the entire prefiltering, we further have reduced the execution time. The task-parallel execution is achieved by a hardware-software pipeline. The first-level prefilter is processed on the coprocessor. When the coprocessors starts to deliver valid data, the second-level prefilter is processed by the host. According to our block building convention, the internal coprocessor prefiltering architecture is adaptable by the user without changing VHDL code. Following extensions of our coprocessor architecture shall support more bioinformatics alignment tools like HHblits.

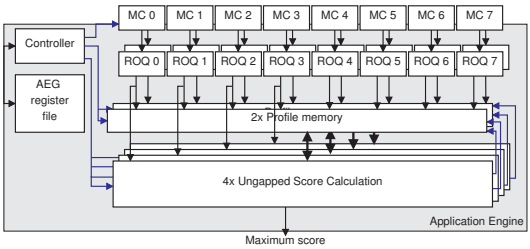


Figure 1: Overview of a fast and highly parallel prefiltering architecture.

4 Supporting Image Processing for Real Time Multimedia Mining

Multimedia mining applications want to draw valuable conclusions from video, image or audio data. Using face recognition, football players can be recognized during a match. So, useful informations about a player can be shown to television viewer. A disparity map calculation can support this face recognition. Therefore, it is important for a live event in television that these data is calculated in real time. Assuming that the input of the calculation are two rectified images, then the disparity is the horizontal difference between corresponding pixels in both pictures. Global or local methods exist for finding such corresponded pixels [SSZ01]. Local methods only consider a small area around each pixel for the correspondence problem. Sum-of-Absolute-Differences (SAD) and Sum-of-Square-Differences (SSD) are well-known local approaches. Global methods are computation intensive but achieving better results than local ones. With dynamic programming (DP) approaches, a good trade-off between computation time and accuracy is achieved. Usually, computation is divided into three stages. A three-dimensional cost matrix is calculated for each scan line in the first stage using SAD or SSD. In the second stage, each element in the cost matrix is aggregated with surrounding elements. Finally, for each scan line a DP matrix is calculated and optimal disparities for a image row is found by backtracking. For suitably porting the most time-consuming part (stage 3) of this algorithm to an FPGA, several problems have to be solved. One problem is that actually only an element at a time can be calculated for the required DP matrix. So, the parallel behavior of a FPGA cannot efficiently be used. Another problem is storing the entire DP matrix for the backtracking step. The Block RAM (BRAM) inside the FPGA is limited, so only a maximum size of a DP matrix is possible. Storing the matrix outside the FPGA causes additional overhead. We are developing an FPGA-based architecture calculating step 3. Currently, we investigate three different methods for finding disparities in one image row. First, backtracking is done inside the FPGA. The backtracking is calculated at the host, so the calculated DP matrix is transfered to host RAM before. Finally, we investigate approaches for finding good disparities without using backtracking. Early investigations have shown, that only finding the minimum entry in a DP column delivers useful disparity values (see Figure 2). There exist very fast and resource saving implementations for determining the minimum value on FPGAs. So, we will transfer the elements of the current calculated column to a reduction circuit determining the minimum value. Therefore, there is no need to store the entire DP matrix.

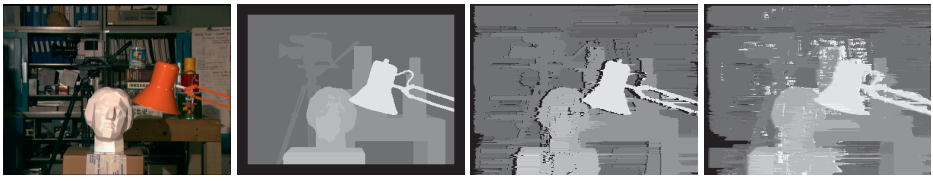


Figure 2: Three disparity maps for the Tsukuba image (Picture 1 from left). Picture 2 is the exact disparity map. Backtracking is used for Picture 3. The right disparity map (Picture 4) is calculated by finding the minimum element of each DP column.

5 Conclusion

The development and efficient utilization of hybrid systems are hot research topics nowadays. Designing a set of building blocks for a designated field of 'Big Data' applications allows domain specialists to easily adapt the different computation units to their needs without special knowledge about programming hybrid systems. So, their applications benefit from different computing resources optimized in terms of speed and energy consumption. Applications like HHblits or image processing task process a huge amount of data. FPGAs in a hybrid system can fast process these data and can inform the host system about data that is suitable for a precise consideration.

References

[Awa09] M. Awad. FPGA supercomputing platforms: A survey. In *International Conference on Field Programmable Logic and Applications (FPL)*, pages 564–568, 2009.

[BDH+10] A. R. Brodtkorb, C. Dyken, T. R. Hagen, J. M. Hjelmervik, and O. O. Storaasli. State-of-the-art in Heterogeneous Computing. *Sci. Program.*, 18(1):1–33, January 2010.

[BN13] M. Bromberger and F. Nowak. Parallel Prefiltering for Accelerating HHblits on the Convey HC-1. In *Mitteilungen*, volume 30, pages 47–57. Gesellschaft für Informatik E.V., Parallel-Algorithmen und Rechnerstrukturen (PARS), September 2013.

[Far07] M. Farrar. Striped Smith-Waterman speeds database searches six times over other SIMD implementations. *Journal of Bioinformatics (Oxford University Press)*, 23(2):156–161, January 2007.

[IBS12] R. Inta, D. J. Bowman, and S. M. Scott. The "Chimera": An Off-the-shelf CPU/GPGPU/FPGA Hybrid Computing Platform. *Int. J. Reconfig. Comput.*, 2012:2:2–2:2, January 2012.

[NBK14] F. Nowak, M. Bromberger, and W. Karl. An Architecture Framework for Porting Applications to FPGAs. In *Accepted for the 11th Workshop on Parallel Systems and Algorithms (PASA), Lübeck, Germany*. Gesellschaft für Informatik E.V., February 2014.

[NBSK13] F. Nowak, M. Bromberger, M. Schindewolf, and W. Karl. Multi-parallel Prefiltering on the Convey HC-1 for Supporting Homology Detection. In *Proceedings of the 20th European MPI Users' Group Meeting*, EuroMPI '13, pages 169–174, New York, NY, USA, September 2013. ACM. International Workshop on Parallelism in Bioinformatics (PBio 2013).

[RBHS12] Remmert, M., Biegert, A., Hauser, A., and Söding, J. HHblits: Lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Journal of Nature methods (Nature Publishing Group)*, 9(2):173–175, February 2012.

[SSZ01] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Stereo and Multi-Baseline Vision, 2001. (SMBV 2001). Proceedings. IEEE Workshop on*, pages 131–140, 2001.

[TL10] K. H. Tsoi and W. Luk. Axel: a heterogeneous cluster with FPGAs and GPUs. In *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*, FPGA '10, pages 115–124, New York, NY, USA, 2010. ACM.

Challenges and Opportunities in Big Data Generation

Christoph Brücke[*], Marie Hoffmann[†], Volker Markl[‡], and Juan Soto[§]

Abstract: The synthetic generation of big data plays a vital role in the development, testing, and evaluation of big data systems. Via the construction of data generator programs (DGPs), we can capture, represent, and regenerate big data sets. DGPs facilitate the transfer of big data sets (e.g., by sharing source codes, in lieu of actually sharing the data), recreate data stored in existing databases, and offer controls to precisely generate data that adheres to user-provided specifications. In this short paper, we will discuss the challenges and opportunities associated with big data generation.

1 Introduction

Big data holds great promise. Various media outlets, such as [MCB+11, MSC13] widely extol the benefits and revolutions that will emerge in the coming years. Currently, there are countless activities (e.g., conferences, trade fairs, and the establishment of new academic programs) taking place worldwide that are all centered on big data. Certainly, surrounding all of the buzz and burgeoning activities are the many technology companies who arduously are working on developing novel solutions to tackle varying engineering challenges, particularly, those arising in each step of the data analysis process. In general, data feeds (e.g., data streams, static data) are gathered, refined (e.g., filtered, extracted, cleansed, and integrated), analyzed (e.g., using machine learning techniques or statistical modeling methods), and then analysts are ultimately informed (e.g., provided with predictive models or visualizations that aid in addressing a problem of interest). However, concerning big data sets practical questions arise, including:

1. How can big data sets (BDS) be efficiently shared?
2. How can we synthetically generate BDS in a scalable manner?
3. How can BDS be shared while preserving privacy or disclosure policies?
4. How can we improve big data system testing and benchmarking?

In this short paper, we would like to raise awareness about the problem of generating big data sets for varying use cases, share our insights concerning problems 1 through 4, and highlight our contributions in the development of a scalable big data generation tool. In section 2, we will present some of the challenges in big data generation and in section 3 their opportunities. In Section 4, we discuss our contributions and ongoing research activities to date. And lastly, in Section 5, we will summarize our main points.

2 Challenges

During our research in the field of big data generation, we identified multiple challenges to be addressed. Some major challenges are listed below.

One major challenge is ***scalable data generation***. With the advent of next generation computing platforms such as Hadoop[1] and Stratosphere[2] there is also a need for next generation data

[*] christoph.bruecke@campus.tu-berlin.de [†] marie.hoffmann@tu-berlin.de [‡] volker.markl@tu-berlin.de
[§] juan.soto@tu-berlin.de
[1] http://hadoop.apache.org/ [2] http://stratosphere.eu

generators. Traditionally, data generators were programmed in a sequential fashion, which makes it nearly impossible to scale them out in a shared nothing environment. Thus, a challenge in big data generation is firstly the ability to generate data that is big enough and secondly to design data generators that are capable of leveraging the existing computing platforms and scale up to hundreds of machines.

Another challenge involves ***capturing and representing existing database schemas and catalogs***. In particular, ensuring that DGPs satisfy varying data model constraints upon execution. These constraints can be divided into two disjoint categories: *hard constraints*, i.e., constraints that must not be violated, such as type, uniqueness, referential integrity, or check constraints, and *soft constraints* that only need to be satisfied approximately. Numerous researchers are actively investigating constraint languages to describe data generators, most notably Arasu et al. [AKL11] and E. Torlak [Tor12]. Their approaches are either focused on soft constraints or on hard constraints. Now, the challenge is to develop a hybrid approach that captures both soft and hard constraints jointly.

Yet another challenge lies in the ability for data owners to ***share*** their ***data with third parties***. Unfortunately, today's networking technologies are limited in their ability to efficiently transfer big data sets (e.g., petabytes, exabytes) quickly across a set of data subscribers. Hence, costly and slow workarounds are used, such as shipping physical hard drives to data subscribers. Another issue regarding transferring data sets to third parties is data cleansing and anonymization.

3 Opportunities

The strategies of data storage systems that have to handle large amounts of data differ from traditional approaches due to the fact that the growth of computational speed cannot keep up with the growth of data. Solving the afore mentioned challenges gives new opportunities for different development phases.

One opportunity for DGPs lies in the field of ***software testing and debugging***. Hardware components in large scale data systems may be swapped and software components are further developed. To ensure the full range of functions, a modified system undergoes regression testing before being made available to customers. Components that directly work on the data such as storage managers or data analysis programs may behave differently depending on data quantity and quality. Hence, debugging erroneous systems involves using data that resembles the original one in size and probability distribution, but is not necessarily identical to the original one.

As of today there is a great need for comprehensive ***big data benchmarks***. Thus it is a major opportunity for DGPs, since a key aspect of such benchmarks is the data. Current data sets, like TPC-H or TPC-DS[3] are characterized by fixed schemas and fixed column distributions. Only the scaling factor is user-defined. Benchmarks are typically shipped with data-generating scripts that can not be executed in parallel and thus are limited to data sizes in the scale of gigabytes. The opportunity here is to provide the user with the means and frameworks to produce industry standard benchmark datasets that scale.

Transferring data in the range of tera- or petabytes over a network is still not feasible despite

[3] Transaction Processing Performance Council. http://www.tpc.org/

optical fiber cables and mechanisms for error correction. Hence, another opportunity is ***data transfer***. This can be resolved by solely capturing relevant data characteristics such as distribution properties or correlations, and then summarizing them in an executable binary, which can be distributed to generate the data on location.

4 DIMA Group Contributions

Members of the DIMA Group at TU Berlin have made numerous contributions in the field of big data generation over the past few years [ABM13]. Among them are the Myriad Toolkit [ASP+11, ATM12] with the accompanying Oligos component.

DIMA researchers have been actively working on an open source scalable parallel data generation toolkit written in C++ called Myriad[4] that is capable of generating arbitrarily large synthetic data sets conforming to user prescribed specifications. The data specification is given by an XML file which can be generated automatically by other tools, such as Oligos, or manually by the user. The XML input specification is compiled into a program that contains an execution plan and generators for setting fields. The binary facilitates inter- and intra-node parallelism, i.e., locally distributed execution and multithreading. Parallel data generation is made possible by employing pseudo-random number generators.

Oligos[5] complements the Myriad toolkit. For increased usability, Oligos, a Java based software tool, was developed to ease capturing existing database schemas and transforming them into XML for use by Myriad. The architecture of both Myriad and Oligos is illustrated by figure 1 below.

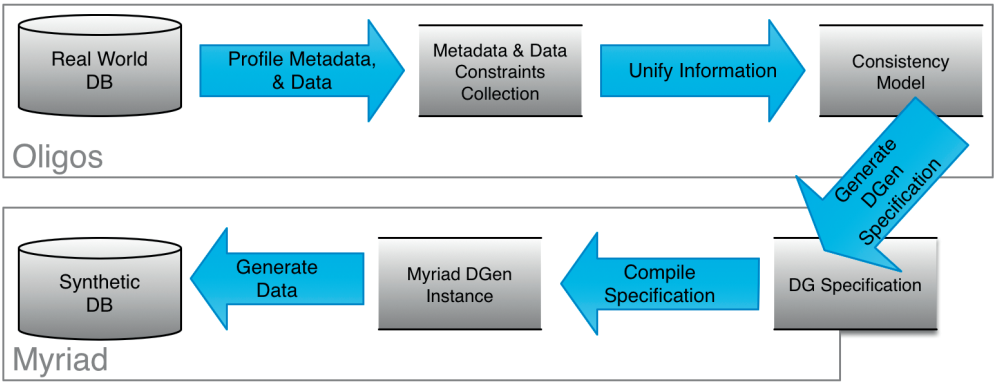


Figure 1: Synthetic Data Generation Process

As part of the big data generation problem we investigate the problems of composite key generation and data description by giving constraints.
Composite keys are keys whose single attributes may not fulfill the uniqueness constraint, only when combined. Generating composite keys in parallel without re-processing previously gener-

[4] http://myriad-toolkit.com/ [5] https://bitbucket.org/carabolic/oligos/

ated keys is not straight-forward, but can be solved by sharing generators that provide pseudo-random permutations.
Data can be characterized by giving a set of constraints of the form $P(< event > \mid < pre\_cond >) = p$. In general, it is not efficiently computable whether there exists a solution for a particular set [KM93]. Our goal is determine conditions under which there exists a solution and how the solution space can be sampled by a parallel data generation framework like Myriad.

5 Conclusion

In this short paper we gave an overview of the challenges and opportunities in big data generation as we see them. We have already laid out a solid foundation with Myriad and Oligos to tackle some of the challenges. Our contributions are given by the Myriad runtime, the data generator specification language based on XML, and the Oligos component. The Myriad Toolkit generally helps users to *write* scalable custom data generators, whereas the data generator specification language and Oligos helps to *design* custom data generators. Indeed, "Big (Data) is Beautiful" and we want to help to shape the future of big data generation. For further inquiries, please contact us at myriad.toolkit@dima.tu-berlin.de.

References

[ABM13] Alexander Alexandrov, Christoph Brücke, and Volker Markl. Issues in big data testing and benchmarking. In *Proceedings of the Sixth International Workshop on Testing Database Systems*, page 1. ACM, 2013.

[AKL11] Arvind Arasu, Raghav Kaushik, and Jian Li. Data generation using declarative constraints. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, SIGMOD '11, pages 685–696, New York, NY, USA, 2011. ACM.

[ASP+11] Alexander Alexandrov, Berni Schiefer, John Poelman, Stephan Ewen, Thomas O Bodner, and Volker Markl. Myriad: parallel data generation on shared-nothing architectures. In *Proceedings of the 1st Workshop on Architectures and Systems for Big Data*, pages 30–33. ACM, 2011.

[ATM12] Alexander Alexandrov, Kostas Tzoumas, and Volker Markl. Myriad: scalable and expressive data generation. *Proceedings of the VLDB Endowment*, 5(12):1890–1893, 2012.

[KM93] Daphne Koller and Nimrod Megiddo. Constructing Small Sample Spaces Satisfying Given Constraints. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 268–277, New York, NY, USA, 1993. ACM.

[MCB+11] James Manyika, Michael Chui, Brad Brown, Jaques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hunf Byers. Big data: The next frontier for innovation, competition, and productivity. May 2011. *MacKinsey Global Institute*, 2011.

[MSC13] Viktor Mayer-Schönberger and Kenneth Cukier. *Big Data: A Revolution that Will Transform how We Live, Work, and Think*. Houghton Mifflin Harcourt, 2013.

[Tor12] Emina Torlak. Scalable test data generation from multidimensional models. In *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, page 36. ACM, 2012.

Scaling in a Distributed Spatial Cache Overlay*

Alexander Gessler, Simon Hanna, Ashley Marie Smith
Universität Stuttgart, Institute for Parallel and Distributed Systems
{gessleah, hannasn, smithae} @studi.informatik.uni-stuttgart.de

Abstract: Location-based services for mobile devices are a type of distributed system that utilizes geographic behavior of its users. Balancing dynamic query workloads and skewed data remains a problem. Scale-in and scale-out are two proposals that temporarily remove or add resources, respectively. To characterize situations where scaling is more efficient, we implemented a distributed spatial cache overlay for 2D data with the goal of evaluating system performance with and without scaling-out. In this paper, we present an experimental setup to benchmark such a system, and measurements of relative scalability under different cache overlay sizes, query rates and workload distributions. Our results indicate that the system achieves almost linear relative scalability for both uniform and non-uniform query distributions.

1 Introduction and Foundations

Location-based services (LBS) are data-intensive applications which use the geographic behavior of the user in order to process queries. These queries access spatial data, which correspond to physical geographic regions. A typical example is a route-planning application for smartphones. The user sends an address as a query, and in response to the query, the application delivers data, such as the corresponding portion of the map. A fundamental problem of LBS is how to allocate workload in the system, so as to guarantee low latency and efficiency in processing these queries.

In an LBS, both query workloads and data have spatial and temporal aspects, which dynamically change and thus require special considerations. Query loads can vary depending on time or geographic density of users. For example, fewer queries are sent late at night or in rural areas. Sometimes many queries request data from one location, for instance when large crowds gather for an event. Moreover, the distribution of data can be skewed, i.e. non-uniformly clustered around certain spatial regions, such as big cities.

Given these spatial and temporal characteristics, we are interested in minimizing decreases in system performance under such loads or data distribution. Much research has focused on load-balancing mechanisms for distributed systems with spatial data. A common technique is to partition data and then build a network overlay, which dedicates nodes to handle requests for certain partitions [3]. Other approaches estimate query loads or calculate weights to place more nodes around expected hotspots [1]. Yet it is difficult to predict or

*Based on a student project supervised by Carlos Lübbe at IPVS, University of Stuttgart

respond to changes, such as moving hotspots. A more effective approach dedicates nodes in the overlay to cache frequently accessed data; this network is referred to as a *distributed spatial cache overlay* [2].

However, none of the previously mentioned load-balancing approaches in distributed spatial overlays can increase throughput beyond what is already the system maximum. Thus, extreme spikes in query workloads can exceed the system's maximum capabilities. Adding and then removing additional resources to the LBS can address these temporary spikes. The idea is that the overlay automatically decides when to add or remove nodes, based on self-measured load. Removing nodes is referred to as *scaling-in*, whereas adding nodes is referred to as *scaling-out.* In the scope of our project, we investigate whether scaling-out mitigates dynamic peaks in a distributed spatial cache overlay.

2 Architecture and Methodology

Our system is inspired by the concepts presented in [2]. The data region is partitioned into a 2D grid. Then a distributed cache overlay is built on top which consists of nodes dedicated to caching data from certain grid partitions. Our grid of cache nodes forms a Delaunay triangulation in a 2D metric space. A cache node's coordinate is known as its *cache focus*, which is important when load-balancing or caching data. Greedy routing is used to forward queries to a target node; the desired data are within a specified distance to that node's cache focus. Based on these general principles, we implemented a scaling-out mechanism for the distributed spatial cache overlay in a cluster environment.

Our system architecture, as shown in Figure 1, consists of a grid of cache nodes that are organized spatially and deployed on a cluster. Nodes receive external queries, which they can forward. An external GUI can be used to launch and shut the system down. The system runs remotely on a cluster of variable size and can test the grid under different load-balancing and scaling mechanisms. Furthermore, we included an administrative node to initialize overlay construction and a basic logging system. Our implementation of scaling-out relies on asynchronous message-passing and handshake confirmation between nodes. The idea is that when a node's load exceeds a given threshold, the node communicates with its neighbors to confirm whether the overload can be relieved by adding a new node.

The goal of our experiment was to quantify latency under uniform and non-uniform query loads. In the uniform case, the target coordinates of the queries are distributed uniformly. However in the non-uniform case, we can simulate the formation of hotspots: the target coordinate is obtained by sampling a Gaussian distribution with a standard deviation of 0.18 times the area of the data region that is centered around a point in the grid. We measured request latency of the cache overlay under three experimental groups [1] , depending on the type of distribution and presence of scaling-out: (1) uniform plus scaling-out, (2) hotspot plus scaling-out , (3) hotspot.

The distributed spatial cache overlay was launched on a cluster of up to 32 (virtual) ma-

[1] We briefly examined the effect of uniform distributions without scaling-out. The latency was, as expected, worse than uniform with scaling, but better than our hotspot tests.

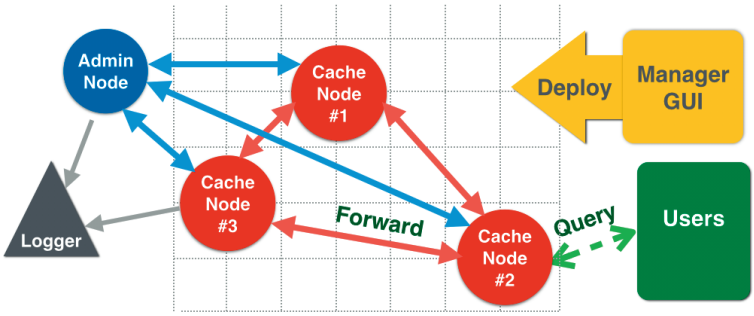


Figure 1: System Architecture

chines. Our benchmarking program sent each cache node k queries per second. This query rate corresponds to about $1000/k$ milliseconds between two successive queries. For every experimental group, we ran the benchmark program with query rates of $k = 15\ \mathrm{s}^{-1}$ and $k = 20\ \mathrm{s}^{-1}$ and a given number of nodes ($n = 8, 12, 16$). [2] Each run lasted 10 seconds, so the total number of queries sent to the grid during each run was $10 \times n \times k$.

3 Results and Discussion

For a given grid size n and query rate k, Figures 2 and 3 show the median latencies in milliseconds of all queries, according to our experimental groups.

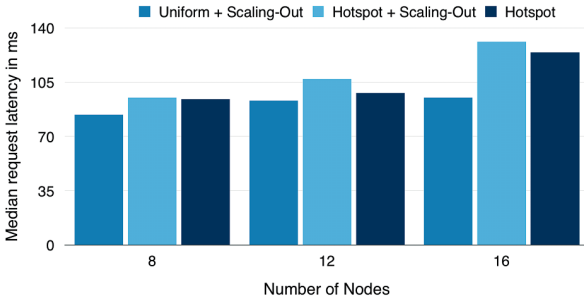


Figure 2: Median Request Latency in ms, k=15

We hypothesized that with scaling-out, our system could approximate linear scalability in both the uniform and non-uniform cases. Our results support our hypothesis. Under the higher request rate $k = 20$, latency increases markedly for the hotspot group. In contrast, the median times for the hotspot (with scaling-out) group when $n = 16$ remain within 30%

[2] The remaining (virtual) machines are used during scaling.

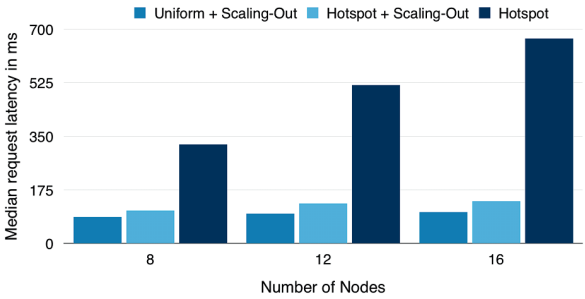


Figure 3: Median Request Latency ms, k=20

of the median latencies when $n = 8$, although the total number of nodes and therefore the amount of queries being routed to the hotspot has doubled.

4 Conclusion and Future Work

In our study, we designed, implemented and evaluated a distributed spatial cache overlay that is flexible enough to allow implementations of sophisticated load-balancing schemes. This cache overlay serves as the basis for further comparisons of various load-balancing mechanisms to see whether linear scalability can be achieved. We confirmed that non-uniform workloads cause the system's performance and scalability to degrade under higher query rates. Using our scaling-out mechanism, we were able to scale the system's performance almost linearly in a cluster environment with sufficient bandwidth.

Based on our current results, our primary focus will be to implement scaling-in load-balancing to handle changes in both global and relative load. Since our current implementation exhibits a routing time of $\mathcal{O}(\sqrt{n})$, we propose that our nodes keep skiplist-like connections to 2nd or 3rd neighbors as well as local routing shortlists, in order to achieve $\mathcal{O}(\log n)$ routing. With local routing shortlists, cache nodes track recent queries which entered the overlay through them, along with the cache foci of all the nodes the query visited. Nodes could then improve the first routing decision of subsequent queries that enter.

References

[1] T. Scholl, et al. "Workload-Aware Data Partitioning in Community-Driven Data Grids." Extending DataBase Technology (*EDBT*): ACM, 2009.

[2] C. Lübbe, et al. "Holistic load-balancing in a distributed spatial cache." Mobile Data Management (*MDM*): IEEE Computer Society, 2013: pp. 267-270.

[3] S. Wee and H. Liu. "Client-side load balancer using cloud." Symposium on Applied Computing (*SAC*): IEEE Computer Society, 2010, pp. 399-405.

Prozess-Optimierung durch zentralisierte Datenbanklösung und automatisiertes Lean-Reporting

Benjamin Gollmer

Hochschule Albstadt-Sigmaringen
Fakultät Business and Computer Science
benjamingollmer@yahoo.de

Art der Arbeit: Bachelor-Arbeit
Betreuer der Arbeit: Prof. Dr. Walter Hower

Abstract: In der heutigen Zeit werden in Unternehmen immer mehr Informationen gespeichert. Diese enormen Datenmengen lassen sich meist nur mit hohem Aufwand maschinell verarbeiten und werden daher in der Regel in mühsamer Handarbeit aufgearbeitet. Durch diese Vorgehensweise verlieren die betroffenen Mitarbeitern wertvolle Zeit. Diese Problematik fand ich im Rahmen meiner Bachelorarbeit bei einem renommierten Automobilhersteller vor. Beinahe jede Herausforderung, welche meinerseits für die Optimierung eines Hauptprozesses ermittelt wurde, konnten auf die oben beschriebene Problematik zurückgeführt werden, also galt es eine Lösung zu finden. Zu Beginn der Arbeit wurden alle Vorgänge und Abläufe des Prozesses, sowie alle benötigten Tools analysiert. Es wurde meinerseits ein Soll-Konzept erarbeitet, welches teils noch im Rahmen meiner Arbeit operativ umgesetzt wurde und teils erst zukünftig realisiert wird.

1 Einführung

Die aufkommenden Datenmengen in Unternehmen wachsen stetig. Ermittlung, Speicherung und Verarbeitung von Informationen werden dadurch zunehmend erschwert.

Diese Erkenntnis erlangte ich im Rahmen meiner Bachelorarbeit, welche die Optimierung eines Entscheidungs-Prozesses eines renommierten Automobilherstellers zum Ziel hatte. Der betrachtete Prozess wird monatlich durchlaufen und generiert dabei ein enormes Datenaufkommen, welches sich durch eine nicht optimale redundante Erfassung und Verarbeitung der Daten nochmals steigert. Diese Redundanz und die Tatsache, dass in der Regel alle Daten per Hand erfasst werden, ist der Grundstein der Herausforderungen, welche einen reibungslosen Ablauf des Prozesses verhindern.

Einer der wichtigsten Faktoren des Prozesses ist die Zeit. Innerhalb des Prozesses gibt es einige Sub-Prozesse, welche zu definierten Terminen abgeschlossen werden müssen, was in der Regel nicht möglich ist, da die betroffenen Mitarbeiter zu viel der zur Verfügung stehenden Zeit für die Erfassung und Verarbeitung gesammelter Daten aufwenden müssen. Zurückzuführen ist dies auf die oben erwähnte Redundanz. Daten werden stets an *n* Stellen erfasst und müssen daher auch an *n* Stellen aktualisiert werden, was in der Praxis mehrmals täglich vorkommt.

Durch die Datenerfassung per Hand besteht ein großes Fehlerpotential, das den stetigen Anstieg der Datenmengen immer mehr steigert. Daher wurden Optimierungspotentiale und -maßnahmen erarbeitet um zukünftig das aufkommende Datenvolumen besser beherrschen zu können und das Berichtswesen des Prozesses schlanker und effektiver zu gestalten.

Die ermittelten Daten wurden ohne systematische Ablage auf allen Abteilungslaufwerken, in diversen Dateien, verstreut abgelegt. Diese Vorgehensweise birgt die Gefahr, dass Daten verloren gehen oder in Vergessenheit geraten.

2 Motivation

Die Erfassung und Verarbeitung von redundanten Daten ist nicht praktikabel und schafft unnötige Probleme. Der Prozess-Ablauf wird gestört, die Qualität der Arbeit leidet und die betroffenen Mitarbeiter verlieren ihre Motivation.

Dies ist für ein Unternehmen nicht wünschenswert und birgt einige Gefahren, daher muss an dieser Situation etwas geändert werden. In der heutigen Zeit gibt es Mittel und Wege die Arbeitsweise effizienter und für die Mitarbeiter angenehmer zu gestalten. Also stellt sich die Frage, warum werden diese Potentiale nicht genutzt?

Durch die Optimierung der Arbeitsweise der betroffenen Mitarbeiter wird die Effizienz gesteigert, wodurch eine Senkung des Termin- und Zeitdrucks erfolgt.

Meinerseits wurden Optimierungspotentiale erarbeitet welche als Hauptziel, die Abschaffung der manuellen, redundanten Datenerfassung und die Einführung einer systematischen Datenablage in einer Datenbank, im Fokus hatten. Dies ist bei den betroffenen Daten enorm wichtig, da es sich um äußerst wichtige Informationen wie Entwicklungskosten, Terminpläne, Stücklisten und Ähnliches für angestrebte Fahrzeugprojekte handelt, welche den Entscheidungsträgern als Basis dienen.

Die Einführung einer systematischen und zentralisierten Datenerfassung und –ablage löst in diesem speziellen Fall eine unstrukturiete Erfassung und Verbreitung der Daten ab. Als Beispiel wäre hier zu nennen, dass zu dem Zeitpunkt der Erstellung dieser Arbeit die Daten der Fahrzeugstücklisten von verschiedensten Mitarbeitern an verschiedensten Ablageorten erfasst und auf verschiedensten Wegen wie per Telefon, per Mail und mündlich verbreitet wurden. Dadurch war es für die zuständigen Mitarbeiter der zantralen Planung beinahe unmöglich die Daten in einen sinnigen Zusammenhang zu bringen und des öfteren verschwanden wichtige Informationen wodurch wichtige Entscheidungen anhand einer inkorrekten Basis erfolgten.

3 Lösungsansatz

Zukünftig sollte eine integrierte Datenbanklösung eingesetzt werden, die sich nahtlos in die bestehende Systemlandschaft des Unternehmens einbinden lässt. Durch diese Lösung könnten alle manuell zu befüllenden Dokumente des Prozess, durch Eingabemasken einer Datenbank ersetzt werden. Die Machbarkeit wurde überprüft und stellt kein Problem dar. Dadurch würde die unüberblickbare Fülle an Dokumenten und Makro-Tools entfallen und die betroffenen Mitarbeiter entlastet. Als weiterer Vorteil der Datenbanklösung ist zu nennen, dass alle prozessrelevanten Daten an zentraler Stelle einmalig gespeichert und somit an allen benötigten Stellen zur Nutzung zur Verfügung stehen; dadurch werden anfallende Aktualisierungen ebenfalls nur an einer Stelle vorgenommen.

Ebenso besteht die Möglichkeit, direkt aus der Datenbank wichtige Dokumente wie Übersichten, Auswertungen, Projektüberleitungen oder Berichte automatisch generieren zu lassen. Aktuell müssen solche Dokumente in mühsamer Handarbeit erstellt werden, was wiederum einen Großteil der verfügbaren Zeit in Anspruch nimmt.

Weiter ist zu nennen, dass eventuell auch abteilungsübergreifende Freigaben der Datenbank erfolgen könnten, um die Informationsqualität und -dichte, durch abteilungsübergreifenden Input, zu verbessern.

Neben den genannten Vorteilen der Datenbanklösung ist der wichtigste Punkt die systematische Ablage der ermittelten Daten. Vor der Optimierung wurden die Daten unstrukturiert auf diverse Netzlaufwerke verteilt abgelegt und gerieten in Vergessenheit. In einer strukturierten Datenbank sind alle Daten jederzeit wieder auffindbar, was einen wirklichen Mehrwert generiert. Dadurch können die Daten vergangener Prozesse eingesehen und ausgewertet werden. Dies macht Sinn, da im Rahmen des zu optimierenden Entscheidungs-Prozesses des Öfteren ähnliche oder praktisch gleiche Themen behandelt werden; so könnten wichtige Referenzdaten vergangener Prozesse mit aktuellen Prozessen verglichen oder übergeleitet werden.

Nicht aufgeführte Details sind der Geheimhaltung seitens des Unternehmens geschuldet.

4 Ausblick

Die im laufe der Bachelor-Arbeit operativ umgesetzten Optimierungspotentiale werden beibehalten und in die bestehende Systemlandschaft integriert.

Aktuell wird auf Basis der Erkenntnisse der Arbeit eine Integration weiterer Datenquellen in die zentralisierte Lösung anhand eines Prototypes getestet und bei Erfolgreichem Abschluss der Tests professionell umgesetzt.

Literaturverzeichnis

Bär, Reinhard; Purtschert, Philippe: Lean-Reporting: Optimierung der Effizienz im Berichtswesen;
Springer, 2013 (Hrsg.: Prof. Dr. Walter Hower)

Moormann, Jürgen: Lean Reporting und Führungsinformationssysteme bei deutschen Finanzdienstleistern; Hochschule für Bankwirtschaft, 1995

Lean Management in der Praxis; Zürich; Verl.: industrielle Organisation, 1993

Diverse unternehmensinterne Quellen ohne bestimmte Bezeichnung

Ergebnisse Mitarbeiter-Interviews

ruleDQ: Ein Regelsystem zur Datenqualitätsverbesserung medizinischer Informationssysteme

Mihael Gorupec und Gregor Endler
Lehrstuhl für Informatik 6 (Datenmanagement)
Friedrich-Alexander-Universität Erlangen-Nürnberg
m.gorupec@gmail.com

Abstract: ruleDQ realisiert ein Regelsystem zur automatisierten Messung und Verbesserung von Datenqualität. Anwendern ohne informatisches Fachwissen wird es ermöglicht, eigenständig Datenqualitätsregeln aufzustellen. Die Regeln werden von ruleDQ regelmäßig ausgewertet und erlauben eine Quantifizierung und Analyse von Datenqualitätsmängeln. ruleDQ folgt dabei den Prinzipien eines kontinuierlichen Datenqualitätsmanagements um nachhaltige Verbesserungen zu ermöglichen.

1 Einführung und Motivation

Im Gesundheitswesen besteht der gegenwärtige Trend des Zusammenschließens mehrerer individueller Praxen zu medizinischen Versorgungszentren [HAE09]. Dies hat die Erhöhung der Konkurrenzfähigkeit zum Ziel, die Versorgungszentren können durch die Zusammenlegung von Ressourcen unter anderem Kosten einsparen und Kunden einen vielfältigeren Service anbieten [EBL13].

Diese Zusammenschlüsse führen zu der Notwendigkeit einer zentralen administrativen Instanz, den Praxismanagern. Praxismanager sind für die Ressourcenplanung und Unternehmenssteuerung verantwortlich. Außer an das Personal, ergeben sich auch neue Herausforderungen an die informationstechnologische Infrastruktur. Praxismanager benötigen für ihre Aufgaben einen einheitlichen Blick auf alle Daten der einzelnen Verbundspartner. Durch die Zusammenführung von heterogenen Daten aus unterschiedlichen Quellen ergeben sich aber Herausforderungen bei der Einhaltung von gemeinsamen Datenqualitätsstandards.

Erschwerend kommt hinzu, dass es in der informationstechnologischen Landschaft von medizinischen Versorgungszentren zu ständigen Änderungen kommt. Durch die Hinzunahme von neuen Mitgliedern, durch Änderungen von Budgets und Verträgen oder durch neue Gesetzgebung müssen Datenqualitätsanforderungen ständig neu evaluiert und angepasst werden [End12].

2 Anforderungen und Ziele

Das Ziel dieser Arbeit war es, ein Konzept zur Messung und Verbesserung von Datenqualität zu schaffen und mit Hilfe eines Regelsystems zu realisieren.

Die späteren Anwender sind Domänenexperten, die in der Lage sind die vorhandenen Daten zu interpretieren und Regeln zu erkennen. Allerdings mangelt es an informatischem Fachwissen, um die gefundenen Regeln etwa in Form von SQL umzusetzen. Die Lösung muss es Anwendern daher erlauben Regeln auf möglichst einfache und leicht verständliche Weise zu formulieren.

Zur Umsetzung dieser Anforderungen wurde die Verwendung von regelbasierten Systemen als geeignet gesehen. Regelbasierte Systeme sind Applikationen, die Problemlösungs-Know-How automatisieren, indem sie Expertenwissen mit Hilfe von Regeln abbilden. Regeln bestehen in regelbasierten Systemen aus Wenn-Dann-Sätzen. Dies beruht darauf, dass menschliche Experten ihre Problemlösungs-Techniken für gewöhnlich in Form von Situations-Aktions-Regeln ausdrücken. Der Wenn-Teil einer Regel besteht aus einer Bedingung, welche definiert wann die Regel ausgelöst wird. Der Dann-Teil beschreibt die Aktion, welche ausgelöst wird wenn die Bedingung der Regel erfüllt ist [HR85].

Die geschaffene Lösung musste zudem den Prinzipien eines kontinuierlichen Datenqualitätsmanagements folgen, welches für eine nachhaltige und effektive Datenqualitätssteigerung nötig ist. Punktuelle Datenreinigungen haben nur einen kurzfristigen Effekt, die dadurch erzielten Verbesserungen gehen gerade bei sich häufig ändernden Daten schnell verloren [Red97]. Das Thema Datenqualität darf daher nicht als eine einmalige Aktion betrachtet werden. Um Datenqualität effektiv zu verbessern, bedarf es ganzheitlicher Methoden, die Daten über ihren gesamten Lebenszyklus hinweg betrachten um ein definiertes Niveau an Qualität zu garantieren [BCFM09].

3 Ergebnis

Eine im Zuge der Arbeit durchgeführte Analyse von Referenzunternehmen zeigte, dass keine vollständig automatisierten Methoden zur Messung von Datenqualität oder Behandlung von Mängeln in den Versorgungszentren existierten. Zudem gaben Verantwortliche an, häufig unter verschiedenen Datenqualitätsproblemen zu leiden. Daher wurde im Rahmen dieser Arbeit prototypisch eine Applikation zur automatisierten Messung und Verbesserung von Datenqualität entworfen und implementiert (*ruleDQ*). Die Applikation erlaubt es den Anwendern eigenständig einfache boolesche Regeln für ihre Daten zu formulieren. Die Regeln werden von der Applikation anschließend in SQL übersetzt und kontinuierlich ausgewertet. Als Ergebnis dieser Evaluierung von Regeln stehen quantitative Messwerte, welche Rückschlüsse über das Niveau von Datenqualität ziehen lassen. Die Applikation identifiziert regelverletzende Datensätze und erlaubt dem Nutzer so die Analyse der aufgetretenen Mängel.

ruleDQ wurde dabei nach dem Vorbild eines regelbasierten Systems konzipiert. Regelbasierte Systeme fordern die Trennung von Regeln, Daten und Prozessen. Geschäftsregeln

werden unabhängig vom Ausführungscode formuliert und verwaltet. Nach [Los02] besteht ein regelbasiertes System aus folgenden drei Modulen:

1. Einem Benutzerinterface zur Erstellung und Verwaltung von Regeln
2. Einer persistenten Regelbasis
3. Einem Modul zur Ausführung von Regeln

ruleDQ folgt diesem Aufbau und bietet zusätzlich dazu ein Modul zur Analyse der Ergebnisse der Regelauswertung. Das theoretische Konzept hinter ruleDQ lehnt sich an das von Wang geschaffene Total Data Quality Management (TDQM) an [WZL01]. TDQM basiert auf der Betrachtung von Daten ähnlich zu Produkten in der Fertigungsindustrie. Die in diesem Bereich weitverbreitete Qualitätssicherungs-Methodik des Demingkreises [Dem86] wurde dabei auf Daten übertragen. Der Demingkreis beschreibt eine iterative Problemlösungsmethodik die aus vier Phasen besteht: Planen, Umsetzen, Überprüfen, Handeln. TDQM definiert analog zum Demingkreis vier Phasen welche kontinuierlich durchlaufen werden müssen, um eine nachhaltige Datenqualitätsverbesserung zu erreichen. Die Phasen des TDQM gliedern sich in Definition, Messung, Analyse und Verbesserung.

In der Definitions-Phase müssen relevante Datenqualitätsdimensionen ausgewählt und entsprechende Anforderungen an diese definiert werden. In ruleDQ geschieht dies in Form von Geschäftsregeln. Will ein Anwender zum Beispiel sicherstellen, dass bestimmte Medikamentengruppen bei einer Diagnose nicht verschrieben werden, so kann er mit Hilfe von ruleDQ zu diesem Zweck eine Regel anlegen, etwa in der Form `(Diagnose = 'R50.80') AND (Medikation != 'Placebo')`.

Nach der Formulierung der Anforderungen folgt deren Anwendung auf die Daten. Es wird untersucht inwiefern die Daten den Anforderungen genügen. Dazu dienen im Falle von ruleDQ quantitative Metriken. Die vorher formulierten Regeln werden in ein SQL-Statement geparst und auf Daten in einer relationalen Datenbank angewandt. Ein mögliches Ergebnis beim obigen Beispiel wäre etwa, dass 120 von 2400 Tupeln in der Datenbank die Regel verletzten und somit bei 5% der Diagnosen unerwünschte Medikamentengruppen verschrieben werden.

In der Analyse-Phase werden Ursachen mangelnder Datenqualität festgestellt. Die Ursachen können vielfältig sein, unter anderem können fehlerhaften Anwendungen, menschliche Fehler oder schlecht gestaltete Prozesse verantwortlich sein. Zur Unterstützung in dieser Phase können mit ruleDQ regelverletzende Datensätze analysiert werden. Der Anwender kann also prüfen in welchen Fällen die Regel verletzt wurde und in obenstehendem Beispiel welche Ärzte die unerwünschten Medikamente benutzt haben.

In der Verbesserungs-Phase sollen die Fehler und deren Ursachen nachhaltig beseitigt werden. Dazu gilt es permanent qualitätssichernde Maßnahmen zu implementieren, wie etwa durch Prozess-Redesign. Einmalige Datensäuberungsmaßnahmen können davor initial zur Anwendung kommen. ruleDQ bietet für diese Phase die Versendung von Warnungen bei einem Absinken unter ein spezifiziertes Datenqualitätsniveau an. So kann automatisiert eine E-Mail als Warnung verschickt werden, wenn z.B. bei mehr als 1% der Diagnosen die

unerwünschte Medikamentengruppe benutzt wird. Nach Abschluss aller Phasen erfolgen stets eine neue Iteration und die erneute Definition von Anforderungen [BCFM09].

4 Fazit

Der Kontext dieser Arbeit liegt in der medizinischen Domäne. Da ruleDQ jedoch weder Annahmen über das Schema der Daten noch über deren Semantik trifft, kann es auch in anderen Domänen eingesetzt werden. ruleDQ bietet den Anwendern in allen Phasen des TDQM Unterstützung und trägt so zu einem kontinuierlichen Datenqualitätsmanagement bei. Durch die fortlaufende Überwachung wird ein erneutes unbemerktes Absinken der Qualität verhindert und eine nachhaltige Verbesserung der Datenqualität ermöglicht. Zukünftige Verbesserungen sind z.B. mit der Erweiterung von ruleDQ um Kontextsensitivität zu erreichen, um etwa die möglichen Optionen zur Regelformulierung in Abhängigkeit des Datentyps eines ausgewählten Feldes zu beschränken oder zu erweitern.

Literatur

[BCFM09] Carlo Batini, Cinzia Cappiello, Chiara Francalanci und Andrea Maurino. Methodologies for data quality assessment and improvement. *ACM Comput. Surv.*, 41(3):16:1–16:52, Juli 2009.

[Dem86] W Edwards Deming. Out of the crisis. Cambridge, MA: Massachusetts Institute of Technology. *Center for Advanced Engineering Study*, Seite 6, 1986.

[EBL13] Gregor Endler, Philipp Baumgärtel und Richard Lenz. Pay-as-you-go data quality improvement for medical centers. In E. Ammenwerth, A. Hörbst, D. Hayn und G. Schreier, Hrsg., *Proceedings of the eHealth2013*, Seiten 13–18, 2013.

[End12] Gregor Endler. Data quality and integration in collaborative environments. In SIGMOD/PODS und ACM, Hrsg., *SIGMOD/PODS 2012 PhD Symposium*, New York, NY, USA, 2012.

[HAE09] W. Hellmann, T. Antwerpes und S. Eble. *Gesundheitsnetzwerke managen: Kooperationen erfolgreich steuern*. MWV Medizinisch Wiss. Verlag-Ges., 2009.

[HR85] Frederick Hayes-Roth. Rule-based systems. *Commun. ACM*, 28(9):921–932, September 1985.

[Los02] David Loshin. Rule-based data quality. In *Proceedings of the eleventh international conference on Information and knowledge management*, CIKM '02, Seiten 614–616, New York, NY, USA, 2002. ACM.

[Red97] Thomas C. Redman. *Data Quality for the Information Age*. Artech House, Inc., Norwood, MA, USA, 1st. Auflage, 1997.

[WZL01] Richard Y. Wang, Mostapha Ziad und Yang W. Lee. *Data Quality*, Jgg. 23 of *Advances in Database Systems*. Kluwer, 2001.

Big Data – Big Challenges – Big Chances: Datenmanagement in den Informatikunterricht!

Andreas Grillenberger

Friedrich-Alexander-Universität Erlangen-Nürnberg
Didaktik der Informatik, Martensstr. 5a, 91058 Erlangen
andreas.grillenberger@fau.de

Abstract: Die strukturierte Speicherung von Informationen stellt ein zentrales Thema in der Informatik und damit auch im Informatikunterricht dar. Durch die zunehmende Bedeutung von Big Data findet in diesem Bereich zurzeit ein Paradigmenwechsel statt, der sich gleichzeitig auch auf den Alltag auswirkt. Sowohl die fachlichen als auch die gesellschaftlichen Aspekte von Big Data haben dabei starken Einfluss auf den Unterricht. Insbesondere müssen die Inhalte im allgemeinbildenden Schulunterricht verschiedenen Kriterien genügen, durch Innovationen wie NoSQL-Datenbanken werden jedoch Konzepte, die bisher als grundlegend angesehen wurden, in Frage gestellt.

In diesem Beitrag wird die Relevanz von Big Data für die Unterrichtsgestaltung gemeinsam mit den Herausforderungen und Möglichkeiten, die sich durch die Einflüsse von Big Data ergeben, analysiert. Dadurch soll die Grundlage für eine zukunftssichere Gestaltung zukünftiger Lehrpläne zum Thema Datenmanagement geschaffen werden.

1 Einleitung

„*Big Data ist ein Ausbildungsthema*“ – obwohl sich dieses Zitat der Gesellschaft für Informatik [GI 13] speziell auf die berufliche (Aus-)Bildung bezieht, hat das Thema auch deutliche Einflüsse auf den Informatikunterricht an allgemeinbildenden Schulen: Die Aufbereitung und Speicherung von Informationen stellt, neben der durch die Öffentlichkeit oft als Kernthema der Informatik betrachteten Softwareentwicklung, eine zentrale Aufgabe sowohl in der Informatik [CS06, Artikel „Informatik“] als auch im Informatikunterricht dar. Während sich bereits verschiedene Paradigmenwechsel in der Softwareentwicklung auch auf den Unterricht ausgewirkt haben – zuletzt der Wechsel zur objektorientierten Programmierung – findet ein solcher Wandel aktuell auch im Datenmanagement statt [MC13]. Dieser Paradigmenwechsel wird dabei insbesondere durch die Verarbeitung großer, unstrukturierter und schnell variierender Datenmengen – Big Data – beeinflusst.

Durch diese Entwicklungen wird sowohl die Informatik als auch der Informatikunterricht vor verschiedene Herausforderungen gestellt. Gleichzeitig ergibt sich aber auch die Möglichkeit, neue Anwendungsgebiete der Informatik darzustellen und den Informatikunterricht motivierend und modern zu gestalten. Durch eine Analyse dieser Herausforderungen und Möglichkeiten, sowie der Relevanz von Big Data für den Informatikunterricht, wird eine Grundlage für zukünftige Lehrpläne zum Thema Datenmanagement im Informa-

tikunterricht geschaffen, die einen zukunftssicheren und aktuellen Unterricht ermöglicht.

2 Aktueller Forschungsstand und Informatikunterricht

Die Themen Datenbanken und Datenverwaltung/Datenmanagement sind in nahezu allen Informatiklehrplänen in Deutschland, aber auch in den GI-Empfehlungen zu Bildungsstandards in der Informatik [Puh08], obligatorisch verankert. Trotz der aktuellen Entwicklungen in den Gebieten Datenbanken und Datenmanagement herrscht in den letzten Jahren jedoch nahezu Stillstand in der fachdidaktischen Diskussion zu diesen Themen, seit zu Beginn der 1990er Jahre die Relevanz von relationalen Datenbankmanagementsystemen (RDBMS) als Unterrichtsthema ausführlich diskutiert wurde (z. B. [Wit94]).

Dies führt zu einer deutlichen Einigkeit bei der Setzung von Schwerpunkten im Datenbankunterricht über alle Bundesländer hinweg, da kaum widersprüchliche Positionen zu diesem Thema existieren. Die wichtigsten inhaltlichen Schwerpunkte im Unterricht stellen daher der Entwurf von relationalen Datenmodellen sowie SQL-Abfragen dar (z. B. [ISB09b]), zur Datenverwaltung wird üblicherweise auf RDBMS zurückgegriffen. Als Beispiele werden dazu oft kleine und fiktive Datenbanken (beispielsweise „Sportverein“ [ISB07]) betrachtet, nur in seltenen Fällen wird mit größeren Datenmengen gearbeitet. Es ist daher absehbar, dass diese Grundlage alsbald ihre Tragfähigkeit verlieren wird.

3 Relevanz von Big Data für den Informatikunterricht

Um für den Unterricht relevante Inhalte festzulegen, können verschiedene Kriterienkataloge zur Auswahl allgemeinbildender Unterrichtsinhalte herangezogen werden, beispielsweise die Kriterien für fundamentale Ideen der Informatik (nach Schwill [Sch93]) oder für Great Principles of Computing (nach Denning [Den03]). Ein wichtiges Kriterium stellt dabei üblicherweise die Forderung nach einer langfristigen Relevanz der Unterrichtsinhalte dar. Dadurch soll verhindert werden, dass kurzlebiges Wissen vermittelt wird, das die Lernenden später kaum in Alltag und/oder Beruf anwenden können. Dieses Kriterium wird nach Schwill als „Zeitkriterium“, bei Denning als „recurrent“ bezeichnet. Während die bisherigen Themen im Datenbankunterricht im Sinne dieser Kriterien als für den Unterricht relevant angesehen wurden, müssen solche Bewertungen im Kontext aktueller Entwicklungen neu getroffen werden: Bisher konnte beispielsweise die Vermeidung von Redundanzen und Inkonsistenzen bei der Speicherung von Daten in normalisierten RDBMS als zentrale Idee der Datenspeicherung aufgefasst werden [EN09]. In nicht-relationalen NoSQL[1]-Datenbanken wird die redundante Datenspeicherung hingegen häufig eingesetzt, um das Antwortverhalten zu verbessern. Damit werden jedoch Inkonsistenzen in Kauf genommen [EFH+11]. Die Normalisierung von Datenbeständen stellt damit ein Konzept von RDBMS dar, das kaum Bedeutung in NoSQL-Datenbanken hat und dessen zukünftige Relevanz daher kaum vorhergesagt werden kann. Die Ideen von Redundanz und Inkonsistenz sind

[1] „NoSQL“ wird im Sinne von „Not Only SQL“ als Oberbegriff für nicht-relationale DBMS verwendet.

mithin für den Unterrichtskontext neu zu bewerten. Andere, oft übergeordnete, Konzepte bleiben hingegen auch in Bezug auf NoSQL-Datenbanken weiterhin relevant, z. B. Datensicherheit, Datenschutz und die strukturierte Datenspeicherung selbst, obwohl NoSQL-Datenbanken statt eines expliziten Datenschemas, wie es bei RDBMS üblich ist, oft ein implizites Schema verwenden. Anhand dieser Beispiele zeigt sich, dass die bisher im Unterricht vermittelten Inhalte in Bezug auf ihre zukünftige Relevanz neu bewertet und Schwerpunkte neu gesetzt werden müssen, um einen zeitgemäßen, dem Stand der Wissenschaft entsprechenden und zugleich zukunftssicheren Informatikunterricht zu gewährleisten.

4 Chancen und Möglichkeiten

Trotz der Herausforderungen, vor welche der Unterricht durch Big Data gestellt wird, werden zugleich auch vielfältige neue Möglichkeiten für die Unterrichtsgestaltung eröffnet. Indem die Einflüsse von Big Data auf den Alltag im Unterricht betrachtet werden, trägt dieser auch wesentlich zu einem besseren Verständnis gesellschaftlich relevanter Themen bei. Beispiele für solche Themen stellen dabei u. a. die Vorratsdatenspeicherung oder Überwachungsprogramme von Geheimdiensten, z. B. das PRISM-Programm der NSA, dar.

Gleichzeitig wird auch ein Bewusstsein für den Wert von Daten geschult, die auf den ersten Blick wertlos erscheinen: Während bisher im Datenschutz insbesondere auf persönliche Daten wie Namen, Adressen und Geburtsdaten geachtet wird, muss im Kontext von Big Data auch der Wert von unscheinbaren Daten beachtet werden. Solche Daten werden beispielsweise beim Besuch von Webseiten automatisch im Hintergrund übertragen. Während bisher zum Schutz persönlicher Daten oft das Prinzip der Datensparsamkeit ausreicht, ist das daher nicht mehr der Fall, wenn Big-Data-Analysen möglich sind. Es muss an dieser Stelle also die Beurteilungskompetenz der Lernenden geschult werden, so dass jeder basierend auf dem erworbenen Wissen selbst entscheiden kann, ob – in Anbetracht des Nutzens einer Anwendung – eine Datensammlung durch diese in Kauf genommen wird.

Auch durch aktuelle Entwicklungen am Arbeitsmarkt, insbesondere durch die Entstehung neuer und die Neuausrichtung bestehender Berufe, werden Anforderungen an den Informatikunterricht gestellt, da der allgemeinbildende Schulunterricht auch die Berufswahl der Schülerinnen und Schüler vorbereiten soll (vgl. z. B. [ISB09a]). Dies geschieht insbesondere, indem Einblicke in die Tätigkeiten verschiedener Berufsgruppen gewährt werden – eine Möglichkeit, die sich auch im Informatikunterricht bietet: Während die Tätigkeiten von Programmierern und auch von Datenbank-Administratoren im Unterricht meist deutlich werden, trifft dies oft nicht auf neuere informatische und informatiknahe Berufe zu. Ein Beispiel für einen solchen Beruf stellt der „Data Scientist“ [DP+12] dar: Dieser Beruf verbindet informatische, mathematische und statistische Aspekte und stellt somit auch die Vielfältigkeit und interdisziplinäre Bedeutung der Informatik dar. Indem Tätigkeiten solcher Berufsgruppen im Informatikunterricht dargestellt werden, wird außerdem gleichzeitig mit der Vorbereitung der Berufswahl auch der Blick auf das Fach Informatik geschärft: Während in der Öffentlichkeit Informatiker häufig mit Programmierern gleichgesetzt werden, wird durch die Betrachtung beispielsweise des Data Scientist deutlich, dass die Tätigkeiten eines Informatikers deutlich vielfältiger als reine Programmierung sein können.

5 Ausblick

Wie dargestellt bieten sich durch die Einflüsse von Big Data und NoSQL vielfältige Chancen für den Informatikunterricht – aber auch Herausforderungen, die in diesem Zusammenhang gelöst werden müssen. Für eine mögliche Neugestaltung des Unterrichts zum Thema „Datenmanagement“ müssen diese ausführlich analysiert und diskutiert werden, damit der Informatikunterricht – gerade auch durch die Betrachtung von modernen Themen – motivierend und auf einem aktuellen Stand bleibt und somit die Begeisterung von Schülerinnen und Schüler für Informatik und informatiknahe Berufe frühzeitig geweckt werden kann.

Die im Rahmen dieses Beitrags dargestellten Einflüsse, Herausforderungen und Möglichkeiten werden durch den Autor weiter untersucht werden. Unter anderem ist derzeit eine detailliertere Analyse der Auswirkungen auf die im Informatikunterricht zum Thema Datenmanagement vermittelten Konzepte in Arbeit.

Literatur

[CS06] V. Claus und A. Schwill. *Duden - Informatik A - Z: Fachlexikon für Studium, Ausbildung und Beruf*. Dudenverlag, Bibliographisches Institut & F.A. Brockhaus AG, 2006.

[Den03] Peter J. Denning. Great Principles of Computing. *Commun. ACM*, 46(11):15–20, 2003.

[DP+12] Thomas H Davenport, DJ Patil et al. Data scientist: the sexiest job of the 21st century. *Harvard business review*, 90(10):70–77, 2012.

[EFH+11] Stefan Edlich, Achim Friedland, Jens Hampe, Benjamin Brauer und Markus Brückner. *NoSQL*. Hanser, Carl Gmbh + Co., 2011.

[EN09] Elmasri, Ramez A. und Navathe, Shamkant B. *Grundlagen von Datenbanksystemen*. Pearson Deutschland GmbH, München, 3. aktualisierte auflage. Auflage, 2009.

[GI 13] GI (Gesellschaft für Informatik e.V.). Handlungsempfehlungen an die politischen Akteure (Big Data Days). http://www.gi.de/fileadmin/redaktion/Hauptstadtbuero/Handlungsempfehlungen.pdf, 2013. zuletzt überprüft: 05.03.2014.

[ISB07] ISB (Staatsinstitut für Schulqualität und Bildungsforschung). Informatik am Naturwissenschaftlich-technologischen Gymnasium, Jahrgangsstufe 9 (Handreichung), 2007.

[ISB09a] ISB (Staatsinstitut für Schulqualität und Bildungsforschung). Das Gymnasium in Bayern (Ebene 1 des Lehrplans des achtjährigen Gymnasiums in Bayern), 2009.

[ISB09b] ISB (Staatsinstitut für Schulqualität und Bildungsforschung). Lehrplan für das Gymnasium in Bayern, Fach Informatik (NTG), 2009.

[MC13] Viktor Mayer-Schönberger und Kenneth Cukier. *Big Data - Die Revolution, die unser Leben verändern wird*. FinanzBuch Verlag, München, 2013.

[Puh08] Hermann Puhlmann et al. Grundsätze und Standards für die Informatik in der Schule: Bildungsstandards Informatik für die Sekundarstufe I. *LOG IN*, 150/151, 2008.

[Sch93] Andreas Schwill. Fundamentale Ideen der Informatik. *Zentralblatt für Didaktik der Mathematik*, 1993.

[Wit94] Helmut Witten. Datenbanken - (k)ein Thema im Informatikunterricht? *LOG IN*, 2, 1994.

Fahrgastinformationssysteme im Kontext von Big Data

Christopher Jud, Bastian Wohlhüter, Mursel Avdiu

Universität Stuttgart
Lehrstuhl für Allgemeine Betriebswirtschaftslehre und
Wirtschaftsinformatik II (Unternehmenssoftware)
cjud@uni-hohenheim.de
bwohlhue@uni-hohenheim.de
mavdiu@uni-hohenheim.de

Art der Arbeit: Seminararbeit
Betreuer/in der Arbeit: Prof. Dr. Georg Herzwurm, Tobias Schäfer

Abstract: Fahrgastinformationssysteme liefern Fahrgästen von Mobilitätssystemen eine Vielzahl von Informationen, die sowohl für die Planung einer Reise als auch während und nach der Reise benötigt werden. In diesem Kontext fallen wachsende Datenmengen an, um den Anforderungen der Anwender gerecht zu werden. In diesem Beitrag wird ein Überblick über die Anforderungen an ein Fahrgastinformationssystem thematisiert sowie dessen Herausforderungen beleuchtet. Zudem wird ein Ausblick über die weitere Entwicklung solcher Systeme gegeben.

1 Grundlagen und Anforderungen an Fahrgastinformationssysteme

Unabhängig davon, ob es sich um die tägliche Fahrt zur Arbeit, zur Vorlesung oder in den Urlaub handelt, wollen Fahrgäste jederzeit sowohl über Tarife und Abfahrten als auch über technische Störungen auf der Strecke informiert werden. Aus diesem Grund werden von den Fahrgästen zunehmend höhere Ansprüche an Fahrgastinformationssysteme (FIS) gestellt, die in Folge dessen mit sehr großen Datenmengen einhergehen. [De13] Durch diese FIS wird den Fahrgästen die Möglichkeit geboten, sich über Ereignisse, die die Reise betreffen, zu informieren und so auftretenden Störungen auf der Strecke vorzubeugen, indem vom System eine alternative Route vorgeschlagen wird.

Entlang der Reisekette ergeben sich verschiedene Anforderungen an FIS. Bereits vor der Fahrt sucht der Kunde über ein FIS nach einer passenden Route. Dabei greift das System auf eine Datenbasis von verschiedenen miteinander kooperierenden Verbundunternehmen zurück und zeigt darüber hinaus zugehörige Tarifinformationen an. [Ro09]

Ebenso sind Kunden im Zeitalter von mobilen Endgeräten in der Lage, sich bereits vor Fahrtantritt mobil über Verspätungen und Staus auf der Autobahn oder in der Stadt zu informieren. Zu diesem Zweck werden gegenwärtige Ist- oder auch Prognosedaten mit den Solldaten abgeglichen. [Fo09] Da sich diese Informationen im Laufe der Zeit verändern können, ist die Verwendung von Echtzeitdaten notwendig. [Ro09] [Zö11] Diese müssen stets aktuell und zuverlässig sein. [Fo09] Die multimodale[1] Fahrplanauskunft stellt im Hinblick auf FIS eine große Herausforderung dar. Neben Informationen zu den eigenen Verkehrsmitteln müssen die Betreiber auch anbieterübergreifende Daten über Alternativen, wie Auto, Fahrrad oder auch Mitfahrgelegenheiten in ihre Datenbank integrieren, um dem Kunden die Informationen kommunizieren zu können. Dadurch, dass lokale, regionale, nationale und internationale Verbünde miteinander kooperieren und sich gegenseitig Fahrpläne und Tarifinformationen zur Verfügung stellen, ist zu erwarten, dass die Datenbasis weiter anwächst. Auch während der Fahrt greift ein FIS auf Daten aus der Datenbasis zurück, um den Fahrgast beispielsweise über auftretende Störungen zu informieren und die Auswirkung dieser Störungen auf die Reisezeit zu berechnen. Darüber hinaus sollte der Fahrgast vor dem Erreichen einer Haltestelle über weitere Anschlussmöglichkeiten – möglichst mit Uhrzeit sowie Gleis- oder Bussteigbezeichnung – informiert werden. [Fo09] Bei Nicht-Erreichen von Anschlussverkehrsmitteln werden die Passagiere über die nächsten alternativen Reisemöglichkeiten informiert, damit diese ihr Reiseziel – unter Umständen auch unter Verwendung anderer Verkehrsmittel – erreichen. [Fo09]

2 Echtzeitdaten und -informationen im Kontext von FIS

Anbieter müssen eine Vielzahl von Daten erfassen, um die Versorgung mit Echtzeitinformationen zu ermöglichen. [Ke13] Als Beispiel seien hier unter anderem die Bewegungsdaten der einzelnen Verkehrsmittel genannt. Somit können technische Störungen an diesen Verkehrsmitteln schnell erfasst werden. Weiterhin müssen Informationen zu geplanten Baustellen berücksichtigt werden, da diese für Verzögerungen im Reiseablauf sorgen können. Soll der Verkehrsfluss auf den Straßen berücksichtigt werden, kann angenommen werden, dass die Datenmenge weiter ansteigt. So ist vorstellbar, dass für eine zeitnahe Erfassung von Verkehrsstörungen unter anderem Daten von Navigationssystemen oder Telemetrie-Systemen von Fahrzeugen gesammelt werden. Hinzu kommen Funktionen, die in der ursprünglichen Anwendung von FIS nicht enthalten sind. [Zö11] So ist zum Beispiel eine Preiskalkulation sowie Bezahlfunktionalität über das FIS eine Funktion, die von den Nutzern zunehmend erwartet wird. Vor allem wenn zwischen Angeboten verschiedener Anbieter gewechselt wird, fallen hier entsprechend Daten an. Gerade bei der Abrechnung von Leistungen (wie Fahrtickets) über ein FIS stellen sich – neben der Erfassung der Preisinformation – zudem einige Fragen (z.B. entsprechende Verrechnungssätze), auf die an dieser Stelle nicht weiter eingegangen werden soll. Weiterhin sollten Echtzeit-Informationen aus sozialen Medien im Kontext von FIS verwendet werden. Diese Informationen können

[1] Die multimodale Reiseplanung berücksichtigt das Angebot mehrerer Mobilitätssysteme für die Nutzung.

für die Stauvermeidung oder Berücksichtigung von bestimmten Ereignissen in die Reiseplanung miteinbezogen werden. Neben der Erfassung und Aggregation der Daten aus den unterschiedlichsten Quellen ist die Auswertung dieser Daten ein weiterer Aspekt, dem genauere Beachtung geschenkt werden muss. Das FIS muss die relevanten Informationen finden und in einen Kontext bringen können. Da zu erwarten ist, dass die FIS zu einem Großteil auf mobilen Endgeräten verwendet werden, welche von der Rechenleistung und Speicherkapazität im Vergleich zu Endgeräten wie Laptop und Desktop-PC stark begrenzt sind, ist der Ort der Aufbereitung der Daten und die darauf folgende Auswertung ein wichtiger Aspekt.

Die Echtzeitdatenerfassung im Rahmen der Verwendung des öffentlichen Personennahverkehrs wurde von der Bundesregierung bereits in Forschungsprojekten gefördert. Eines dieser Forschungsprojekte war die Anwendung „cairo“. Diese Anwendung stellt ein breites Angebot an Nutzungsmöglichkeiten, wie Störungswarnung, Verbindungsauskunft und Echtzeiterfassung von Verkehrsmitteln zur Verfügung. Die Ergebnisse können bisher lediglich im Forschungskontext betrachtet werden. Für Endanwender ist die Anwendung zurzeit noch nicht nutzbar. Zusätzlich werden für die anbieterübergreifende Integration von Daten weitere Anwendungen angeboten (an dieser Stelle seien die Anwendungen memobility und Moovel genannt). Diese haben allerdings einen stark eingegrenzten Fokus und sind daher, gerade für die multimodale Nutzung von Verkehrsmitteln, nur bedingt geeignet. So werden zum Beispiel lediglich die Anbieter von Carsharing-Systemen (memobility) oder nur eine Auswahl von Mobilitätssystemen berücksichtigt bzw. Konkurrenzsysteme außen vor gelassen (Moovel) und sind mit dem Fokus, der mit cairo adressiert wurde, nicht vergleichbar.

3 Fazit und Ausblick

In diesem Beitrag wurden einige Aspekte großer Datenmengen im Kontext der FIS beleuchtet. Das Ziel für die Entwicklung von FIS sollte sein, dass der Nutzer über den gesamten Reiseverlauf von der Planung bis zur Ankunft am Zielort (wie in Kapitel 1 vorgestellt) von dem FIS mit (Echtzeit-)Informationen versorgt wird, die für ihn relevant sind, ohne ihn mit der Masse von Informationen zu überfordern. Diese Ansätze folgen dem SmartData-Paradigma. [BMWi13]

Für die weitere Forschung müssen diese Betrachtungen weiter untersucht werden. Gerade im Hinblick auf die zunehmende Nutzung von FIS auf mobilen Endgeräten ergeben sich einige Fragestellungen. So ist zu untersuchen, inwiefern die Datenaufbereitung und -aggregation auf diesen Systemen umgesetzt werden kann. Ein Ansatz könnte zum Bespiel die Verwendung von Rechen- und Speicherkapazität in der Cloud sein. Im Hinblick auf die Diskussion um die flächendeckende Erfassung von Daten durch Geheimdienste im Jahr 2013 muss auch der Datenschutz-Aspekt dieser Anwendung tiefergehend beleuchtet und geklärt werden, wie die Anfragen und Auswertungen anonymisiert werden können. Andernfalls ist zu befürchten, dass FIS mit entsprechender Funktionalität weniger Akzeptanz bei Endanwendern finden könnten, da

die Gefahr der Erstellung von Bewegungsprofilen durch die Auswertung der Abfragen des FIS besteht. Neben diesen Befürchtungen müssen auch mögliche organisatorische Begrenzungen untersucht werden. So ist unter anderem zu klären, wer für das Angebot haftbar ist (vor allem bei der Verwendung von Daten verschiedener Anbieter) und wie die Verrechnung von Leistungen erfolgen kann. Für FIS ist zukünftig zu erwarten, dass die Berücksichtigung von kontextbezogenen Daten und Informationen immer weiter in den Vordergrund rückt. So ist vorstellbar, dass durch den Einsatz von Augmented-Reality Funktionalitäten FIS durch Points of Interest angereichert werden. Hierzu gehören unter anderem Informationen zu Hotels und Sehenswürdigkeiten sowie die Berücksichtigung von sozialen Medien. Soziale Medien werden im Big Data Kontext immer wieder als Informationsquelle genannt. Diese können für FIS verwendet werden, um die Streckenplanung im Falle von Behinderungen situationsgerecht anzupassen und somit den Reisekomfort für den Nutzer zu steigern.

Bei der Verwendung von Daten mehrerer Anbieter sollte zudem betrachtet werden, inwiefern Angebote unter anderem durch Werbung manipuliert werden können. Dadurch wäre es dem Betreiber des FIS möglich, gezielt Anbieter zu bevorzugen und so eine Vermarktung von priorisierten Angeboten anzustreben. Im Zuge einer gleichberechtigten und aussagekräftigen Routenplanung sollte diese Modifikation des Angebots unterbunden werden.

Literaturverzeichnis

[BMWi13] Bundesministerium für Wirtschaft und Technologie (BMWi): Smart Data – Innovationen aus Daten, Berlin, 2013.

[De13] Dettenbach, J.: Intelligente Fahrgastinformation: Immer auf dem Laufenden. In: Verkehr und Technik 2013, Heft 1. Erich Schmidt Vrlg, Berlin, 2013; S. 23-24.

[Fo09] Forschungsgesellschaft für Straßen- und Verkehrswesen: Hinweise zur Fahrgastinformation im öffentlichen Verkehr, FGSV-Verlag, Köln, 2009.

[Ke13] Digitalisierung und Innovation: Planung – Entstehung – Entwicklungsperspektiven, SpringerGabler, Wiesbaden, 2013; S. 299 - 324.

[Ro09] Roß, J.: Integrierte verbundweite Fahrgastinformation und Anschlusssicherung. In: Verkehrsverbünde – Durch Kooperation und Integration zu mehr Attraktivität und Effizienz im ÖPNV, Verbund deutscher Verkehrsunternehmen, Hamburg, 2009; S. 136-154.

[Sc13] Schelewsky, M., Jonuschat, H., Bock, B. und Jahn, V. (2013), Einfach und komplex - Nutzeranforderungen an Smartphone-Applikationen zur intermodalen Routenplanung, in: InnoZ-Baustein Nr. 14 (Innovationszentrum für Mobilität und gesellschaftlichen Wandel), Berlin, 2013.

[Zö11] Zöller, S. et. al.: Innovative Technologie für mobile Fahrgastinformationssysteme. In: Proceedings of HEUREKA ’11 - Optimierung in Verkehr und Transport. FGSV-Verlag, Köln, 2011; S. 29-48.

Effiziente verteilte Metadaten-Verwaltung auf Basis von ID-Bereichen in DXRAM

Florian Klein

Florian.Klein@uni-duesseldorf.de

Abstract: Für große interaktive Anwendungen sind die Zugriffszeiten auf Plattenspeicher zu langsam. DXRAM ist ein Projekt, dass sich zum Ziel gesetzt hat, Knoten in einem Rechenzentrum zu aggregieren und Milliarden von kleinen Objekten permanent im RAM zu halten. Um diese große Anzahl von Objekten zu speichern und schnelle Zugriffszeiten zu garantieren, müssen die Metadaten effizient und kompakt verwaltet werden. In diesem Aufsatz wird ein neuartiger Ansatz für eine skalierbare Metadaten-Verwaltung auf Basis von ID-Bereichen präsentiert.

1 Einleitung

RAM-basierte Speichersysteme sind in den letzten Jahren sehr populär geworden. Besonders große Webanwendungen nutzen häufig Caches, wie das weit verbreitete memcached System. Facebook zum Beispiel hält 75% der Anwendungsdaten in knapp 1.000 memchached Servern, um kurze Zugriffszeiten für die interaktiven Nutzer zu gewährleisten [ORS+11]. Trotz dessen treten teure Cache-Misses auf. Das Wiederbefüllen des Caches nach einem Fehler ist besonders zeit-intensiv. Im September 2010 verursachte ein Software Fehler bei Facebook das Löschen von 28 TB Daten aus memcached, so dass der Dienst fast 2,5 Stunden nicht erreichbar war, während der Cache von den Datenbank-Servern neu gefüllt wurde [Fac].

RAMCloud war eines der ersten Projekte, dass sich diesen Problemen angenommen hat, mit dem Ziel alle Daten permanent im RAM zu halten. Eine transparente Hintergrundprotokollierung auf Disk-Speicher sorgt für Persistenz, zusammen mit einer schnellen Wiederherstellung der Daten bei Knotenausfällen [OAE+10]. DXRAM folgt dem Ansatz von RAMCloud, ist aber speziell für (soziale Netzwerk-) Graphen konstruiert worden. Im Gegensatz zum Tabellen-basierten Datenmodell mit zentralem Koordinator von RAMCloud, wurde ein Key-Value Datenmodell mit verteiltem Super-Peer Overlay für die Koordination implementiert. Die Super-Peers verwalten dabei nicht nur die Metadaten, sondern überwachen auch die Protokollierung und den Wiederherstellungsprozess. Die persistente Speicherung der Daten erfolgt, ähnlich wie in RAMCloud, durch ein asynchrones Protokollierungsverfahren, jedoch optimiert für Flash-Speicher.

Dieser Aufsatz betrachtet die Verwaltung der Metadaten im Super-Peer Overlay, welche einen neuartigen Ansatz, basierend auf ID-Bereichen, realisiert.

2 Metadaten-Verwaltung

DXRAM wurde für die Verwaltung von binären Daten in einem Rechenzentrum entworfen. Das Hauptentwurfsziel ist die Unterstützung von Milliarden kleiner Objekte (16-64 Byte), wie sie beispielsweise beim Speichern von (sozialen Netzwerk-) Graphen auftreten. Durch das Vorhalten der Daten im RAM und die transpartente Protokollierung auf Flash-Speicher werden die Programmierer von der Synchronisierung von Cache und Sekundärspeicher entlastet und ein schneller Zugriff auf alle Daten gewährleistet.

Die Funktionalität des DXRAM-Kerns umfasst die Verwaltung von RAM- und Flash-Speicher, die Netzwerkkommunikation, die Daten- und Metadaten-Verwaltung (Chunks und Super-Peer Overlay) und die Koordination von Backup- und Wiederherstellungsprozess. Ein Chunk entspricht einem Key-Value-Paar im zu Grunde liegenden Datenmodell. Außerdem gibt es noch zwei Dienste für Sperren und Datenmigration. Eine ausführliche Beschreibung aller Komponenten und Dienste kann in [KS] nachgelesen werden.
Der Kern organisiert binäre Daten in Chunks, die aus einer global eindeutigen *CID* (Chunk ID), einem Längenfeld und den eigentlichen Daten bestehen. Die CID is ein 64-Bit Wert, der aus der Knoten ID des Chunk Erzeugers (NID_C) und einer lokalen ID (*LID*) zusammen gesetzt wird. Die NID_C ist eine 16-Bit große global eindeutige Nummer, die den Erzeuger des Chunks identifiziert (der Besitzer des Chunks kann auch ein anderer Knoten sein, beispielsweise wenn der Chunk migriert wurde). Die LID ist ein 48-Bit großer lokal eindeutiger Wert, der bei jedem lokal erzeugten Chunk um eins inkrementiert wird. Die Größe von NID und LID erlauben es DXRAM insgesamt 65.536 Knoten mit jeweils bis zu 2^{48} (~ 280 Billionen) Chunks zu adressieren. Aus zwei Gründen wurde auf ein Hashing-Verfahren für die CIDs verzichtet und das hier beschriebene Design gewählt. Zum Einen können bei der Verwaltung der Metadaten mehrere Chunks in CID-Bereiche zusammengefasst werden. Zum Anderen können Chunks einfach zwischen zwei Knoten migriert werden, ohne eine Hash-Funktion anzupassen.

Ungefähr 5-10% der zur Verfügung stehenden Knoten in DXRAM werden als dedizierte Super-Peers verwendet. Als Beispiel: In einem System aus 1024 Knoten mit jeweils 32 GB RAM werden 64 Knoten als Super-Peers verwendet. Demnach stehen 2 TB für Metadaten und 30 TB für Daten zur Verfügung.

Jeder Super-Peer SP_{NID} verwaltet die Chunks im Bereich $SP_{NID-1} \leq CID < SP_{NID}$. Die Chunksuche wird mit Hilfe einer NID_C-*Tabelle* durchgeführt, die für jede mögliche NID einen Eintrag besitzt. Die Anzahl an 64-Bit Einträgen entspricht 2^{16}, so dass die Tabelle lediglich 512 KB Speicher belegt. Jeder Eintrag verweist auf einen *CID-Baum*, der die CIDs enthält, die vom entsprechenden Knoten erzeugt wurden. Ein CID-Baum existiert aber nur für die Knoten, für die der Super-Peer zuständig ist. Im genannten Beispiel ist jeder der 64 Super-Peers für 15 der 1024 Knoten zuständig. Bei der Chunksuche wird der NID-Teil der CID als Index in die NID_C-Tabelle verwendet, um den zugehörigen CID-Baum zu erhalten. In dem erhaltenen CID-Baum wird anschließend nach dem LID-Teil gesucht. Als Resultat erhält man die NID des Knotens, der den Chunk aktuell im Speicher hält.
Wenn der CID-Baum für jeden möglich Chunk einen Eintrag enthalten würde, wäre der Baum allerdings viel zu groß und unperformant. Aus diesem Grund wird auf die fort-

laufend erzeugten CIDs zurück gegriffen und lediglich ein `CID-Bereich (start, ende)` zusammen mit der NID_O (o: actual owner) gespeichert. Wenn beispielsweise auf einem Knoten die Chunks mit den CIDs 1 - 1000 vorhanden sind, dann speichert der Super-Peer lediglich einen Eintrag im CID-Baum. Da die Super-Peers für das Mapping von vielen CIDs von all ihren Knoten zuständig sind, ist diese kompakte Speicherweise sehr wichtig.
Wird ein Chunk migriert, tritt eine Aufsplittung des CID-Bereiches ein. Wird beispielsweise der Chunk 500 migriert, so muss der Eintrag (1, 1000) aufgeteilt werden in zwei neue Einträge (1, 499) und (501, 1000). Migrationen werden allerdings nur durchgeführt, um Knoten zu entlasten, die über Chunks mit sehr vielen Zugriffen (Hot-Spots) verfügen, so dass dies kein Problem für die Metdadaten-Verwaltung ist.

Eine weitere Optimierung stellt die Struktur des CID-Baums dar. Wir verwenden einen modifizierten B-Baum um die CID-Bereiche zu speichern und nach einzelnen CIDs zu suchen. Ein B-Baum ist von sich aus schon in Bereiche aufgeteilt und für die Suche optimiert. Die Blätter speichern die NID_O des CID-Bereichs, der durch den Pfad von Wurzel zum Blatt definiert ist. Abbildung 1 zeigt ein Beispiel für einen CID-Baum. In diesem Beispiel wird der Knoten gesucht, der den Chunk 52137 speichert. Nachdem mit der NID 52 der passende CID-Baum aus der NID_C-Tabelle gewählt wurde, wird in diesem nach der LID 137 gesucht. Dazu wird dem Pfad zwischen 100 und 202, dann dem Pfad zwischen 120 und 181 gefolgt und schließlich das Blatt mit der NID_O 34 gefunden.

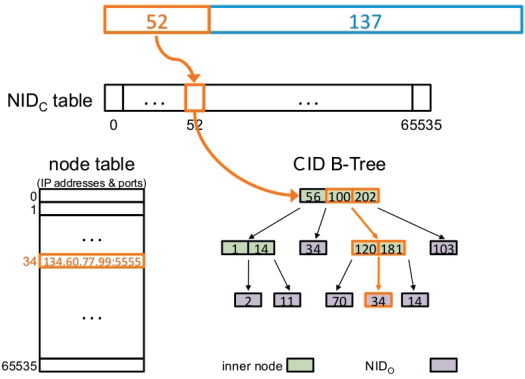


Abbildung 1: CID-Baum

Wenn ein Peer ausfällt, werden seine gespeicherten Daten vom Sekundärspeicher wiederhergestellt. Die Chunks eines Knotens werden dafür in *Backup-Zonen* unterteilt. Für jede Backup-Zone existiert genau ein CID-Bereich im CID-Baum des Super-Peers und drei Backup-Knoten, die die zugehörigen Chunks in ihrem Sekundärspeicher protokollieren. Die Größe der Backup-Zonen kann konfiguriert werden, so dass beim Wiederherstellungsprozess bis zu mehrere hundert Knoten die Daten parallel vom Sekundärspeicher laden. Dieses parallele Verfahren verhindert, dass das Netzwerk oder die lokalen Flash-Speicher einen Flaschenhals bilden. Durch eine Ordnung der Backup-Knoten werden die Daten immer auf dem ersten Backup-Knoten wiederhergestellt. Sollte dies scheitern, wird der zweite Backup-Knoten kontaktiert, usw. Nachdem die Daten geladen wurden, müssen schließlich die Metadaten repariert werden.
Um die Reparatur der Metadaten zu Vereinfachen wird in den Blättern des CID-Baums nicht nur die NID_O gespeichert, sondern auch die NIDs der drei Backup-Knoten für den CID-Bereich bzw. die Backup-Zone. Die vier 16-Bit NIDs passen dabei genau in einen Long Wert. Die NID_O ist in den höchsten 16 Bit abgelegt, dahinter folgen in geordneter Reihenfolge die NIDs der Backup-Knoten. Bei der Reparatur der Metadaten werden diese

Long Werte um 16 Bit verschoben. Dadurch steht die NID des ersten Backup-Knotens nun in den höchsten 16 Bit und entspricht damit der neuen NID_O. Die letzten (jetzt leeren) 16 Bit werden zu einen späteren Zeitpunkt mit der NID eines neu gewählten Backup-Knotens aufgefüllt. Anschließend können die Metadaten wieder verwendet werden.

Die gespeicherten Metadaten eines jeden Super-Peers sind auf seinen Nachbarknoten repliziert. Wenn ein Super-Peer ausfällt, können die replizierten Metadaten auf den Nachbarn verwendet werden, bis ein neuer Super-Peer ernannt wurde. Dieser kopiert sich anschließend die Metadaten und übernimmt die Rolle des ausgefallenen Super-Peers. Sollten alle replizierten Metadaten verloren sein, so können diese rekonstruiert werden, indem die zugehörigen Peers befragt werden.

3 Fazit

In diesem Aufsatz wurde ein neuartiger Ansatz für eine skalierbare Verwaltung von Metadaten. Durch das Zusammenfassen von mehreren Objekten zu ID-Bereichen kann die Menge der Metadaten erheblich reduziert und effizienter verwaltet werden. Sequentiell erzeugte Chunk IDs sorgen für große ID-Bereiche und gleichzeit für eine kleine Anzahl zu verwaltender Daten im Super-Peer Overlay. Die ID-Bereiche werden im Super-Peer Overlay mit Informationen für das Recovery kombiniert und in einer kleinen Anzahl von B-Bäumen verwaltet.

Ein Prototyp wurde bereits fertig gestellt und in der nächsten Zeit werden wir mit der Evaluation und Optimierung beginnen.

Literatur

[Fac] More Details on Today's Outage | Facebook, Sept. 2010. http://www.facebook.com/note.php?note_id=431441338919.

[KS] F. Klein und M. Schoettner. DXRAM: A Persistent In-Memory Storage for Billions of Small Objects. In *Proceedings of the 14th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, PDCAT '13. http://pdcat13.csie.ntust.edu.tw/download/papers/P10013.pdf.

[OAE+10] John Ousterhout, Parag Agrawal, David Erickson, Christos Kozyrakis, Jacob Leverich, David Mazières, Subhasish Mitra, Aravind Narayanan, Guru Parulkar, Mendel Rosenblum, Stephen M. Rumble, Eric Stratmann und Ryan Stutsman. The case for RAMClouds: scalable high-performance storage entirely in DRAM. *SIGOPS Oper. Syst. Rev.*, 43(4):92–105, Januar 2010.

[ORS+11] Diego Ongaro, Stephen M. Rumble, Ryan Stutsman, John Ousterhout und Mendel Rosenblum. Fast crash recovery in RAMCloud. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, Seiten 29–41, New York, NY, USA, 2011. ACM.

Evaluierung von Zugriffsmöglichkeiten auf die NoSQL-Datenbank Apache Cassandra

Tobias Münch

Hochschule Osnabrück
Fakultät für Ingenieurwissenschaften und Informatik
tobias.muench@hs-osnabrueck.de

Art der Arbeit: Bachelorarbeit im Studiengang Medieninformatik

Betreuer/in der Arbeit: Prof. Dr. Frank M. Thiesing, Hochschule Osnabrück (GI-Vertrauensdozent)
M.Sc. Lars Knemeyer, LMIS AG

Abstract: Es werden aktuelle Zugriffsmöglichkeiten aus der Java Platform Enterprise Edition auf die Apache Cassandra Datenbank untersucht. Dabei werden APIs und objektrelationale Mapper, welche als Open Source-Projekt zur Verfügung stehen, miteinander verglichen. Ziel ist es, die verschiedenen Zugriffsmöglichkeiten nach dem Implementierungsaufwand und der Geschwindigkeit zu bewerten. Ferner werden die sekundären Aspekte Dokumentation und Entwicklungsaktivität der Projekte analysiert.

1 Motivation

Die Apache Cassandra Datenbank ([@ACP13]) ist eine NoSQL-Hybrid-Datenbank, welche sich aus den NoSQL-Kategorien Wide-Column-Store und Key-Values-Stores ableitet [EFHB10]. Sie erfüllt die Voraussetzungen für einen Einsatz im professionellen Umfeld [SPI13], jedoch gibt es eine Menge an APIs und objektrelationalen Mappern (ORM), welche noch nicht hinsichtlich des professionellen Einsatzes geprüft wurden. Es gilt diese Menge an ausgewählten Kriterien zu überprüfen, zu bewerten und Anwendungsbereiche für die Zugriffsmöglichkeiten zu definieren, so dass dritten Anwendungsentwicklern eine Orientierungshilfe gegeben wird.

Um die zahlreichen verschiedenen Zugriffsmöglichkeiten zu ordnen, wurde eine Auswahl an populären APIs und ORMs getroffen, welche in der Java Platform Enterprise Edition (Java EE) genutzt werden können. Diese verschiedenen Zugriffsmöglichkeiten werden in den primären Aspekten Implementationsaufwand und Geschwindigkeit untersucht. Zu dem Implementationsaufwand zählt auch die Nutzbarkeit der Java Persistence API (JPA). Ferner werden sekundäre Aspekte, wie Dokumentation und aktive Weiterentwicklung des Projektes, analysiert.

2 Evaluation

Die ausgewählten Zugriffsmöglichkeiten sind in Tabelle 1 aufgelistet. Die Astyanax API wurde als Vertreterin der direkteren Zugriffsmöglichkeiten bestimmt. Ihr gegenüberstehen die OR-Mapper Astyanax ORM, Playorm, Kundera und Hector, welche von unterschiedlichen Herstellen und Gemeinschaften entwickelt werden.

Bezeichnung	Typ	Beschreibung
Astyanax	API	Die Astyanax API (Netflix) bietet eine performante API [@AST13].
Astyanax ORM	ORM	Der Astyanax ORM (Netflix) basiert auf der Astyanax API [@AST13].
Playorm	ORM	Playorm (Bufallo-Software) ist ein OR-Mapper. Primäres Ziel ist es Beziehungen NoSQL-kompatibel zu speichern [@BSP13].
Kundera	ORM	Kundera wird von Impetus entwickelt und ist ein Cross-Database ORM mit vollständiger JPA 2.0 Unterstützung [@KUN13].
Hector	ORM	Hector ist ein Community-Projekt, welches sowohl eine API als auch einen ORM bereitstellt. Es wird derzeit an einer JPA 1.0-Implementierung gearbeitet [@HEC13]

Tabelle 1: Die zu untersuchenden Zugriffsmöglichkeiten auf die Apache Cassandra Datenbank

2.1 Versuchsaufbau

Damit verschiedene Aspekte der unterschiedlichen Mapper betrachtet werden können, wird ein simples Beispiel-Datenmodell gewählt, welches in Abbildung 1 visualisiert ist.

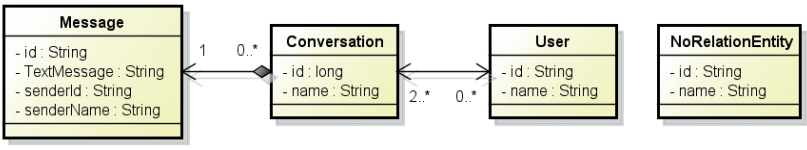


Abbildung 1: Klassendiagramm der Versuchsentitäten

Teilnehmer (*User*) können an Unterhaltungen (*Conversation*) teilnehmen, in welchen Nachrichten (*Message*) gespeichert werden. Dadurch wird eine n:m-Beziehung zwischen Benutzern und Unterhaltungen dargestellt. Eine 1:n Beziehung findet sich zwischen Unterhaltungen und Nachrichten. Mit einer einzelnen Entität (*NoRelationEntity*) wird das Persistieren von Objekten ohne Beziehungen untersucht.

2.1 Vergleich

Der Vergleich wurde auf einem Testsystem mit einem Intel® Core™ i5-2410M @ 2.30 Ghz, der Grafikkarte NVIDEA NVS 4200M und 8GB Arbeitsspeicher durchgeführt. Als Application-Server wurde der TomEE[1] gewählt. Es sind die Operationen Lesen, Schreiben und Löschen auf den Versuchsentitäten - jeweils mit 10.000 Objekten - durchgeführt worden, wobei eine Unterhaltung 50 Nachrichten und zehn Benutzer besitzt.

Performance

Bei der Performance-Analyse des Datendurchsatzes hat sich gezeigt, dass die Astyanax API im Durschnitt am schnellsten bei den Zugriffsmöglichkeiten ist. Der Astyanax ORM und Playorm sind als zweitbeste Möglichkeiten zu sehen. Kundera und Hector sind im Vergleich zu den zuvor genannten Lösungen deutlich langsamer. Bei den ORMs belasten Beziehungen zwischen Entitäten die Performance verhältnismäßig stark. Dies ist bei n:m-Beziehungen am deutlichsten zu sehen.

Implementierungsaufwand

Der Implementierungsaufwand ist bei ORMs geringer als bei dem direkten Zugriff auf die API. Für Entwickler, welche bereits mit JPA vertraut sind, ist Kundera intuitiv und am einfachsten zu verwenden. Playorm nutzt ähnliche Annotationen wie JPA, jedoch mit dem Präfix „NoSql“. Hector und der Astyanax ORM nutzen die Annotationen von JPA zum Teil, jedoch sind an einigen Stellen die JPA-Spezifikationen nicht vollständig umgesetzt. Die Astyanax API ist gut zu verwenden, jedoch müssen eigene Data-Access-Objekte (DAO) für die Entitäten erstellt werden.

Die Testfälle für Hector ORM konnten nur zum Teil umgesetzt werden, da zentrale Methoden des OR-Mappers nicht implementiert sind, wie das Löschen von Entitäten.

Dokumentation

Die Dokumentation der Astyanax API, von Kundera und von Playorm sind positiv zu bewerten, denn es werden sowohl die Funktionsweisen beschrieben als auch anhand von Beispielen verdeutlicht, so dass das Einarbeiten in die Technologie leicht fällt. Die Aystanax ORM-Dokumentation ([@EPN13]) enthält teilweise Lücken und ist nicht vollständig. Die Dokumentation von Hector([@HOM13]) erweist sich teilweise als falsch beziehungsweise veraltet. Dies behindert beim ersten Kontakt mit dem ORM erheblich.

Entwicklungsaktivität

Die Entwicklungsaktivität bei Astyanax ORM/API, Kundera und Playorm ist positiv zu bewerten. Die Reaktionszeit auf Anfragen lag bei den Projekten bei unter 48 Stunden und es werden auch die Tickets aus Github aktiv abgearbeitet. Bei dem Hector ORM gab es keine aktiven Änderungen innerhalb der letzten sieben Monate.

[1] http://tomee.apache.org/

3 Ergebnis

Die Evaluierung der verschiedenen Zugriffsmöglichkeiten zeigt, dass je nach Anwendungsfall eine andere Zugriffsvariante gewählt werden sollte. Wenn der primäre Schwerpunkt auf Performance gesetzt ist, kann die Astyanax API empfohlen werden. Sollte jedoch der Schwerpunkt auf einem möglichst geringen Implementierungsaufwand liegen, kann Kundera empfohlen werden, da die JPA 2.0-Spezifikation vollständig unterstützt wird.

Die Verwendung von Hector ORM kann nicht empfohlen werden, da große Teile des EntityManagers, wie das Löschen von Entitäten, nicht implementiert sind. Die entsprechenden Methoden enthalten lediglich leere Implementierungen. Ein Hinweis darauf über eine Exception oder eine Log-Ausgabe existiert nicht.

Literaturverzeichnis

[EFHB10] Stefan Edlich, Achim Friedland, Jens Hampe, Benjamin Brauer, Markus Brückner: NoSQL - Einstieg in die Welt der nichtrelationaler Web 2.0 Datenbanken, Carl Hanser Verlag, München, 2010

[SPI13] Kai Spichale, Apache Cassandra: NoSQL für schnelle Daten, Analyse und Suche, Java Magazin Ausgabe 11|2013

[@KUN13] Kundera, https://github.com/impetus-opensource/Kundera/

[@HEC13] Hector, https://github.com/hector-client/hector

[@AST13] Astyanax, https://github.com/Netflix/astyanax/

[@EPN13] Entity persister · Netflix/astyanax Wiki, https://github.com/Netflix/astyanax/wiki/Entity-persister

[@BSP13] Buffalo Software » Playorm documentation, http://buffalosw.com/wiki/playorm-documentation/

[@HOM13] Hector Object Mapper - Hector - Java Client for Cassandra, http://hector-client.github.io/hector/build/html/content/HOM/hector-object-mapper.html

[@ACP13] The Apache Cassandra Project, http://cassandra.apache.org/

Performance Analyse zur Speicherung von IATI Daten im Kontext einer Java EE Umgebung

David Paulus

paulusda@hs-pforzheim.de

Abstract: Diese Arbeit untersucht die Frage: Welche Art der Datenhaltung eignet sich für IATI Daten besonders, wenn sie im Kontext einer Java EE Anwendung verarbeitet werden sollen? Die Untersuchung beschränkt sich auf fünf Systeme, darunter die NoSQL Datenbanken MongoDB und BaseX. Untersuchungskriterium ist die Messung der Antwortzeiten der fünf Systeme. Die Ergebnisse zeigen, dass BaseX und MySQL die ersten Plätze belegen.

1 Hintergrund

Die *International Aid Transparency Initiative* (IATI)[1] stellte im Jahr 2008 den selbst entwickelten IATI Standard erstmalig vor. Sinn und Zweck des Standards ist, dass Akteure der Entwicklungszusammenarbeit ein gemeinsames Datenformat nutzen, in welchem sie Projektdaten und Aktivitäten veröffentlichen und über eine zentrale Stelle, die IATI Registry[2], abrufbar machen. Dieser Idee liegt der Wunsch nach einer transparenteren internationalen Entwicklungszusammenarbeit zugrunde. Der Standard definiert unter anderem Finanzflüsse, Budgets, Zeitrahmen, Projektbeschreibungen und geographische Daten.[3] Das standardisierte Veröffentlichungsverfahren erleichtert die Vergleichbarkeit der Daten, deren Zugänglichkeit und Transparenz. Um aus IATI-konformen XML Daten leicht verständliche und einfach zugängliche Informationen für ein breites Publikum zu gewinnen, sind neue (Web-)Anwendungen erforderlich, die die Daten aufbereiten, durchsuchbar machen und visualisieren. Heute befinden sich bereits einige Anwendungen, die auf die ein oder andere Weise mit IATI Daten arbeiten, in Entwicklung oder sind bereits fertiggestellt. Einige stützen sich auf moderne NoSQL Datenbanken, andere vertrauen auf traditionelle relationale Systeme.

[1] http://iatistandard.org
[2] http://iatiregistry.org
[3] http://iatistandard.org/codelists/

2 Vorgehen

Welche Speichervarianten für bestimmte Daten besonders geeignet sind, hängt von den Daten selbst, sowie vom technischen Kontext der Anwendung, in dem die Daten verarbeitet werden sollen, ab. Für diese Arbeit wurde Java EE als technischer Anwendungskontext gewählt, um 1) von den Vorteilen einiger Java EE APIs und Komponenten zu profitieren und 2) eine erste, in Java geschriebene IATI Anwendung prototypisch zu implementieren. Die für diese Untersuchung ausgewählten Datenbanken und IATI API Systeme wurden mit den deutschen IATI Daten[4] befüllt und an einen Java EE Client angebunden. Der Client stellt an jedes System jeweils eine Anfrage nach dem gesamten Datensatz (2400 IATI Aktivitäten) und misst die jeweiligen Antwortzeiten.

3 Bisherige Arbeiten

Diese Arbeit beschränkt sich auf die Messung von Antwortzeiten verschiedener Datenbanken und Systeme. Weitere Evaluationskriterien, die für die Wahl des richtigen Systems entscheidend sind, wie Funktionsumfang, Dokumentationsgrad, Lizenz etc., werden nicht berücksichtigt. Eine Gegenüberstellung mehrerer NoSQL Datenbanken bzgl. weiterer, auch nicht technischer Merkmale, kann [Vai13], [Cat11] und [HJ11] entnommen werden. Gegenüberstellungen von NoSQL- und relationalen Datenbanken finden sich in [SF12], [MK14] und [TB11]. Standardisierte Benchmarks wie in [CST+10] beschrieben, bieten eine sehr gute Grundlage für zukünftige Entscheidungen über die Wahl der richtigen Speichervariante.

4 Untersuchte Systeme

Die Untersuchung verfolgt den Zweck, Implementierungen aus den Gruppen der relationalen, der dokumentenorientierten und der nativen XML Datenbanken einander gegenüberzustellen sowie zwei Systeme, die REST APIs für IATI Daten anbieten, miteinander zu vergleichen. Zu den Datenbanksystemen zählen BaseX[5], eine in Java implementierte native XML Datenbank, MongoDB[6], eine in C++ implementierte dokumentenorientierte NoSQL Datenbank und MySQL als Vertreter der relationalen Datenbanken. Bei den beiden Systemen, die IATI REST APIs bereitstellen, handelt es sich um OIPA (Openaid IATI Parser and API)[7], das in Python implementiert ist und eine MySQL Datenbank verwendet sowie den IDS (IATI Datastore)[8], der ebenfalls in Python implementiert ist und die IATI Daten in einer PostgreSQL Datenbank hält.

[4] http://www.bmz.de/iati/IATI_AcitvityData_R1_002.xml
[5] http://basex.org
[6] http://www.mongodb.org
[7] https://github.com/openaid-IATI/OIPA-V2
[8] https://github.com/IATI/iati-datastore

5 Durchführung und Ergebnisse

Die zu untersuchenden Datenbanktypen und IATI API Systeme sowie ein Java EE Client, der Anfragen an die fünf Systeme stellt und deren Antwortzeiten misst, wurden auf einer dedizierten virtuellen Maschine installiert. BaseX, IDS und OIPA speichern die originalen deutschen IATI Daten. Um auch die MySQL Datenbank mit IATI Daten zu füllen, wurde der Java EE Client um eine Java Persistence API (JPA) Konfiguration erweitert, mittels derer die originalen Daten aus der BaseX Datenbank geladen und leicht abgeändert (Anpassung von Tabellen- und Spaltennamen) in die MySQL Datenbank übertragen wurden. Ebenso wurde mit der MongoDB Datenbank verfahren.

Unmittelbar vor dem Absenden der Datenbank- bzw. API-Anfragen und unmittelbar nach Erhalt der Antworten wurden mittels `System.currentTimeMillis()` Zeitstempel gespeichert und aus den jeweiligen Werten die Differenz gebildet. Es wurden fünf Messungen pro System durchgeführt. Tabelle 1 listet die Ergebnisse auf.

Tabelle 1: Ergebnisse Antwortzeitenmessung

	1.	**2.**	**3.**	**4.**	**5.**	**Ø**
BaseX	00:00,812	00:00,773	00:00,873	00:01,352	00:00,854	00:00,933
MongoDB	00:02,524	00:06,841	00:03,609	00:03,579	00:02,885	00:03,888
MySQL	00:01,514	00:01,263	00:01,371	00:01,290	00:01,407	00:01,369
OIPA	01:21,363	01:17,715	01:18,766	01:17,343	01:15,931	01:18,224
IDS	00:29,783	00:28,383	00:33,448	00:38,236	00:28,169	00:31,604

6 Diskussion der Ergebnisse

Auffallend sind die langen Antwortzeiten von OIPA und IDS, deren Ursachen im Rahmen der Untersuchung nicht abschließend geklärt werden konnten. Wahrscheinlich ist jedoch ein Zusammenhang mit Implementierung und Umfang des Technology Stacks der Systeme.

Zu beachten ist, dass die Client-seitige Verarbeitung der Antworten stark variiert. MySQL und MongoDB schneiden bei der Konvertierung der Antworten in Java Objekte am besten ab, was mit der JPA bzw. EclipseLink Unterstützung der Systeme begründet werden kann. Die Verarbeitung von BaseX-, OIPA- und IDS-Antworten stellte sich im Vergleich dazu als zeitaufwendiger heraus.

7 Fazit

Um die Ursachen der langen Antwortzeiten von OIPA und IDS zu klären, sind weitere Untersuchungen erforderlich. Außerdem wünschenswert sind Tests mit weiteren dokumen-

tenorientierten sowie nativen XML Datenbanken wie zum Beispiel eXist-db[9].

Werden nur die Antwortzeiten berücksichtigt, eignen sich für die Verwaltung von einigen tausend IATI Datensätzen innerhalb eines Java EE Anwendungsumfelds in erster Linie die native XML Datenbank BaseX sowie die von EclipseLink unterstützten Systeme MongoDB und MySQL.

Da IDS automatisch alle verfügbaren Daten von der IATI Registry in eine eigene PostgreSQL Datenbank überträgt, eignet sich das System vor allem dann, wenn mit vielen verschiedenen IATI Daten (d.h. mehrerer Staaten, Ministerien, Organisationen) gearbeitet werden soll. OIPA bietet mit einem Python Django Interface eine einfach zu bedienende Administrations- und Wartungsumgebung.

Literatur

[Cat11] Rick Cattell. Scalable SQL and NoSQL data stores. *ACM SIGMOD Record*, 39(4):12–27, 2011.

[CST+10] Brian F Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan und Russell Sears. Benchmarking cloud serving systems with YCSB. In *Proceedings of the 1st ACM symposium on Cloud computing*, Seiten 143–154. ACM, 2010.

[HJ11] Robin Hecht und Stefan Jablonski. NoSQL evaluation: A use case oriented survey. In *Cloud and Service Computing (CSC), 2011 International Conference on*, Seiten 336–341. IEEE, 2011.

[MK14] D. McCreary und A. Kelly. *Making Sense of NoSQL: A Guide for Managers and the Rest of Us*. Manning Publications Company, 2014.

[SF12] P.J. Sadalage und M. Fowler. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Pearson Education, 2012.

[TB11] Bodgan George Tudorica und Cristian Bucur. A comparison between several NoSQL databases with comments and notes. In *Roedunet International Conference (RoEduNet), 2011 10th*, Seiten 1–5. IEEE, 2011.

[Vai13] G. Vaish. *Getting Started with Nosql*. Packt Publishing, Limited, 2013.

[9] `http://exist-db.org`

Macht ”Big Data” synthetische Datensätze überflüssig?

Thomas Schmid

schmid@informatik.uni-leipzig.de

Abstract: Seit die Speicherung von Mess- und Verlaufsdaten massiv zugenommen hat, verschiebt sich der Fokus mehr und mehr auf deren Auswertung: Kaum eine praktische Anwendung kommt noch ohne Methoden der induktiven Statistik oder des maschinellen Lernens aus. Bedeutet die ungeahnte Fülle an empirischen Daten das Ende von Theorie und Simulation? Anhand der Modellierung impedanzspektroskopischer Messdaten werden hier Nutzen und Risiken synthetischer Datensätze betrachtet.

1 Einleitung

Ziel induktiver Statistik ist es, aus Stichproben allgemein gültige Aussagen abzuleiten. Dabei gilt: Je größer die Stichprobe, desto unwahrscheinlicher ist Zufall. Gleichzeitig ist die Wahrscheinlichkeit Zusammenhänge aufzudecken umso größer, je größer die Zahl der einbeziehbaren Variablen ist. Ganz offensichtlich bietet das häufig als ”Big Data” bezeichnete Phänomen immer umfangreicherer und kleinteiligerer Datenerfassung beste Bedingungen für den Einsatz solcher Techniken.

Doch auch größte Datensätze haben eine zentrale Einschränkung: Bedingt durch ihre endliche Fall- und Variablenzahl stellen sie stets nur ein unvollständiges Abbild oder gar Zerrbild der Realität dar. Nach der Allgemeinen Modelltheorie Stachowiaks gilt darüber hinaus noch eine weitere Beschränkung: Jedes Abbild kann sein Original nur für bestimmte Betrachter, ein bestimmtes Zeitintervall und unter Einschränkung auf bestimmte gedankliche oder tatsächliche Operationen repräsentieren [Sta73].

Je mehr Variablen ein Modell umfasst, desto schlechter ist es geeignet, Zusammenhänge zu erklären. Die Suche nach minimalen Modellen ist daher nicht nur ein typischer Prozess im menschlichen Alltag, sondern für Wissenschaftler sogar ein zentrales Identitätsmerkmal [Nie90]. Ihre Suchstrategie beruht dabei auf Falsifikation, also Theorien und Modellannahmen darauf zu prüfen, ob sie sich durch empirische Belege widerlegen lassen.

Neben Theorie und Empirie hat sich die computergestützte Simulation komplexer Zusammenhänge als weitere Möglichkeit zur Entwicklung und Überprüfung wissenschaftlicher Modelle etabliert. Verbreitet ist die Erzeugung und Auswertung synthetischer Daten vor allem dort, wo die Erhebung von Messdaten teuer oder aufwendig ist[1]. Doch auch wo Daten leicht erhebbar sind – wie mit der in Elekrotechnik und Physiologie häufig angewandten Impedanzspektroskopie –, ist die Verwendung synthetischer Datensätze sinnvoll.

[1] Etwa bei der Simulation von Molekülbewegungen (für eine Einführung siehe zum Beispiel [All04]).

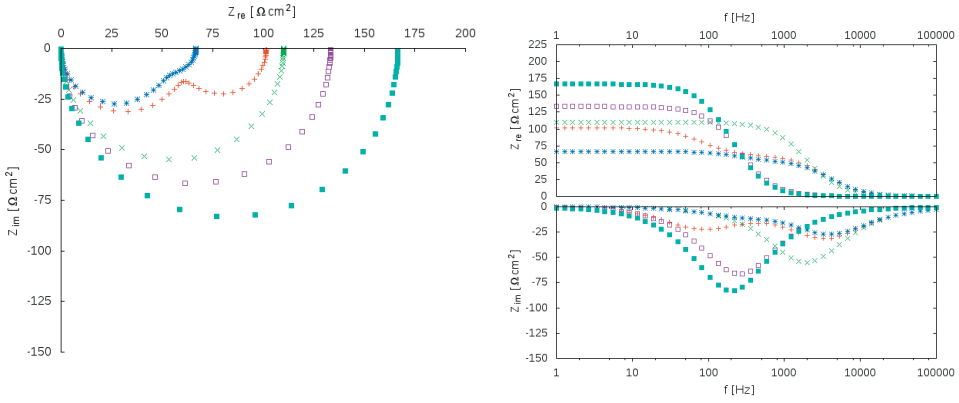
2 Synthetische Impedanzspektren als Beispiel-Datensätze

Eine Impedanz Z beschreibt das Verhältnis von Spannungs- zu Stromänderung in Wechselstromkreisen. Für impedanzspektroskopische Messungen wendet man daher Wechselspannungen variierender Frequenzen auf Untersuchungsobjekte an und misst die Stromänderung. Typischerweise werden 50 bis 100 Frequenzen zwischen 1 Hz und 100 kHz verwendet. Die Ergebnisse der Messungen können in verschiedenen Darstellungen repräsentiert werden, etwa in der komplexen Ebene (Abb. 1a) oder als Bode-Plot (Abb. 1b).

Als Modelle für Untersuchungsobjekte werden elektrische Schaltkreise angenommen [OT08]. Diese können eine theoretisch beliebige Anzahl passiver Elemente wie Widerstände, Kondensatoren oder Induktivitäten enthalten. Für bestimmte menschliche Darm- oder Nierengewebe etwa hat sich ein Modell aus zwei hintereinandergeschalteten Widerstand-Kondensator-Gliedern mit einem dazu parallel liegenden Widerstand etabliert [GZS+12].

Die Annahme eines konkreten Schaltkreis-Modells sowie konkreter Werte für dessen Variablen erlaubt es, die zugehörigen theoretischen Messergebnisse zu berechnen [SGB10]; zusätzlich lassen sich auch messgeräte-spezifische und in der Praxis nicht vernachlässigbare Abweichungen von theoretischen Messergebnisse nachbilden [SBG13]. Durch systematische Variierung der Modellparameter wiederum lässt sich das Verhalten des Modells insgesamt als synthetischer Datensatz abbilden [SGB13, SBG13, SGB14].

Zur Bewertung eines Schaltkreis-Modells wird analysiert, wie gut damit das beobachtete Messverhalten eines Untersuchungsobjekts erklärt werden kann. Traditionell wird dies durch Regressionsanalyse einzelner Messkurven mit einer synthetischen Kurve überprüft [OT08]. Grundsätzlichere Modelle, die über einzelne Messungen hinaus gehen, lassen sich entwickeln, indem auf synthetische Impedanz-Datensätze mit großen Fallzahlen Methoden des maschinellen Lernens angewendet werden [SGB13, SGB14].



a) b)

Abbildung 1. Überlappende Impedanzspektren für ein Schaltkreis-Modell aus zwei seriellen RC-Gliedern und einem parallen Widerstand (cf. [SBG13]), wobei die beiden zugehörigen Zeitkonstanten entweder sehr ähnlich (■, □, ×) oder sehr verschieden (+, ∗) sind. Darstellung als a) Nyquist-Plot (Real- gegen Imaginärteil) und b) Bode-Plot (Real- bzw. Imaginärteil gegen Messfrequenz).

3 Data-Mining und Stachowiaks dritte Modell-Eigenschaft

Ein typischer Arbeitsschritt in der praktischen Datenanalyse ist die Reduktion der Komplexität eines Datensatzes. Genauer gesagt, sollen Variablen identifiziert werden, die wenig zur Beschreibung der Zielvariablen beitragen, und dann vernachlässigt werden. Dafür sind unter den Schlagwörtern Feature Selection oder Dimension Reduction zahlreiche Algorithmen entwickelt worden[2], die dabei helfen sollen, die Zahl der Variablen auf ein Minimum zu reduzieren.

Um durch Reduktion ein minimales Modell zu erhalten, das den angestrebten Zweck erfüllt, muss bereits das noch nicht reduzierte Modell für diesen geeignet sein[3]. Gleichzeitig können nach Stachowiak für ein Modell weder totale Intersubjektivität noch unbeschränkte Geltungsdauer noch absolute Zweckfreiheit angenommen werden [Sta73]. Fasst man Datensätze als (diskrete) Modelle auf, sind folglich bereits für noch nicht reduzierte Datensätze Fallunterscheidungen notwendig.

Kriterien zur Fallunterscheidung sollten sich in der Regel, ebenso wie potentielle Modelleigenschaften, aus dem Anwendungszweck ergeben. Für Untersuchung von synthetischen Impedanzspektren etwa impliziert dies nicht nur unterschiedliche Datensätze für unterschiedliche Schaltkreise und unterschiedliche Fragestellungen, sondern auch eine Unterscheidung hinsichtlich Schaltkreis-Konfiguration [SBG13] oder Zielwert [SGB13, SGB14].

4 Maschinelles Lernen mit synthetischen Datensätzen

Um Zielwerte auch für nicht-definierte Eingangswerte vorhersagen zu können, wird ein reduziertes diskretes Modell in ein kontinuierliches Modell überführt. Während dies traditionell durch Regressionsanalyse erreicht wurde, hat sich dafür im vergangenen Jahrzehnt der Einsatz von Algorithmen aus dem Bereich des maschinellen Lernens eingebürgert[4]. Analog zu Regressionskurven ist deren Qualität unmittelbar abhängig von der Verteilung und dem Wertebereich aller Variablen des zugrundeliegenden Datensatzes[5].

Nutzt man einen empirischen Impedanz-Datensatz zur Ableitung eines kontinuierlichen Modells, definieren die spezifischen Messobjekte und -umstände nicht nur dessen globale statistische Eigenschaften, sondern auch den Zweck dieses Modells. Wird dagegen ein synthetischer Datensatz erstellt, lässt sich dies durch die Wahl des Schaltkreises und der Bauelemente frei definieren [SGB10]. Auch können dadurch intrinsische Modelleigenschaften von extrinsischen (etwa Messfehlern) unterschieden werden [SBG13].

So wie Zielwerte einzelner synthetischer und empirischer Impedanzspektren können auch kontinuierliche Modelle aus empirischen und synthetischen Impedenz-Datensätzen nicht direkt miteinander verglichen werden. Denn die gesuchten Zielwerte sind durch Messung oft nicht oder nur näherungsweise bestimmbar. Betrachtet man jedoch die Differenzen unterschiedlicher Bestimmungsmethoden, können mit Algorithmen des maschinellen Lernens Simulation und Empirie durchaus abgeglichen werden [SBG13].

[2] Für eine Einführung siehe zum Beispiel [Mla06].

[3] Dies stellt eine typische Fehlerquellen für "Big-Data"-Anwendungen dar, da anders als bei Experiment und Simulation meist Datensätze untersucht werden, die nicht für den angestrebten Zweck erhoben wurden.

[4] Für eine Einführung in gängige Verfahren siehe z.B.

[5] Im Unterschied dazu sind jedoch jenseits dieser Wertebereiche meist keine sinnvollen Vorhersagen möglich.

5 Schlussfolgerungen

Im Zeitalter von ”Big Data” stehen für viele neue Anwendungen Mess- und Verlaufsdaten beinahe beliebigen Umfangs zur Verfügung. Doch auch größte empirische Datensätze stellen stets nur verkürzte Abbilder der Realität dar. Als solche unterliegen sie stets einem gegebenen Zweck. Für synthetische Datensätze dagegen lässt sich dieser Zweck frei definieren; insbesondere lassen sich eng begrenzte Modelle theoretisch sogar mit beliebiger Genauigkeit repräsentieren.

Das Erzeugen und Auswerten synthetischer Datensätze ist eine etablierte Säule wissenschaftlichen Arbeitens. Diese erlauben nicht nur, mithilfe von Data-Mining-Methoden minimale Modelle zu entwickeln, sondern auch die Evaluation solcher Modelle mit Methoden des maschinellen Lernens; auch ein effizienter Abgleich mit empirischen Daten ist auf diese Weise möglich. Synthetische Datensätze bilden somit ebenso wie empirische Daten eine unverzichtbare Grundlage für die Entwicklung realistischer Modelle und nützlicher Software-Anwendungen.

References

[All04] Michael P Allen. Introduction to Molecular Dynamics Simulation. *NIC Series*, 23:1–28, 2004. Computational Soft Matter: From Synthetic Polymers to Proteins.

[GZS+12] Dorothee Günzel, Silke S. Zakrzewski, Thomas Schmid, Maria Pangalos, John Wiedenhoeft, Corinna Blasse, Christopher Ozboda, and Susanne M. Krug. From TER to trans- and paracellular resistance: lessons from impedance spectroscopy. *Annals of the New York Academy of Sciences*, 1257(1):142–151, 2012.

[Mla06] Dunja Mladenić. Feature selection for dimensionality reduction. *Lecture Notes in Computer Science*, 3940:84–102, 2006. Volume-Name: Subspace, Latent Structure and Feature Selection.

[Nie90] J. Niehans. *History of Economic Thought*. The John Hopkins University Press, Baltimore, 1990.

[OT08] Mark E. Orazem and Bernard Tribollet. *Electrochemical Impedance Spectroscopy*. John Wiley & Sons, 2008.

[SBG13] Thomas Schmid, Martin Bogdan, and Dorothee Günzel. Discerning apical and basolateral properties of HT-29/B6 and IPEC-J2 cell layers by impedance spectroscopy, mathematical modeling and machine learning. *PLOS ONE*, 8(7), 2013.

[SGB10] Thomas Schmid, Dorothee Günzel, and Martin Bogdan. Using an Artificial Neural Network to Determine Electrical Properties of Epithelia. *Lecture Notes in Computer Science*, 6352:211–216, 2010.

[SGB13] Thomas Schmid, Dorothee Günzel, and Martin Bogdan. Efficient prediction of x-axis intercepts of discrete impedance spectra. In *Proceedings of the 21st European Symposium on Artificial Neural Networks (ESANN)*, 2013.

[SGB14] Thomas Schmid, Dorothee Günzel, and Martin Bogdan. Automated Quantification of the Relation Between Resistor-Capacitor Subcircuits from an Impedance Spectrum. In *Proceedings of the 7th International Conference on Bio-inspired Systems and Signal Processing*, 2014.

[Sta73] Herbert Stachowiak. *Allgemeine Modelltheorie*, chapter 2.1.1, pages 131–133. Springer, Wien, 1973.

StoryTelling : Connecting The News Articles

Vishal Vishal, Navdeep Uniyal, Mohit Makhija, Denduluri Chaitanya, Vamsi Sripathi, Sandesh Nair

Technische Universität Darmstadt
Karolinenplatz 5,64289, Darmstadt

vishal.vishal@stud.tu-darmstadt.de, navdeep.uniyal@stud.tu-darmstadt.de, [mohit.makhija, venkatakrishna_chaitanya.denduluri, vamsi_krishna.sripathi, sandesh.nair]@stud.tu-darmstadt.de

Project Type: Project Group
Supervisor: Dr. Benedikt Schmidt

Abstract: This is the era of information and technology, where knowledge in any form is just a click away. Even though there is an abundance of information one of the biggest challenge that the most of us are facing today is information overload. The enormous amount of data that is being circulated online is way beyond our brain's consumption limit. This excessive and unnecessary bout of information limits our inherent abilities to identify relationships between information and to see the holistic picture. News articles are an interesting example of this problem. The navigation between different articles while generating an in-depths understanding of the relationship and the topic evolution over time is very complex. In this paper we discuss storytelling as a technique that aims at minimizing the effort of the user to browse through news papers and to uncover the relationships and the evolution of topics over time. Therefore, we implement a state of the art algorithm for the creation of so called news article chains. We use the algorithm to provide a query system for the New York Times corpus, containing 1.8 Million documents. The system generates news article chains based on user queries which unfold the relationship and evolution of a topic. The prototype is used to analyse the applicability and performance of the state of the art approach.

1. Introduction

There is a common problem which readers face while reading latest news. Readers might read a news article without knowing the most recent evolution of the topic. Thus, they lack background information and it is a challenge to derive the whole news story and to set the articles they are reading into context. There is no easy way to get the exact information from a set of thousands of articles. In this paper we present a state of the art [DSC10] approach to generate chains of connected news articles and present a system which implements this method to query the New York Times corpus. The system is used to analyze the applicability and performance of the state of the art approach.

In reality when we search for news about a particular topic we have a corpus of millions of documents to search in. In Storytelling we aimed to filter out the documents from the corpus which provide required background information and further find a coherent chain that links them together and delivers “a story” to the user. For example if a user searches for “Snowden accused of leaking US secrets”he wants a structured set of documents to

understand the topic.. As a result of Storytelling a linked chain of the news articles is generated that covers the important parts of the news and presents to the user a story.

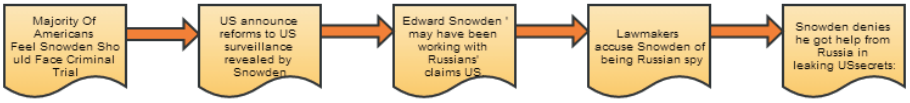


Figure 1: Chain of news articles generated as result.

Our work builds on the work done in [DSC10] which provides a solid base to address the issue of information overload. Despite interesting results, the complexity of the approach results in a major challenge once it comes to the consideration of large news corpora to generate chains from billions of news articles. To address this issue our system implements [DSC10] as base. On top of the approach we use filter methods like user queries and clustering to pre-filter the documents as a foundation for the creation of news article chains. In the following, we discuss the process of generating news article chains we implemented following [DSC10]. As a next step we present our system. Finally, we conclude with summary and an outlook.

2. Approach And Techniques Used

Once the user has filtered documents, we have followed a strategy to create a chain. These chains are the weighted set of documents that are most closely related to each other. According to [DSC10] the process of finding a good chain can be seen as:

- **Formation of coherence:** This means to what extent a document is related to the other documents. This is important to find the connection between the documents.
- **Finding influence of a word on document set:** This gives the importance of the word while making a transition from one document to the other. This helps in finding the best chain of all.

In addition to the above approach we have focussed on implementing the above mentioned techiniques to be used on real world news articles that includes the filtering of documents from a large dataset and then use it make a relvant chain as a story.

In order to find a story between these filtered news articles, we are finding the relation between the words and documents. For this we are dividing our documents to a corpus of words and using lemmatizing[1] and stop words[2] removal to filter the data-set to get the initial corpus. As stated in [DSC10] the chain is as strong as its weakest link. So we have implemented the strategy mentioned in [DSC10] to maximize the weight of the weakest link in the chain, i.e. maximize the weight of edge with minimum weight.In order to determine how documents are related to each other, we created a bipartite graph containing all words and documents. For every word w in document d, we added edges (d,w) and (w,d) to the graph. But since words can appear in a document and still be irrelevant, or the other way: Words NOT appearing in a document can still be important, we needed to assign importance to the edges. We chose to calculate TF-IDF-values and use them as weights for the edges.

To express the relationship of one document to another, we did random walks. A random walk starting at one document should reach the other document frequently, if the two documents are highly related.

2.1 Influence Calculation: First, in our graph, we have normalized the document-to-word edges over words and the word-to-document edges over documents, so that we could interpret the values as random walk probabilities. The start point in the chain would always be the document which the user is reading currently. Considering it as a starting point we applied random walk algorithm [MCS90] with restart (replacing the word with the restart vector) and then repeating the operation until it converge in order to get the influence value. In this step we computed the stationary distribution for random walks starting of the graph from one document. Then, repeating for all words, we turned one word to a sink node and computed the stationary distribution again. We did this, because if the selected word was an important one, the stationary distribution of the target document would decrease a lot.

2.2 Scoring the chain: The words in the chains were activated first and then weighted.

- **Smoothness:** As stated in [DSC10] we needed to activate the words in the chain one by one by giving them some value. Here, we implemented the problem by giving the binary values to the words which were present in the document or not. As mentioned in [DSC10] we considered certain constraints, "the word need to be activated just once in the chain" and "no word is active before the chain begins" are two of those constraints on which we formulated the linear program. Also when the word got initialized it was given a binary value.
- **Objective:** Our objective was to find the weakest link and to maximize the weight on that. In order to achieve this we used the linear formulation:

$$\textbf{Minedge} <= \sum \textbf{Active\_word}_{w,i} * \textbf{influence}\ (d_i, d_{i+1} | w)$$

2.3 Finding A Good Chain: Now, as described in [DSC10] we optimized our search jointly over words and chains instead of using the iteration technique. We again formulated a linear program to achieve the goal. In addition to the node activation and initialization (which is done in LP1), we focused on adding weights to the edges between the documents. Variable 'next_node$_{i,j}$' defines if there existed a link between d_i and d_j. We have considered another LP variable 'trans_active$_{w,i,j}$' which denotes if the word *w* was active during the transition from document d_i to d_j. As motivated by [DSC10] we formulated the Linear Problem similar to the one done earlier with one added module:

- **Chain Restrictions:** We considered a target document, *t* and a start document, *s* with *n* number of documents and *n*-1 edges. All the nodes considered just one incoming and one outgoing edge. Active value for node *s* and node *t* was always 1. The output is a chain ordered chronologically.
- **Smoothness:** Similar to the Linear program formulated earlier. We implemented the problem by giving the binary values to the active word which were present in the document or not.
- **Objective:** We defined the objective function which takes the minimum value of all active edges and then maximized it to get the best chain.

$$\textbf{Minedge} <= \textbf{1-next\_node}_{i,j} + \sum \textbf{trans\_active}_{w,i,j} * \textbf{influence}(d_i, d_j | w)$$

3. New York Times Query System

We have tried to port this algorithm for a large corpus of [NYTC] which contains 1.8 million documents and we still are trying to optimize the algorithm. First step is to filter out the documents which are similar in context. As in *figure1* (user is reading news about "Snowden") First challange would be the large dataset which he has to go through to understand the issue. Secondly, the occurence of the events and relation between the news articles. So, our goal was to minimize the efforts of end users by making the data set smaller with news items about "Snowden" which were highly related to each other without losing any relevant information.

In order to test the implementation, we used Solr [SOLR] queries to fetch the data according to the query. To make it efficient we used clustering in Solr by which it makes small subsets of news items which are related to a particular topic. Using this strategy, we could group the related news items together. Thus, making our initial set smaller. Now to bring down the document set to a more realistic number (which could be handled by the prototype) we used filters in the Solr query. Using filters we tried bringing down the set size and then applied our algorithm on the resulting set to get the desired results.

4. Conclusion and Future work

Our primary goal was to help user unravel the hidden connections between two news articles. To achieve this we used clustering along with Apache Solr[SOLR] queries which works on corpus database. This made our system more efficient as it provides documents that are much refined and more similar . Then we developed an efficient algorithm based on the approach described in [DSC10] to get a best coherent chain. We believe the system proposed would help give user the ability to create a logically related chain of news items depending upon his interest on a particular topic.

In the future we would like to explore a way in which user can backtrack to the starting document which is related to the target document that the user is reading so that the links between the two articles can be easily understood. We also plan to make this algorithm more efficient by using K-coherence [DCE12] which requires computing of shorter chains and concatenating them to find the relation.

5. References

[DSC10] Dafna Shahaf and Prof. Carlos Guestrin : Connecting the dots between news articles.ACM SIGKDD , (KDD) 2010.

[NYTC] New York Times Corpus : http://catalog.ldc.upenn.edu/LDC2008T19

[MCS90] Macropol K, Can T, Singh AK - BMC Bioinformatics (2009)

[MBM05] Michael D. Lee, Brandon Pincombe and Matthew Welsh : An Empirical Evaluation of Models of Text Document Similarity(2005).

[SOLR] http://lucene.apache.org/solr6/documentation.html

[DCE12] Dafna Shahaf , Prof. Carlos Guestrin and Eric Horvitz : Train of Thoughts: Generating information maps(2012)

Kontextsensitive Informationsgewichtung auf Basis des Semantic Web

Lars Wesemann
lars.wesemann@unister.de
F & E, Unister GmbH, Leipzig
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Betreuer der Arbeit: Prof. Dr. Klaus Hering, Dr. Andreas Both, Ricardo Usbeck

Abstract: Der ständig wachsende, digitale Informationsberg in Form von un- oder semi-strukturierten Webseiten beherbergt wertvolle Ergebnisse für die meisten gestellten Web-Suchanfragen. Da ein Nutzer jedoch nur begrenzt Zeit und Ressourcen hat, müssen die relevantesten Suchergebnisse an den ersten Positionen einer Suchergebnisseite angezeigt werden. In dieser Arbeit wird ein kontextbasierter Ranking-Algorithmus *mEVA* vorgestellt, welcher auf Basis von Linked Open Data und etablierter Graph-Algorithmen eine Verbindung zwischen Webseiten, semantischem Wissen und Suchanfragen herstellt. Bei der Evaluation auf einem realen Datensatz konnte gezeigt werden, dass mEVA dem als Basisalgorithmus verwendeten *PageRank* überlegen ist.

1 Einleitung

In Zeiten wachsender Informationsmengen verlangen Nutzer einen schnellen Zugang zu Informationen. Insbesondere sollen Suchmaschinen schon an den ersten Positionen ihrer Ergebnisseiten relevante Informationen platzieren.[1] Dabei existieren viele Schwierigkeiten wie bspw. die Extraktion von Wissen aus unstrukturierten Datenmengen oder mehrdeutige Suchanfragen (z.B. `Golf [Geographie]` vs. `Golf [Sport]`). Mit Hilfe entsprechender Kontextinformationen kann die Antwort gezielt gefunden werden.

Das *Semantic Web* [BLHL+01] ist eine Erweiterung des Internets und stellt als Framework Hilfsmittel zur Verfügung, um Dokumente mit Zusatzinformationen anzureichern. Diese Zusatzinformationen werden üblicherweise über *Linked Data* mit anderen Informationen verbunden.

Hier setzt der in dieser Arbeit entwickelte Ranking-Algorithmus *mEVA* an. mEVA ist ein Kontext-basiertes Rankingverfahren auf Basis des Semantic Web. Als Kontext werden allgemeine Suchegebiete wie bspw. Städte oder Inseln verwendet. Insbesondere wurde bei der Entwicklung auf ein praktikables Konvergenzverhaltens, d. h. eine angemessenen Laufzeit auf realen Daten, geachtet. Wir evaluieren die Qualität des Algorithmus gegenüber state-of-the-art Algorithmen mit Hilfe eines speziell generierten Datensatzes.

[1] http://googleblog.blogspot.de/2009/02/eye-tracking-studies-more-than-meets.html (16.01.2014).

2 Verwandte Arbeiten

Eine Besondere Bedeutung im Kontext der Arbeit besitzen Graph-basierte Ranking-Algorithmen, wie bspw. PageRank [PBMW99]. Der PageRank basiert auf die Vernetzung von Webseiten untereinander, wobei gilt: je mehr Links auf eine Webseite verweisen, desto höher ist die Relevanz der Webseite für eine Suche.

Um Informationen des Semantic Web zu nutzen, wurden in den letzten Jahren diverse Ranking-Algorithmen entworfen. Ein Linked-Data-Graphen-Gewichtetes Verfahren ist ReconRank [HHD06]. Dieser Ranking-Algorithmus beachtet die Struktur zwischen den Entitäten einer Ontologie und basiert auf den PageRank.

Der in dieser Arbeit vorgestellte Algorithmus basiert auf dem von Julia Stoyanovich entwickelten Ranking-Verfahren EVA [Sto10, Kap. 4]. Bei diesem Verfahren wird eine Verbindung zwischen den Dokumenten und Linked Data hergestellt und zum Ranking der Dokumente genutzt. mEVA erweitert diesen Algorithmus um eine Kontextanalyse, welche beim anschließenden Graph-basierten Ranking beachtet wird. Die bei Stoyanovich uniform gewählten Knotengewichte werden bei mEVA durch eine Kontextgewichtung ersetzt.

3 Methode

mEVA dient der Positionierung von Dokumenten auf einer Suchergebnisseite. Je weiter oben sich ein Dokument auf der Ergebnisseite befindet, desto relevanter ist es. Für die Berechnung der Relevanz wird ein Web Graph (alle Webseiten und deren Link-Struktur) und ein Linked-Data-Graph (alle Entitäten und ihre Beziehungen kurz: LD Graph) zum General Data Graph (kurz: GDG) zusammengefasst. Dafür wird ein Informationsextraktionsalgorithmus verwendet, welcher auf Label-Matching basiert. Die beiden Graphen werden bidirektional verbunden, d. h. aus den Webseiten extrahierte Semantic-Web-Entitäten werden mit Entitäten aus dem Ontologie-Graphen verbunden, (siehe Abbildung 1).

Gleichung (1) stellt mEVA dar und setzt sich wie folgt zusammen: p ist ein Dokument, $A(p)$ der Relevanzwert eines Dokumentes, N die Anzahl der Knoten im GDG, d der neu eingeführte Dämpfungsfaktor (meist 0.85), $O(x)$ eine Entität aus dem LD Graph und $P(x)$ ein Dokument aus dem Web Graph. w ist die Gewichtsfunktion mit der Dokumente bezogen auf ihren Kontext stärker oder schwächer gewichtet werden können.

$$A(p) = \frac{(1-d)}{N} + d \cdot \sum_{p \in (O(x) \cup P(x))} A(x) \cdot w(x \to p) \tag{1}$$

Befindet sich eine Entität auf einem Dokument im gesuchtem Kontext, so lässt sich Bedingung (2) definieren:

$$w(x \to p) = \begin{cases} boost & p \text{ ist Dokument und } x \text{ Entität vom Kontext} \\ 1 & \text{sonst} \end{cases} \tag{2}$$

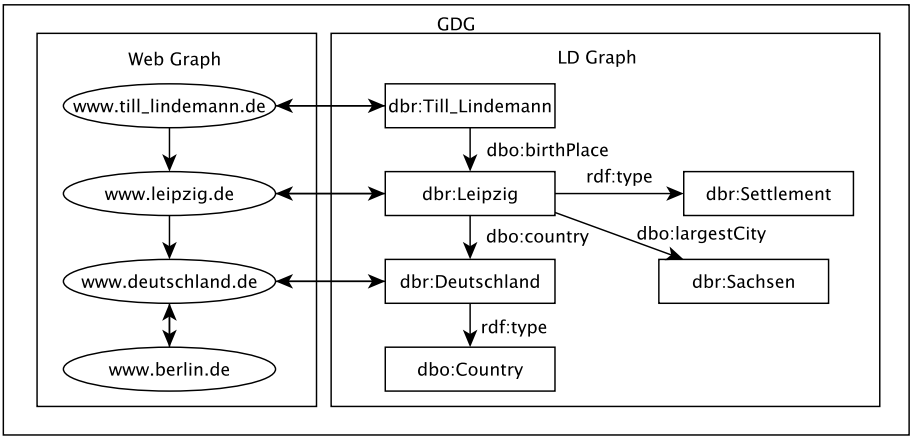


Abbildung 1: Aufbau des GDG.

Die Variable *boost* erhöht die Gewichtung eines Dokumentes und wird > 1 gewählt. Eine Entität ist eine Instanz einer Klasse (über die Relation *rdf:type*) und repräsentiert ein reales Objekt samt Eigenschaften. Mehrere Entitäten können über Relationen miteinander verbunden sein und bilden so den LD Graph. Eine Klasse beschreibt eine Kategorie, wie bspw. *Car* oder *Settlement*. Eine Entität befindet sich in einem bestimmten Kontext, wenn die Klasse (*rdf:Class*) der Entität im gewünschten Kontext liegt.

4 Experiment und Ergebnisse

Anforderungen Im Folgenden soll evaluiert werden wie gut sich mEVA als Anfrage-unabhängiger Ranking-Algorithmus eignet. Dazu vergleichen wir die von mEVA und PageRank zurückgelieferten Webseiten bezüglich ihres Kontextes.

Datensatz Als Korpus wurde wikitravel[2] verwendet. Dieser Korpus enthält 27.144 Webseiten (Web Graph). Als LD Graph wurde die DBpedia[3] in der Version 3.8 genutzt. Es wurden 1000 Dokumente entsprechend ihres Kontextes annotiert, wobei sich insgesamt 163 Dokumente im Kontext *Settlement* befinden und als Fallbeispiel dienen.

Ergebnisse Für die Evaluierung der Ergebnisse wurde die Precision@k [FBY92] von mEVA mit diversen Gewichtungsfunktionen (boost $= 1, \ldots, 1000$) automatisch getestet.

Verwendet man mEVA mit dem optimalen boost $= 60$ im Kontext *Settlement* (Abb. 2), sind auf den ersten fünf Positionen bereits drei kontextbasierte Dokumente, während der PageRank erst auf Position 12 das erste kontextbasierte Ergebnis liefert. Somit ergibt sich

[2] `http://wikitravel.org/de` (09.10.2012).
[3] `http://dbpedia.org` (22.11.2012).

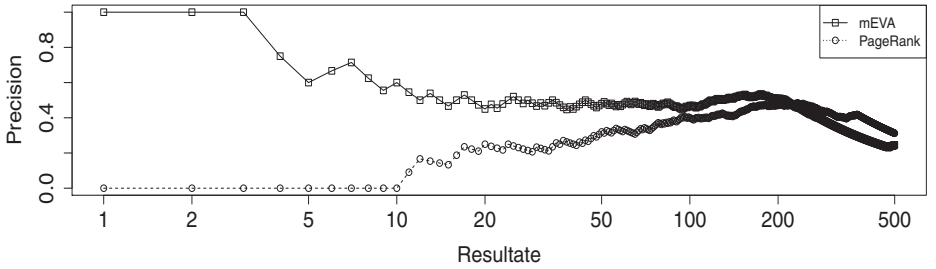


Abbildung 2: Vergleich von mEVA und PageRank (durchgezogene bzw. gepunktete Linie).

für den PageRank eine Precision@5 von 0.0 und für mEVA 0.75, wodurch ersichtlich ist, dass sich mEVA für ein kontextbasiertes Ranking eignet.

5 Zusammenfassung

Das in dieser Arbeit vorgestellte Algorithmus mEVA wurde nach eingehender Literaturrecherche in bestehender Forschung noch nicht untersucht. Wie die Ergebnisse zeigen, ist mEVA für ein kontextbasiertes Ranking von Dokumenten geeignet. Durch die Hilfe des Semantic Web ist eine Kontexterkennung und Nutzung dieser Information erst möglich. Die Kombination großer, unstrukturierter Corpora und Linked-Data, als enzyklopädischer Hintergrund zur Verbesserung des semantischen Hintergrundwissens, dient hier der Verbesserung des Sucherlebnisses des Nutzers.

Auf unserer aktuellen Forschungsagenda stehen neben einer verbesserten Entitäts-Erkennung auch die Ausrichtung von mEVA auf Cloud-Computing zur Effizienzsteigerung.

Literatur

[BLHL+01] T. Berners-Lee, J. Hendler, O. Lassila et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[FBY92] William B. Frakes und Ricardo Baeza-Yates, Hrsg. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[HHD06] A. Hogan, A. Harth und S. Decker. Reconrank: A scalable ranking method for semantic web data with context. 2006.

[PBMW99] L. Page, S. Brin, R. Motwani und T. Winograd. The PageRank citation ranking: Bringing order to the web. 1999.

[Sto10] J. Stoyanovich. *Search and ranking in semantically rich applications*. Dissertation, Columbia University, 2010.

A graph theoretical approach for exploring a board game's complexity

Mareike Bockholt
mareike.bockholt@cs.uni-kl.de

Bachelor's thesis (2013), University of Heidelberg
Supervisors:
Prof. Dr. Gerhard Reinelt (University of Heidelberg),
Prof. Dr. Katharina Zweig (TU Kaiserslautern)

Abstract: In cognitive psychology, human problem solving has been an active field of research over the past decades. During this period, methods in computer science and network analysis have been greatly developed. Looking at psychology's challenging questions from a computer scientist's point of view proves to be worthwhile for both disciplines. In this work, we analyze a single-player puzzle called *Rush Hour* which is solved by moving entities on a board. In a graph theoretical approach, each distinct combination of entities and their location can be represented as a node; an entity's position change can be understood as an edge. We are interested in understanding the difficulty classification of the popular board game *Rush Hour* by using algorithmic approaches. The present paper concentrates on the results of a conducted study involving 74 subjects. Based on the findings, we investigate complexity measures, the subjects's navigation through the problem space as well as the game's perceived complexity. On the one hand, the results of the analysis suggest a more finely nuanced grading by difficulty for the *Rush Hour* levels. On the other hand, they contribute to a better understanding of the human perception of complexity.

1 Introduction

It is well known that problem solving capabilities of humans and computers differ in several aspects: While humans can make use of their experiences, creativity, and some kind of intuition, computers must rely on the given data and algorithms in which not all real world constraints may be implemented. On the other hand, in processing and storing a big amount of information and dependencies, computers clearly outperform humans. Hence, it might be a promising approach to combine the structural advantages of human and artificial problem solving abilities in order to construct human-computer cooperative and interactive systems [AAL+00]. In order to divide subtasks between human and computer agents, it is necessary to better understand why some subtasks may be a challenge to solve for both. In computer sciences, complexity theory has been providing a broad range of results about problems' difficulty for being solved by algorithms. However, in cognitive sciences, there are only a few approaches to systematically analyze a problem's complexity for humans to solve it ([RSF12], [KHS85], [HWP98]).

The present work is aiming to investigate a simple puzzle's complexity by graph theoretical methods. By considering the game's problem space as a graph, network analytic metrics are defined and the correlation between the metrics and the manufacturer's difficulty rating is analyzed, hoping to be able to classify game instances by problem space based measures into distinct complexity groups.

However, the present article primarily describes the findings of a conducted experiment in which the participants played some of the studied games which were selected due to their complexity metrics. The results' analysis reveal essential flaws in human problem solving abilities which could be compensated by a computer-aided system.

2 Backgrounds

Anderson defines problem solving as "goal-oriented sequence of cognitive operations" that transforms a present state into a desired state [And80]. From this definition, the concept of a problem space arises almost immediately. A problem space is defined as a set of problem states and set of operators such that an operator transforms two states into each other. Sequential application of operators yields a path through the problem space. Then, a problem belonging to a problem space can be considered as a set of start states, a set of goal states and a set of path constraints. Therefore, problem solving consists of the task to find a path through the problem space from a start to a goal state without violating the path constraints which can be seen as a searching task.

We are going to consider the sliding block puzzle game *Rush Hour* which takes place on a grid of 6×6 cells, representing a parking lot, with one exit (cf. figure 1(a)). Cars of width 1 and length 2 respective 3 cells are placed on the board vertically or horizontally and can be moved forwards or backwards as long as the for the movement needed cells are not occupied by any other car. Cars cannot move sideways and are not allowed to change their row or column, respectively. Given a configuration of cars placed on the grid, the goal is to find a sequence of moves that allows a particular car (in figure 1(a) the black one) to be moved from the board through the designated exit.

We consider the problem space of a Rush Hour game as the graph $G = (V, E)$ with V the set of all game configurations reachable by allowed moves, $E \subseteq V \times V$ the set of possible moves. Due to the game's simplicity, it is feasible to explicitly construct the graph. The manufacturer of the Rush Hour game provides five game card sets with start configurations of five different levels of difficulty. Leaving identical configurations and configurations with a slightly different goal aside, 173 start configurations with the manufacturer's rating of complexity are available. Implementing the simple game logic and a breadth first search from every start configuration, the problem spaces were explored and, based on theoretical considerations, 24 configurations were selected for experimental analysis.

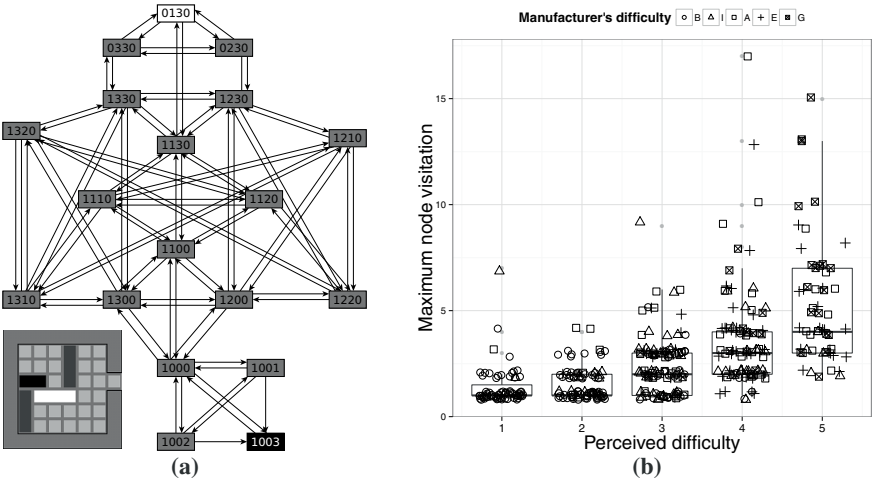


Figure 1: (a) shows the problem space for the configuration shown. The number in the nodes correspond to the positions of the cars in their row or column: the first/second/third/fourth digit corresponds to the black/white/left grey/right grey car's position. White node is the start node, black node a solution node.(b) is the maximum node visitation of all players and all games against its perceived difficulty. Shape represents the producer's difficulty estimation (beginner, intermediate, advanced, expert, grand master)

3 Experiment

Theoretical considerations which are not described in this article, suggested that the producer's complexity statement is only based on the number of needed moves. In order to investigate if the true complexity depends on further factors, an experiment was conducted in which 74 participants played at least six of 24 selected Rush Hour games such that every configuration was played by at least 20 participants. The participants played the game online, but it was made sure that no participant played one game more than once. When a player successfully finished a game, she or he was asked to grade it by difficulty. For the analysis, only data sets of completed games were used.

The results show, as expected, the strong influence of needed moves to solve the game on its perceived difficulty, but indicate that there exist other factors contributing to complexity. Therefore, we were interested in how the participants navigated in the game's problem space while solving the game. It is remarkable that the majority takes exactly the same way through the problem space which is not necessarily the shortest one. This finding is consistent with the observation from cognitive sciences that humans use the same heuristics for solving problems. Additionally, we could find a dependency of a player's difficulty estimation and her individual navigation through the problem space: the more a player "becomes lost" in the problem space, the more difficult the game is perceived. Modeling a player losing her orientation can be done by several approaches: If a player gets lost, her way through the problem space will certainly be longer than required, and indeed, a strong correlation between the perceived difficulty of a game and the quotient of the number of used moves to the number of moves in the optimal solution can be found. Surprisingly, in games rated by the players as very hard, the players need in average five times as many moves than it would be necessary in the optimal solution. Even in the games rated as very

easy, the players need in average about 1.2 times as many moves as the optimal solution takes. Another approach considers configurations which occur several times in a player's solution (clearly not optimal). For every node in the solution of a player, define the *node visitation* as the number of how often this player uses that particular node in her solution. Maximizing over all nodes of a player's solution yields the *maximum node visitation* whose correlation to the perceived difficulty is depicted in figure 1(b). This leads to the conclusion that a problem's complexity does not only depend on objective properties, but there is also a high correlation with the individual performance. The values of the maximum node visitation take on surprisingly high values indicating that getting lost in a huge problem space is a general issue in human problem solving. But this could, though, be easily avoided by computer support, since recognizing a repeating configuration can be done algorithmically without need of completely solving or even knowing the problem. Thus, this analysis of a problem which may seem artificially constructed leads to the suggestion of the following human-computer cooperative system: the human can make use of intuition, creativity, and heuristics to solve the problem with a problem space which may be too large for a purely algorithmical solution, and the computer gives notice of repeating configurations, based on local computations, pointing the human to the right direction.

4 Outlook

There are several approaches that could be promising for further work. The observations that the resulting problem spaces are surprisingly large and that not all contained states may be relevant, as it is not necessarily important in which order moves are taken, leads to the thought whether one can merge equivalent states and therefore reduce the problem space. Reduced problem spaces could be worthwhile to consider. Furthermore, it would be an interesting question how repeating configurations in a human solution can be recognized and stored efficiently without computing the complete problem space or consider every used configuration.

References

[AAL+00] David Anderson, Emily Anderson, Neal Lesh, Joe Marks, Brian Mirtich, David Ratajczak, and Kathy Ryall. Human-guided simple search. In *AAAI/IAAI*, pages 209–216, 2000.

[And80] John R. Anderson. *Cognitive Psychology and its Implications*. W. H. Freeman and Co., New York, NY, 1980.

[HWP98] G. S. Halford, W. H. Wilson, and S. Phillips. Processing capacity defined by relational complexity: implications for comparative, developmental, and cognitive psychology. *Behavioral & Brain Sciences*, 21:803–865, 1998.

[KHS85] Kenneth Kotovsky, John R Hayes, and Herbert A Simon. Why are some problems hard? Evidence from Tower of Hanoi. *Cognitive psychology*, 17(2):248–294, 1985.

[RSF12] M. Ragni, F. Steffenhagen, and T. Fangmeier. A Structural Complexity Measure for Predicting Human Planning Performance. *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, 2012.

Komplexitätstheoretische Klassifizierung von Äquivalenzrelationen für Boolesche Funktionen

Sebastian Flothow

Hochschule RheinMain
Studiengang Master Informatik
sebastian@flothow.de

Art der Arbeit: Master-Thesis
Betreuer: Prof. Dr. Steffen Reith

Abstract: Die Äquivalenz aussagenlogischer Formeln ist ein sehr bekanntes $\mathbf{coNP}$-vollständiges Problem. In der vorliegenden Arbeit wurden verschiedene Varianten dieses Problems untersucht. Zum einen kann die zugrunde liegende Äquivalenzrelation verallgemeinert werden, zum anderen können Schaltkreise über beliebigen Basen betrachtet werden. Die Untersuchung liefert sowohl einige bisher unbekannte $\oplus\mathbf{L}$-vollständige Probleme, als auch solche, die in der Polynomialzeithierarchie zwischen den Klassen $\mathbf{coNP}$ und $\mathbf{\Sigma_2^p}$ liegen.

1 Äquivalenz von Booleschen Funktionen

Verallgemeinerte Formen der Äquivalenz Boolescher Schaltkreise wurden bereits im 19. Jahrhundert untersucht; dazu ließ man z.B. Permutation und Negation der Variablen oder auch Verknüpfungen mehrerer Eingabevariablen zu. Das Ziel bestand anfangs darin, Boolesche Funktionen durch möglichst wenige Basisschaltkreise darstellen zu können, die dann durch einfache Zusatzbeschaltungen angepasst werden können; dabei stellte sich vor allem die Frage, wieviele Äquivalenzklassen es gibt [Thi00].

Vergleichsweise neu ist dagegen die komplexitätstheoretische Betrachtung dieser Äquivalenzrelationen. Für die klassische Äquivalenz ist die Entscheidung, ob zwei Boolesche Schaltkreise äquivalent sind, ein $\mathbf{coNP}$-vollständiges Problem. In [BRS98] werden weitere Äquivalenzrelationen betrachtet, die zusätzlich Ersetzungen der Variablen durch bijektive Abbildungen gestatten. Diese Relationen und die jeweils erlaubten Operation sind:

- Isomorphie: Permutation der Variablen
- Negationsäquivalenz: Negation einzelner Variablen
- Kongruenz: Permutation und Negation
- Lineare Äquivalenz: Lineare Abbildungen

- Affine Äquivalenz: Affine Abbildungen
- Kardinalitätsäquivalenz: Beliebige bijektive Abbildungen

Für die entsprechenden Entscheidungsprobleme wird in [BRS98] gezeigt, dass diese in $\mathbf{\Sigma_2^p}$ liegen, mit Ausnahme der Kardinalitätsäquivalenz jedoch wahrscheinlich nicht in $\mathbf{coNP}$. Sofern die Polynomialzeithierarchie nicht kollabiert, ist keines der Probleme $\mathbf{\Sigma_2^p}$-vollständig [AT96]. Da sich diese Probleme also in der Polynomialzeithierarchie wahrscheinlich zwischen den Klassen $\mathbf{coNP}$ und $\mathbf{\Sigma_2^p}$ befinden, könnten sie eine Trennung der Klassen ermöglichen, oder zumindest Erkenntnisse erbringen, die hierfür hilfreich sind. Zudem ist die Isomorphie von Schaltkreisen ähnlich der Isomorphie von Graphen, die vermutlich zwischen $\mathbf{P}$ und $\mathbf{NP}$ liegt [Sch88, DLN+09], was weitere Erkenntnisse verspricht.

2 Der Postsche Verband

Boolesche Schaltkreise können von Teilschaltkreisen ausgehend induktiv aufgebaut werden. Anschaulich entspricht dies dem Verbinden einzelner Gatter zu einem größeren Schaltkreis. Dies kann durch Hinzufügen fiktiver Variablen, Permutation der Variablen, Gleichsetzung von Variablen und Substitution von Variablen durch Boolesche Funktionen bewerkstelligt werden. Diese vier Operationen werden gemeinsam als *Superposition* bezeichnet [BCRV03, Lau06]. Für eine gegebene Menge Boolescher Funktionen B bezeichnen wir mit $[B]$ die Menge aller Funktionen, die durch wiederholte Anwendung der Superposition aus $B \cup \{\mathrm{id}\}$ gebildet werden können.

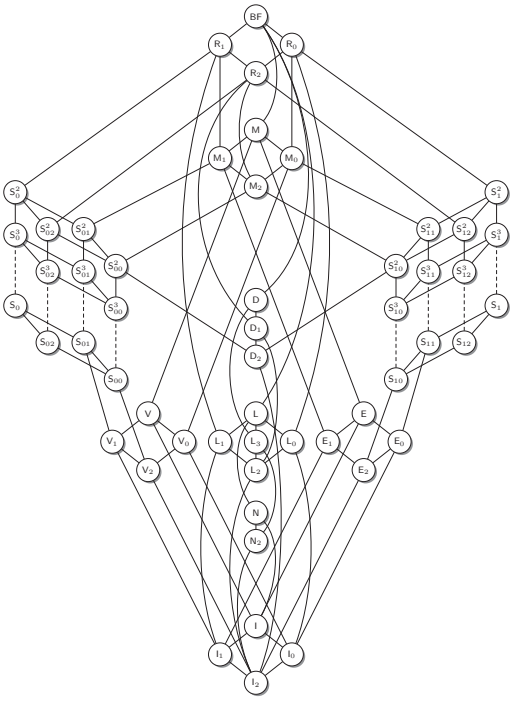


Abbildung 1: Der Postsche Verband

Wie Emil Post 1941 zeigte [Pos41], ist eine Menge der durch Superposition darstellbaren Funktionen abhängig von der Menge der Ausgangsfunktionen. Die Menge aller Booleschen Funktionen hat also mehrere unter Superposition abgeschlossene Teilmengen, die sogenannten *Booleschen Clones*. Gemeinsam bilden diese Clones einen Verband, der als *Postscher Verband* bezeichnet wird (Abbildung 1).

Der Postsche Verband ist anti-isomorph zu dem der Booleschen co-Clones [BCRV04]. Da dieser direkt mit Constraint-Satisfaction-Problemen verbunden ist, könnten die vorgelegten Resultate helfen, z. B. die Ergebnisse aus [BHRV04] zu verallgemeinern. Weitere Informationen und allgemeinere Betrachtungen von Funktionenalgebren über endlichen Mengen finden sich in [Lau06].

3 Äquivalenzrelationen für eingeschränkte Funktionsmengen

Nachdem [BRS98] die Komplexität der in Abschnitt 1 beschriebenen Äquivalenzrelationen für alle möglichen Boolesche Funktionen, die mit der Basis $\{\vee, \wedge, \neg\}$ gebildet werden können, betrachtet hat, bietet es sich an, auch die Komplexität für Teilmengen der Booleschen Funktionen, nämlich den Clones aus dem Postschen Verband, zu untersuchen. Für Äquivalenz und Isomorphie wurde dies in [BCG$^+$12] für sämtliche Clones getan.

In der hier vorgestellten Master-Thesis wurde für in Schaltkreisdarstellung gegebene Funktionen die Komplexität auch für die übrigen Äquivalenzrelationen und Clones betrachtet. Die untersuchten Probleme konnten dabei alle klassifiziert werden, teils jedoch nur mit einer nicht dichten oberen oder unteren Schranke. Die präsentierten Resultate stehen im Zusammenhang mit einer Vielzahl von Ergebnissen, die von der Struktur des Postschen Verbandes abhängen, siehe z. B. [Tho12a, Tho12b, Vol09].

Von den erzielten Resultaten soll hier die Kardinalitätsäquivalenz hervorgehoben werden, die sich deutlich anders verhält als die übrigen Äquivalenzrelationen. Dies zeigt sich bereits bei den Ergebnissen für beliebige Boolesche Funktionen in [BRS98]; von der klassischen Äquivalenz abgesehen ist die Kardinalitätsäquivalenz das einzige Problem, von dem bekannt ist, dass es für die Basis $\{\vee, \wedge, \neg\}$ für eine zentrale Komplexitätsklasse vollständig ist. Zudem liegen Isomorphie, Negationsäquivalenz, Kongruenz, lineare Äquivalenz und affine Äquivalenz alle zwischen $\mathbf{coNP}$ und $\mathbf{\Sigma_2^p}$, während die Kardinalitätsäquivalenz für die Klasse $\mathbf{PP}$ vollständig ist, deren Lage relativ zu $\mathbf{\Sigma_2^p}$ bislang offen ist.

In den meisten Fällen verhält sich die Kardinalitätsäquivalenz wie die anderen betrachteten Relationen: Für separierende und monotone Funktionen ist sie $\mathbf{coNP}$-hart, für lineare Funktionen sowie Konjunktionen und Disjunktionen liegt sie in $\oplus\mathbf{L}$ oder $\mathbf{NL}$. Während aber die übrigen Äquivalenzrelationen auch für selbstduale Funktionen $\mathbf{coNP}$-hart sind, ist die Kardinalitätsäquivalenz trivial entscheidbar: Zwei beliebige n-stellige selbstduale Funktionen haben immer dieselbe Anzahl erfüllender Belegungen, nämlich 2^{n-1}.

Es lässt sich vermuten, dass eine Trennung zwischen der Kardinalitätsäquivalenz und den übrigen Relationen in verschiedene Komplexitätsklassen eintritt, wenn man bessere untere Schranken als $\mathbf{coNP}$ findet. Falls sich Äquivalenzrelationen definieren lassen, die zwischen affiner Äquivalenz und Kardinalitätsäquivalenz liegen, also mehr als die affinen, aber nicht alle bijektiven Abbildungen zulassen, ließen sich eventuell ebenfalls weitere Erkenntnisse über den Unterschied zwischen $\mathbf{\Sigma_2^p}$ und $\mathbf{PP}$ erzielen. Dies spricht dafür, dass eine weitere Untersuchung der Kardinalitätsäquivalenz weitere interessante Resultate erbringen könnte. In der Master-Thesis werden außerdem weitere offene Probleme in diesem Forschungsgebiet aufgezeigt.

Literatur

[AT96] Manindra Agrawal und Thomas Thierauf. The Boolean Isomorphism Problem. In *37th Annual Symposium on Foundations of Computer Science*, Seiten 422–430, 1996.

[BCG+12] Elmar Böhler, Nadia Creignou, Matthias Galota, Steffen Reith, Henning Schnoor und Heribert Vollmer. Complexity classifications for different equivalence and audit problems for Boolean circuits. *Logical Methods in Computer Science*, 8(3), 2012.

[BCRV03] Elmar Böhler, Nadia Creignou, Steffen Reith und Heribert Vollmer. Playing with Boolean Blocks, Part I: Post's Lattice with Applications to Complexity Theory (Complexity Theory Column 42). *SIGACT News*, 34(4):38–52, 2003.

[BCRV04] Elmar Böhler, Nadia Creignou, Steffen Reith und Heribert Vollmer. Playing with Boolean Blocks, Part II: Constraint Satisfaction Problems (Complexity Theory Column 43). *SIGACT News*, 35(1):22–35, 2004.

[BHRV04] Elmar Böhler, Edith Hemaspaandra, Steffen Reith und Heribert Vollmer. The Complexity of Boolean Constraint Isomorphism. In *21st Symposium on Theoretical Aspects of Computer Science*, Seiten 164–175, 2004.

[BRS98] B. Borchert, D. Ranjan und F. Stephan. On the Computational Complexity of Some Classical Equivalence Relations on Boolean Functions. *Theory of Computing Systems*, 31:679–693, 1998.

[DLN+09] S. Datta, N. Limaye, P. Nimbhorkar, T. Thierauf und F. Wagner. Planar Graph Isomorphism is in Log-Space. In *24th IEEE Conference on Computational Complexity*, Seiten 203–214, 2009.

[Lau06] Dietlinde Lau. *Function Algebras on Finite Sets*. Springer, 2006.

[Pos41] Emil Leon Post. The Two-Valued Iterative Systems of Mathematical Logic. *Annals of Mathematics Studies*, 5, 1941.

[Sch88] Uwe Schöning. Graph Isomorphism is in the Low Hierarchy. *Journal of Computer and System Sciences*, 37(3):312–323, 1988.

[Thi00] Thomas Thierauf. *The Computational Complexity of Equivalence and Isomorphism Problems*. Lecture Notes in Computer Science. Springer, 2000.

[Tho12a] Michael Thomas. The Complexity of Circumscriptive Inference in Post's Lattice. *Theory of Computing Systems*, 50(3):401–419, 2012.

[Tho12b] Michael Thomas. On the applicability of Post's lattice. *Information Processing Letters*, 112(10):386–391, 2012.

[Vol09] Heribert Vollmer. The Complexity of Deciding if a Boolean Function can be Computed by Circuits over a Restricted Basis. *Theory of Computing Systems*, 44(1):82–90, 2009.

Improving Hand Pose Estimation by Combining Principal Component Analysis with Biased Particle Swarms

Dennis Hamester
University of Hamburg, Department of Informatics, Knowledge Technology
Vogt-Kölln-Straße 30, D - 22527 Hamburg, Germany
dennis.hamester@gmail.com
http://www.informatik.uni-hamburg.de/WTM/

Abstract: Hand pose estimation is the task of deriving a hand's articulation from sensory input, here depth images in particular. A novel approach states pose estimation as an optimization problem: a high-dimensional hypothesis space is constructed from a hand model, in which particle swarms search for the best pose hypothesis. We propose various additions to this approach. We use principal component analysis (PCA) to measure eigenvectors of hand-finger motion. When this information is combined with a certain biased particle swarm optimization (PSO) variant, accuracy and performance of hand pose estimation increase significantly. Several experiments were performed to measure the improvements of our method.

1 Introduction

The human hand is highly articulated. Humans use hands to manipulate objects in their surroundings and to communicate with other people. Capturing exact hand postures is an important step for Human-Robot Interaction and the development of natural interfaces. Computer vision (CV) can provide cheap and unobtrusive solutions to this problem, especially compared to data gloves.

Solving CV-based hand pose estimation without markers in single camera setups is a very challenging task, because hands can take on vastly different shapes in images. The amount of degrees of freedom (DOFs) contributes to a high-dimensional problem. The problem is further complicated by self-occlusions of the hand, that happen inevitably during the projection onto 2D images.

Significant progress in this area was made by Oikonomidis et al. [OKA11]. Their method deals very well with the high-dimensionality and self-occlusions of the human hand. However, their approach is still computationally demanding. They report that their algorithm can run at about 15 FPS on a high-end PC. This is only half the rate at which the Kinect provides images. Our goal was to improve the performance, possibly to the point of running in real-time. At the same time we did not want to sacrifice any accuracy. We addressed this by exploiting biases in certain variants of particle swarms. We will show, that the optimization behavior of these variants can be aligned with a priori knowledge about how humans perform hand motions. The result was an overall improved convergence behavior, leading to better pose estimation in less time.

The idea to use a priori information has already been applied successfully to hand pose estimation by Bianchi et al. [BSB13]. They determined statistical properties of hand motion

and used these to improve the noisy measurements of a low-cost data glove. Our method differs in the way a priori knowledge is used. We use it to transform the search space of all hand postures, such that certain variants of particle swarm optimization (PSO) perform better due to biases in their behavior. We also do not require an existing pose estimation.

2 Hand Pose Estimation

Hand pose estimation is approached here as an optimization problem. Given a hypothesis $h \in \mathbb{R}^n$, which describes a specific articulation, a target function determines how closely h matches the observation depth image d_o from a Kinect:

$$f(h) = \sum_{v=1}^{y} \sum_{u=1}^{x} \min\left(\left|d_o(u,v) - d_h(u,v)\right|, t\right) \tag{1}$$

The target function f iterates over d_o and computes the sum of pixel-wise differences to a synthetically rendered depth image d_h of a hand according to the hypothesis h. Optimization is defined by finding a hypothesis h which minimizes $f(h)$. In comparison to Oikonomidis et al. [OKA11], we transform the search space $\mathbb{R}^n$ linearly by a change of basis discussed in the next section. The actual optimization of $f(h)$ is performed by a particle swarm.

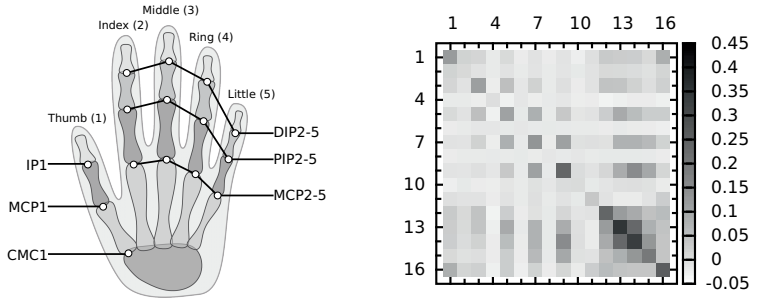
3 Eigenvectors of Hand-Finger Articulations

Our hand model consists of two parts: its shape and a set of joints. The shape defines the model's visual appearance and is used to render artificial images of hand articulations. We placed a set of 16 joints into the model, each with either one or two DOFs. Figure 1(a) shows a schematic joint model with all joints considered here. Furthermore, the hand's position and orientation in space are also considered and add six additional DOFs, leading to a total of 26 DOFs.

The covariance matrix in fig. 1(b) has been estimated with PCA from a dataset of more than 9200 random hand postures. The hand's position, orientation, and DIP joints were stripped from the resulting dataset and not considered for the PCA. Estimating DIP angles correctly is very hard due to their small impact in images. Instead, we reconstructed their values using a common relationship to PIP joints: $\theta_{DIP} = \frac{2}{3}\theta_{PIP}$ [LWH00]. As part of the PCA, we obtained the eigenvectors which are used as a new basis for the space of hand postures during optimization.

4 Particle Swarm Optimization

Many different variants of PSO are described in the literature [PCW12], several of which are subject to biases [SGS10]. These biases tend to push particles onto axes parallels, a circumstance exploited deliberately here. Due to the change of basis to eigenvectors, most significant and relevant changes happen along axes parallels instead of diagonals.



(a) Joint model (b) Covariance matrix

Figure 1: (a) Schematic joint model of a human hand. Joints are shown as small white dots. The CMC1 and MCP2–5 are modelled as 2-DOF joints. The other joints (MCP1, IP1, PIP2–5, DIP2–5) have only one DOF. (b) Covariance matrix of random hand poses. The DOFs are: 1,2=CMC1; 3,4=MCP2; 5,6=MCP3; 7,8=MCP4; 9,10=MCP5; 11=MCP1; 12=PIP2; 13=PIP3; 14=PIP4; 15=PIP5; 16=IP. Position, orientation and DIP joints are not included.

The variant used here is called the *Bratton2007* version [PCW12] and defined by the following particle velocity update:

$$v_i \leftarrow \chi \left[v_i + U(0, \phi_c) \otimes (p_{c,i} - p_i) + U(0, \phi_s) \otimes (p_s - p_i) \right] \tag{2}$$

$U(a, b)$ is a vector of d random numbers, each uniformly distributed in the range given by its parameters and $\otimes$ denotes component-wise multiplication. The parameters ϕ_c and ϕ_s control the influence of the cognitive ($p_{c,i}$) and social (p_s) component. The parameter χ is used to avoid swarm-explosion. It can be computed as follows [CK02]:

$$\phi = \phi_c + \phi_s > 4 \qquad \chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \tag{3}$$

5 Experiment

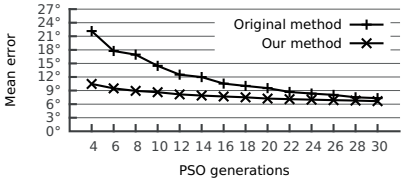
Our goal for the experiments and evaluations was to assess the differences of our pose estimation compared to Oikonomidis et al. [OKA11]. We were primarily interested in measuring the possible accuracy gains through quantitative evaluation. For experiments, we generated a test video from a sequence of hand postures which defined the ground-truth.

Error measurement are based on angle comparisons. Let $\pi_i(x)$ be the projection of the vector x onto its i-th component and $x_1, x_2 \in \mathbb{R}^{26}$ be hand postures. Then

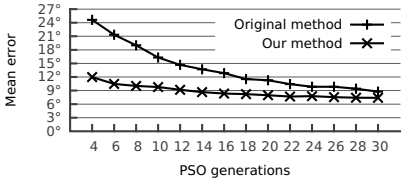
$$e_a(x_1, x_2) = \frac{1}{23} \sum_{i=4}^{26} |\pi_i(x_1 - x_2)| \tag{4}$$

measures the discrepancy of all angles as the mean absolute difference. The first three components correspond to the hand location in 3D space, which was not considered for evaluation.

The vertical axis in fig. 2 shows the mean absolute angle error e_a. A single measurement is the mean error over all frames in the entire video for the given PSO parameters. The



(a) 64 particles



(b) 32 particles

Figure 2: Comparison of our method to the original method [OKA11] without change of basis to eigenvectors and biased PSO.

original method shows a strong dependency on the number of generations. To keep the error below an average of 9° at least 64 particles and 22 generations had to be used. For our method on the other hand, 32 particles and 14 generations already were sufficient. In general, we observed much faster convergence after enabling the PCA and biased PSO. The curves for our method are less steep in fig. 2. This directly translates to an improved performance, because less effort is required to achieve a certain maximum error. Using 64 particles and 25 generations has been suggested before [OKA11]. We reached the same error at 32/18, which is roughly 2.8 times faster.

6 Conclusion

Our method maintains the same level of accuracy as before [OKA11], but is about 2.8 times faster. If a 16% increase in estimation errors is acceptable, our method is able to run five times faster. We expect that even more performance is achievable with our method when the set of possible hand postures is constrained by specific applications. Our idea to combine biased PSO with a change of basis provides a very flexible way of incorporating a priori knowledge.

References

[BSB13] M. Bianchi, P. Salaris, and A. Bicchi. Synergy-based hand pose sensing: Reconstruction enhancement. *Int. Journal of Robotics Research, The*, 32(4):396–406, 2013.

[CK02] M. Clerc and J. Kennedy. The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. 6(1):58–73, 2002.

[LWH00] J. Lin, Y. Wu, and T. S. Huang. Modeling the Constraints of Human Hand Motion. In *Human Motion, Proc. Workshop on*, pages 121–126, 2000.

[OKA11] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Efficient Model-based 3D Tracking of Hand Articulations using Kinect. In *British Machine Vision Conference, Proc. of the*, pages 101.1–101.11, 2011.

[PCW12] S. S. Pace, A. Cain, and C. J. Woodward. A Consolidated Model of Particle Swarm Optimisation Variants. In *Evolutionary Computation, IEEE Congress on*, pages 1–8, 2012.

[SGS10] W. M. Spears, D. Green, and D. F. Spears. Biases in Particle Swarm Optimization. *Int. Journal of Swarm Intelligence Research*, 1(2):34–57, 2010.

Thread Block Lock Free Format für dünnbesetzte Matrix-Vektor-Produkte auf Grafikkarten

Markus Mieth, Ralf Seidler, H. Martin Bücker

markus.mieth@mailbox.tu-dresden.de, {ralf.seidler, martin.buecker}@uni-jena.de

Abstract: Diese Arbeit stellt das neues Datenformat *Thread Block Lock Free* (TBLF) für die effiziente Berechnung von Matrix-Vektor-Produkten für dünnbesetzte Matrizen vor. Wir zeigen die Ergebnisse der Implementierung des neuen Datenformats TBLF auf Grafikkarten mittels CUDA C und vergleichen diese mit einer aus der Literatur bekannten Implementierung des *Compressed Sparse Row* (CSR) Formats. Bei der Verwendung von zufälligen dünnbesetzen Matrizen mit einem maximalen Besetztheitsgrad von 10% wird ein Speed-Up gegenüber CSR von bis zu 6 erreicht.

1 Einleitung

Gegeben sei eine dünnbesetzte, beliebige Matrix $A \in \mathbb{R}^{n \times n}$, deren Struktur bekannt ist, und ein Vektor $z \in \mathbb{R}^n$. Gesucht ist das Matrix-Vektor-Produkt $y = A \cdot z \in \mathbb{R}^n$. Diese wichtige Basisoperation der linearen Algebra soll mit Hilfe einer Grafikkarte berechnet werden, welche die *Compute Unified Device Architecture* (CUDA) unterstützt [NVI12a].

Eine gängige Möglichkeit zur Berechnung des Matrix-Vektor-Produktes besteht darin, die Matrix A in gleichgroße, quadratische Teilmatrizen so aufzuteilen, dass ein CUDA-Block eine Teilmatrix bearbeitet. Jeder CUDA-Block besteht aus t CUDA-Threads. Jedem CUDA-Thread wird genau eine Spalte aus der Teilmatrix und genau das dazugehörige Vektorelement aus z zugeordnet. Die Threads multiplizieren jeweils ihre Matrixelemente mit ihrem Vektorelement, indem sie pro Addition ein (lokales) Register für z_i verwenden und aus dem globalen Speicher ein Matrixelement a_{ij} laden und danach deren Produkt zu dem zugehörigen Eintrag des *Shared Memory* addieren. Zum Schluß speichert der *Shared Memory* die Teilergebnisse der Threads, die durch eine parallele Reduktion der Zeilen innerhalb einer Teilmatrix zusammengefasst werden. Schließlich addiert jeder Thread ein Element aus dem *Shared Memory* per *atomicAdd* zu der dazugehörigen Zeile im Lösungsvektor.

Eine Schwachstelle dieser spaltenweisen Abarbeitung stellt die parallele Reduktion dar. Sie bedeutet pro CUDA-Block einen zusätzlichen Kommunikationsaufwand. Dieses Problem kann gelöst werden, indem die zu bearbeitende Matrix in einem Datenformat abgespeichert wird, welches garantiert, dass pro Schreibvorgang auf dem Lösungsvektor keine zwei CUDA-Threads gleichzeitig auf die selbe Speicherzelle y_i im

Shared Memory zugreifen müssen. Diese Eigenschaft wurde mit dem neuen Datenformat *Thread Block Lock Free* (TBLF) realisiert, welches im Folgenden vorgestellt werden soll. In [Mie13] wird dieses neue Format detaillierter vorgestellt.

2 Thread Block Lock Free Format

Das Datenformat TBLF ist für die dünnbesetzte Matrix A aus Gleichung (1) in Tabelle 1 beispielhaft angegeben. Die Nichtnulleinträge aus A werden im Array `data` Thread-Blockweise hintereinander abgelegt. Die zugehörige Zeilennummer innerhalb des Blockes wird im Array `ptr` gespeichert. Um den gegenseitigen Ausschluss sicherzustellen, wurde ein neutrales Element eingeführt. Dieses Element bezeichnet die Anzahl der gestarteten Threads pro Thread-Block, `blockDim`. Im Beispiel hat das neutrale Element den Wert 2 und wird an den Stellen `ptr`$[2,5,10]$ angenommen. Das Array `thread_blocks` beinhaltet Beginn- und End-Markierungen in der linearen Indizierung der Felder `data` und `ptr` des jeweiligen Thread-Blockes.

$$A = \left(\begin{array}{cc|cc} a_0 & b_0 & 0 & 0 \\ 0 & b_1 & c_1 & 0 \\ \hline a_2 & 0 & c_2 & d_2 \\ 0 & b_3 & 0 & d_3 \end{array}\right) \quad (1)$$

$$\begin{array}{rc|cccc|cc|cccc|cc|} \texttt{data} = & [& a_0 & b_1 & 0 & b_0 & c_1 & 0 & a_2 & b_3 & c_2 & d_3 & 0 & d_2 & &] \\ \texttt{ptr} = & [& 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 0 & 1 & 2 & 0 & &] \\ \texttt{thread\_blocks} = & [& 0 & & & & 4 & & 6 & & 8 & & & & 12 &] \end{array}$$

Tabelle 1: TBLF Format der Matrix A aus (1)

Mit dieser Datenstruktur wird das Matrix-Vektor-Produkt aus Algorithmus 1 wie folgt ausgeführt. Innerhalb eines Thread-Blockes führt jeder Thread k Iterationen in der For-Schleife aus. Der initiale Index der Arrays `data` und `ptr` für einen Thread ergibt sich aus dem Wert von `thread_blocks[my_blockId]+threadId`. Nach einem Schritt addiert er zu seinem aktuellen Index den Wert von `blockDim`, um zu seinem nächsten Element zu kommen. Die Ergebnisse eines Schrittes werden in den dazu angelegten *Shared Memory* geschrieben. Wenn anschließend alle Schritte abgearbeitet wurden, addieren die Threads diese $\lceil n/blockDim \rceil$ Elemente aus dem *Shared Memory* zum Lösungsvektor im globalen Speicher. Das neutrale Element stellt sicher, dass die ausmaskierten Threads auf ein Element im *Shared Memory* schreiben, welches für das Ergebnis nicht von Bedeutung ist. Des Weiteren ist im Programm keine unnötige Verzweigungen vorhanden, was den Durchsatz weiter erhöht. Um dies zu realisieren, werden für einen CUDA-Block $\lceil n/blockDim \rceil + 1$ Elemente im *Shared Memory* angelegt. Tabelle 2 zeigt eine mögliche parallele Abarbeitung des Algorithmus.

Durch die Aufteilung der Matrix in mehrere Teilmatrizen ist eine parallel Abarbeitung auf mehreren CUDA-Blöcken gegeben. Sie ist aber nur sinnvoll, wenn die Schrittanzahl in jedem Thread-Block minimal ist. Die minimale Anzahl an k Schritten ist schon von Anfang an ersichtlich, da diese der maximalen Anzahl an Nichtnulleinträgen pro Zeile oder Spalte eines Matrix-Blockes entspricht. Für das Finden der Schritte stellte sich heraus, dass dies ein NP-schweres Problem ist, für dessen Lösung wir eine neue Heuristik namens *kSearch* entwickelt haben [Mie13].

Algorithmus 1 : Dünnbesetztes Matrix-Vektor-Produkt im TBLF Format mit CUDA

Input : $n \times n$ matrix A, stored in TBLF format as: (data, ptr, thread_blocks), CUDA-specific variable to identify a thread and block, number of threads per block and number of blocks launched (threadId, blockId, blockDim, gridDim), input vector z of size n

Output : $y \leftarrow A \cdot z$

1 init shared memory array $shared[threadId] \leftarrow 0$;
2 $my\_blockId \leftarrow blockId \cdot gridDim$;
3 $my\_z \leftarrow z[blockId \cdot blockDim + threadId]$;
4 sync threads in block;
5 **for** $i \leftarrow thread\_blocks[my\_blockId] + threadId$ **to** $thread\_blocks[my\_blockId + 1]$ **in** $blockDim$ **steps do**
6 $\quad shared[ptr[i]] += data[i] \cdot my\_z$;
7 $\quad$ sync threads in block;
8 `atomicAdd(`$y[blockId \cdot blockDim + threadId], shared[threadId]$`);`

	Thread-Block 0	Thread-Block 1	Thread-Block 2	Thread-Block 3
data:	a_0 b_1 0 b_0	c_1 0	a_2 b_3	c_2 d_3 0 d_2
ptr:	0 1 2 0	1 2	0 1	0 1 2 0
Schritt 0:	$a_0 \cdot z_0$ $b_1 \cdot z_1$	$c_1 \cdot z_2$ $0 \cdot z_3$	$a_2 \cdot z_0$ $b_3 \cdot z_1$	$c_2 \cdot z_2$ $d_3 \cdot z_3$
Schritt 1:	$0 \cdot z_0$ $b_0 \cdot z_1$	atomicAdd: $y_0^r += y_0^1$, $y_1^r += y_1^1$	atomicAdd: $y_2^r += y_0^2$, $y_3^r += y_1^2$	$0 \cdot z_2$ $d_2 \cdot z_3$
Schritt 2:	atomicAdd: $y_0^r += y_0^0$, $y_1^r += y_1^0$			atomicAdd: $y_2^r += y_0^3$, $y_3^r += y_1^3$

Tabelle 2: Beispiel einer parallelen Ausführung mit dem TBLF Format

3 Performance-Ergebnisse

Die Messungen wurden auf einer Nvidia Kepler K20 GPU [NVI12b] auf einem System mit zwei Intel(R) Xeon(R) X5650 CPUs [Int] ausgeführt. Als Testmatrizen wurden mittels *scipy.sparse* [Oli07] zufällige, dünnbesetzte quadratische Matrizen der Ordnung 2^{12} und 2^{13} in doppelter Genauigkeit mit unterschiedlichen Besetztheitsgraden von 1, 2 bis 10 % erzeugt. Als Vergleich dient die Implementierung des CSR Formates aus [BG08]. In Abbildung 1 sind die Ergebnisse in GFlops für eine Thread-Blockgröße von 1024 gezeigt. Hierbei wurden die Messungen sieben mal wiederholt und das arithmetische Mittel der Ergebnisse gebildet. Die Abbildung zeigt, dass das TBLF Datenformat mit steigendem Besetztheitsgrad an Geschwindigkeit gewinnt. Das TBLF Format hat bei 10 % mit 9.4 GFLops sein Maximum erreicht. Im Vergleich dazu liegt das CSR Format bei 1.6 GFlops. Dies ist ein Speed-Up von rund 6.

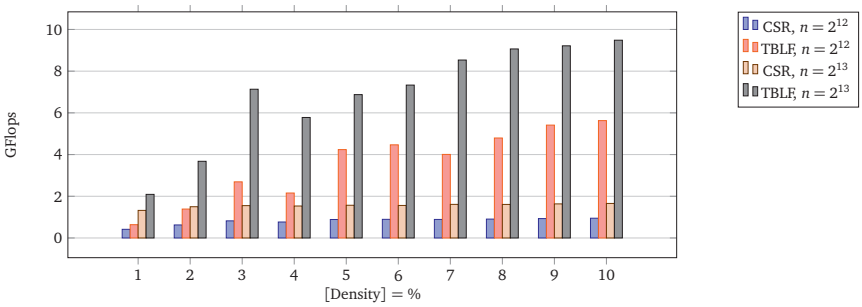


Abbildung 1: Performance-Ergebnisse für TBLF mit 1024 CUDA-Threads

4 Zusammenfassung

In dieser Arbeit wurde das neue TBLF Format eingeführt. Beim Entwurf dieser Datenstruktur zur Berechnung von Matrix-Vektor-Produkten auf Grafikkarten wurde besonderer Wert auf die Vermeidung von Synchronisationskonflikten auf dem *Shared Memory* gelegt. Im Gegensatz zu traditionellen seriellen Datenstrukturen werden im TBLF Format Multiplikationen mit wenigen Nullen zugelassen. Die Performance-Ergebnisse im Vergleich zum CSR Format mit zufälligen, dünnbesetzten Matrizen zeigen, dass diese Vorgehensweise einen Geschwindigkeitsvorteil von bis zu 6 liefert.

Literatur

[BG08] Nathan Bell and Michael Garland. Efficient Sparse Matrix-Vector Multiplication on CUDA. NVIDIA Technical Report NVR-2008-004, NVIDIA Corporation, 2008.

[Int] Intel. Intel(R) Xeon(R) Processor X5650 (12M Cache, 2.66 GHz, 6.40 GT/s Intel(R) QPI).

[Mie13] Markus Mieth. Modellierung und Implementierung dünnbesetzter Matrix-Vektor-Produkte auf Grafikkarten. Bachelorarbeit, Institut für Informatik, Friedrich-Schiller-Universität Jena, 2013.

[NVI12a] NVIDIA Corporation. NVIDIA CUDA C Programming Guide, 2012.

[NVI12b] NVIDIA Corporation. Whitepaper: NVIDIA's Next Generation CUDA Compute Architecture: Kepler GK110, 2012.

[Oli07] Travis E. Oliphant. Python for Scientific Computing. *Computing in Science and Engineering*, 9(3):10–20, 2007.

Konvergenznachweis von asynchronen Algorithmen

Benjamin Saul
Martin-Luther-Universität Halle-Wittenberg
Betreuer: Wolf Zimmermann

Abstract: In dieser Arbeit wurden Kriterien untersucht, die für die korrekte asynchrone Ausführung von Algorithmen notwendig sind. Insbesondere erfolgt die Betrachtung von Fixpunktiterationen. Asynchron heißt in diesem Falle, dass die beteiligten Prozessoren nicht auf die Ergebnisse der anderen warten müssen. Als Hauptkriterium wird die Monotonie von Funktionen über vollständige Halbordnungen verwendet. Ist diese gegeben, dann konvergieren synchrone und asynchrone Ausführung gegen den gleichen eindeutigen Fixpunkt. Untersucht wurde, ob und wie dieses Kriterium auf verschiedene Probleme angewendet werden kann.

1 Motivation asynchroner Algorithmen

Bei einer synchronen Fixpunktiteration lesen die Prozessoren gleichzeitig den gesamten Datenspeicher, berechnen ihr Teilergebnis und schreiben dieses im Anschluss wieder in den Speicher zurück. Auf den Start der nächsten Iteration wird solange gewartet, bis alle Prozessoren ihr Ergebnis geschrieben haben. Bei der asynchronen Ausführung können die Prozessoren unverzüglich mit der nächsten Iteration beginnen (Abb. 1). Dafür wird dann mit den gerade vorhandenen und eventuell veralteten Daten gerechnet. Wird eine solche Iteration in einer Mehrkernumgebung ausgeführt, kann also auf die Cache-Kohärenz verzichtet werden.

Die Korrektheit des Ergebnisses der Berechnung kann formal für viele Anwendungen gezeigt werden. Eine bisher hauptsächlich für Methoden der Programmanalyse verwendete Nachweisform über vollständige Halbordnungen soll hier verwendet werden, um diese Korrektheit nachzuweisen. Die Hoffnung besteht, dass diese Kriterien leichter nachgewiesen werden können als vergleichbare Kriterien von bspw. Frommer et al. [FS00].

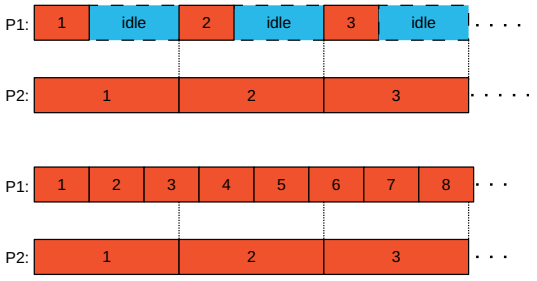


Abbildung 1: Vergleich einer synchronen (oben) und einer asynchronen Iteration (unten).

2 Theoretische Grundlagen des Konvergenznachweises

Grundlagen über die Theorie asynchroner Fixpunktiterationen sind in [BT97] beschrieben. Diese sind hier nicht für das weitere Verständnis notwendig. Der hier verwendete Ansatz für den Konvergenzbeweis stammt von Cousot [Cou77] und verwendet vollständige Halbordnungen.

Definition 1 (Vollständige Halbordnung, Kette)**.** *Ein Tupel $(X, \sqsubseteq)$ heißt Halbordnung, gdw. $\sqsubseteq$ ein reflexiver, transitiver und antisymmetrischer zweistelliger Operator über der Menge X ist. Eine total geordnete Teilmenge $K \subseteq X$ heißt Kette. Eine Halbordnung heißt vollständig, gdw. es ein kleinstes Element $\bot \in X$ (d.h. $\bot \sqsubseteq x$ für alle $x \in X$) gibt und jede Kette K eine eindeutig bestimmte kleinste obere Schranke $\bigsqcup K$ besitzt.*

Die Iterationen der einzelnen Prozessoren werden als Iterationsfunktionen $f_i : X_i \to X_i$ für $i \in \{1, \ldots, n\}$ bezeichnet. Die Mengen X_i bilden zusammen den Datenbereich des Programmes. Folgende Kriterien für alle $i \in \{1, \ldots, n\}$ wurden untersucht:

1. $(X_i, \sqsubseteq_i)$ sind endliche vollständige Halbordnungen.
2. Aus $x \sqsubseteq y$ folgt $f_i(x) \sqsubseteq_i f_i(y)$ für alle $x, y \in X$ (Monotonie)
3. Es gilt $x^0 \sqsubseteq f_i(x^0)$ (Präfixpunkt, bspw. $x^0 = \bot$).

Dabei ist $(X_1 \times \ldots \times X_n, \sqsubseteq)$ punktweise über den Relationen aus $(X_i, \sqsubseteq_i)$ definiert, analoges gilt für f. Mit diesen Kriterien gilt nach dem Fixpunktsatz von Knaster-Tarski, dass $\lim_{t\to\infty} f^t(x^0) = x^*$ der eindeutig bestimmte kleinste Fixpunkt von f ist. Für die asynchrone Iteration wird diese Gleichung verändert, sodass auch veraltete Daten gelesen werden können. Cousot zeigte, dass diese Kriterien ausreichen, um die Existenz von x^* und die Konvergenz der Iteration auch asynchron zu bedingen. Der Satz wurde in dieser Arbeit auch auf nicht endliche Grundmengen ausgeweitet.

Gegenstand meiner Untersuchungen war es nun herauszufinden, ob und wie gut man diese Halbordnungen auf den Grundmengen der Algorithmen finden kann bzw. wie gut Monotonieeigenschaften der Iterationen nachzurechnen sind. In diesem Artikel werden zwei Algorithmen betrachtet: Kürzeste-Wege-Probleme mit Bellman-Ford, bei denen sich die Kriterien schnell nachweisen lassen, und das Listenrang-Problem, welches mit den hier vorgestellten Kriterien nicht betrachtet werden kann.

3 Asynchrone Konvergenz beim Kürzeste-Wege-Problem

Eine elegante Art, die kürzesten Wege zu einem gegebenen Zielknoten zu bestimmen, ist der Algorithmus von Bellman-Ford [BT97]. Jeder Knoten speichert sich den aktuellen kürzesten Weg von sich selbst aus betrachtet. Mit jeder Iteration wird ein kürzerer Weg über einen Nachbarknoten gesucht und die eigene Entfernung nach Bedarf aktualisiert. Für die Parallelisierung ist es ohne Probleme möglich, die Berechnung einzelner Knoten des Graphen auf verschiedene Prozessoren zu verteilen. Die Iterationsvorschrift in einem Knoten i mit Distanz d_i (Distanzvektor d) zum Knoten 0, Nachbarschaftsbeziehung $N[i]$ und Kantengewichten e_{ij} ergibt sich also mit

$$f_i(d) = \min_{j \in N[i]} e_{ij} + d_j.$$

Initialisiert werden die Abstände mit ∞. Die Grundmenge des Algorithmus lässt sich als endliche Teilmenge von $\mathbb{Q}$ auffassen, da nur Kombinationen von Distanzen auftreten. Damit kann man die durch $\mathbb{Q}$ gegebene Halbordnungsrelation verwenden und identifiziert $\sqsubseteq$ mit $\geq$ als Vergleichsoperator. Aus der $\min$-Funktion mit $x \sqsubseteq y$ ergibt sich

$$\begin{aligned} f_i(x) &= \min_{j \in N[i]} e_{ij} + x_j \\ &\sqsubseteq \min_{j \in N[i]} e_{ij} + y_j, && \text{da } x_i \geq y_i \\ &= f_i(y), && \text{Definition von } f. \end{aligned}$$

Da dies für alle Knoten i erfüllt ist, folgt daraus die Monotonie. Die Präfixpunkteigenschaft von ∞ gilt nach Definition, da $x \leq \infty$ für alle $x \in \mathbb{Q}$ erfüllt ist. Die Kriterien von Cousot sind also erfüllt und der Algorithmus wird auch asynchron korrekt ausgeführt. Der Beweis lässt sich auch auf die berechneten Pfade erweitern. Ähnliche Algorithmen, welche auch iterativ das Minimum oder Maximum über den Nachbarschaftsknoten berechnen, können auf gleiche Weise behandelt werden.

4 Asynchrone Konvergenz beim Listenrang mit Pointerjumping

Ein weiterer betrachteter paralleler Algorithmus ist die Bestimmung eines Listenranges mit Pointerjumping [CLR08]. Bei gegebener Liste soll die Entfernung eines jeden Listenelementes zum Listenende bestimmt werden. Jedem Knoten der Liste wird ein Prozess zugeordnet. Die Zeiger und Abstände der einzelnen Elemente seien in den beiden Vektoren $next$ und $dist$ abgespeichert. Die Abstände zum Listenende werden mit 1 initialisiert. Anschließend wird schrittweise der Abstand des nachfolgenden Elementes aufsummiert und der Zeiger auf das nachfolgende Listenelement weiter gerückt. Jeder Prozessor i berechnet, solange $next[i]$ nicht auf das Listenende zeigt, in jedem Schritt folgende Funktionen

$$dist[i] := dist[i] + dist[next[i]] \qquad \text{und} \qquad next[i] := next[next[i]].$$

Zeigen alle Einträge von $next$ auf das Listenende, so ist der Algorithmus beendet und die korrekten Abstände stehen in $dist$. Bei der Untersuchung der asynchronen Konvergenz gibt es aber ein Problem: es ist nicht möglich, die Monotonie dieser Funktionen auf irgendeiner vollständigen Halbordnung nachzuweisen. Es kann keine feste Ordnung auf den Elementen geben, die dies ermöglicht. Die Positionen der Listenelemente relativ zueinander ändern sich durch das Pointerjumping, was sich als Überholen von Zeigern (siehe Abbildung 2) zeigt. Es ist damit nicht möglich, mit den Kriterien von Cousot eine asynchrone Konvergenz zu zeigen. Dies zeigt, dass selbst ein anscheinlich monotones Problem, bei dem die Listenzeiger immer weiter ans Ende rücken, nicht zwangsläufig mit den hier vorgestellten Kriterien untersucht werden kann. Allerdings gibt es andere Kriterien, wie in [BT97], deren Nachweis hier möglich ist und die asynchrone Konvergenz garantieren.

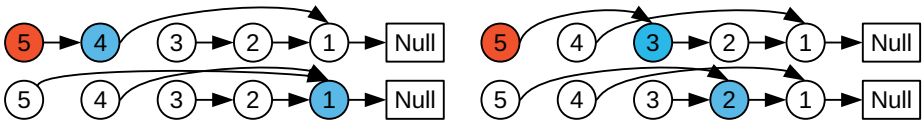


Abbildung 2: Beispielanwendung von zwei Pointerjumping-Operationen.

5 Zusammenfassung und Anwendungen

Bei der asynchronen Ausführung von Algorithmen entfallen die kommunikationsbedingten Wartezeiten bzw. Cache-Kohärenzen von Mehrkernarchitekturen. Ohne die durch Synchronisierung bedingten Wartezeiten können die beteiligten Prozessoren ohne Aufwand besser ausgelastet werden. Außerdem ergeben sich Vorteile bei der Berechnung. Bei einigen Algorithmen, wie Kürzeste-Wege-Problemen, können sich die Eingangsdaten während der Laufzeit verändern. So lassen sich z. B. Routing-Algorithmen ableiten, die auf Ausfälle und Neuverbindungen reagieren können, ohne den Wegfindungsprozess neu starten zu müssen. Außerdem kann durch die veränderte Reihenfolge der Iterationen der Fixpunkt unter Umständen schneller erreicht werden.

Der Nachweis der korrekten Konvergenz einer asynchronen Fixpunktiteration ist hier über vollständige Halbordnungen nach [Cou77] geführt worden. Die Arbeit von Cousot entstand zeitlich vor verwandten Arbeiten wie [BT97] und beruht auf dem Fixpunktsatz von Knaster-Tarski. Korollare zu Bertsekas Arbeit wurden in [FS00] zusammengefasst. Für Anwendungen aus der Graphentheorie konnte auch dort mit vergleichbarem Aufwand der Konvergenznachweis geführt werden. Außerdem kann oftmals auf bereits vorhandene Halbordnungsrelationen zurückgegriffen werden. Allerdings muss man sich dabei immer bewusst sein, dass es auch Algorithmen wie das Pointerjumping gibt, deren Konvergenz nicht über vollständige Halbordnungen gezeigt werden kann. Der Nachweis hier kann allerdings über allgemeinere Aussagen zur asynchronen Konvergenz aus [BT97] geführt werden.

Automatisierte Nachweise für die Konvergenz sind insgesamt denkbar. Durch die Eigenschaft, dass jede Verkettung von monotonen Funktionen wieder eine monotone Funktion ist, kann sich der geführte Nachweis auf die elementaren Funktionen beschränken. So kann ein Programmierer ein synchrones, leichter verständliches Programm schreiben, welches dann auch asynchron ausgeführt werden kann. Derartige Programmierung wird z. B. in [WXDG13] unterstützt. Die Laufzeit asynchroner Algorithmen hängt sehr stark von der Reihenfolge der Berechnungen ab. Hier kann man durch geschickt gewählte Berechnungsstrategien die gesamte Laufzeit verkürzen. Asynchrone Algorithmen bieten also ein großes Optimierungspotential für parallelisierte Anwendungen.

Literatur

[BT97] Dimitri P. Bertsekas und John Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods (Optimization and Neural Computation)*. Athena Scientific, 1997.

[CLR08] Henri Casanova, Arnaud Legrand und Yves Robert. *Parallel Algorithms (Chapman & Hall/CRC Numerical Analysis and Scientific Computing Series)*. Chapman and Hall/CRC, 2008.

[Cou77] Patrick Cousot. Asynchronous iterative methods for solving a fixed point system of monotone equations in a complete lattice. *Res. rep. RR*, 88, 1977.

[FS00] A. Frommer und D.B. Szyld. On asynchronous iterations. *Journal of computational and applied mathematics*, 123(1):201–216, 2000.

[WXDG13] Guozhang Wang, Wenlei Xie, Alan Demers und Johannes Gehrke. Asynchronous Large-Scale Graph Processing Made Easy, 2013.

Erkennung von Motiv- und Themenvariationen

Jannik Arndt
Universität Oldenburg
jannik.arndt@uni-oldenburg.de

Abstract: Die algorithmische Erkennung von Motiven stellt einen wichtigen Bereich des Music Information Retrieval dar. Da ein zentrales Merkmal von Musik jedoch ist, dass Motive und Themen variiert werden, ist es eine wichtige Aufgabe, diese Variationen erkennen und zuordnen zu können. Dies kann bei regelbasiertem Vorgehen durch eine Kombination eines *Sliding Window* mit einem Entscheidungsbaum gelöst werden, bei komplexeren Veränderungen wird jedoch eine Heuristik benötigt, die menschliche Erkennungsprozesse so modelliert, wie sie in der Musikpsychologie erklärt werden.

1 Einleitung

In den meisten Fällen ist Musik schön, weil sie *nicht* einfach, sondern komplex, nicht repetetiv, sondern abwechslungsreich, nicht einfach zu durchschauen, sondern tiefgründig ist. Das wiederum macht es schwierig, sie mit mathematischen Methoden zu erfassen, die zwar ebenso komplex, abwechslungsreich und tiefgründig sein können, am Ende aber immer logisch sind. Diesen Luxus bietet Musik nicht. Ihre Logik ist abhängig von Epochen, Stilen und Sozialisierungen.

Während viele Konzepte der Musik wie z. B. die Harmonik mit der Zeit einen starken Wandel erfahren haben, hat sich eins relativ konstant gehalten: Die Melodie. Von frühen Werken der Renaissance über klassische und romantische Sinfonien und (die meisten) modernen Werke bis hin zu aktuellen Stücken der Popmusik können wir instinktiv aus einem Haufen von Schall eine Melodie oder eine Variation davon wahrnehmen. Der Gedanke liegt nahe, dass dies etwas sein müsste, was relativ leicht auch algorithmisch umzusetzen ist. Dieses Paper zeigt, dass dies jedoch nur bei der Teilmenge der regelbasierten Variationen der Fall ist.

Algorithmsche Motiv- bzw. Melodieerkennung ist ein Teilbereich des Music Information Retrieval (MIR) [LInR08]. Dieser unterteilt sich in Anwendungen auf Audiodaten, also in der Regel Aufnahmen von Aufführungen, und Anwendungen auf symbolischen Daten, sprich Noten im weitesten Sinne. Diese werden häufig auch als *Standard Music Notation* (SMN) bezeichnet und sind daher symbolisch, weil sie lediglich die Anweisungen, wie etwas zu spielen ist, speichern und wiedergeben.

2 Erkennung bekannter Variationen

Die Musiktheorie bietet eine Auswahl an Vorgehensweisen, um Motive und Themen zu variieren. Die geläufigsten sind in Abb. 1 dargestellt, wobei a) das Ausgangsmotiv aus Mozarts *Sonata facile* (KV 545) ist. Dies sind:

- Transposition, mit den Unterscheidungen
  1. im Kontext einer anderen Tonart, also intervallgetreu (Abb. 1 b)),
  2. in der Ausgangstonart, aber auf einem anderen Ton beginnend, also u. U. nicht intervallgetreu, sondern angepasst,
- Stauchung (Abb. 1 c)) oder Streckung (Abb. 1 d)), z. B. werden aus allen 8teln 16tel, oder aus allen 8tel 4tel,
- Krebs, also vertikale Spiegelungen oder "rückwärts wiedergegeben" (Abb. 1 e)),
- Umkehrung, also horizontale Spiegelungen, dabei nicht unbedingt intervallgetreu, sondern an die Tonart angepasst (Abb. 1 f)),
- Umspielung, also das Motiv mit zusätzlichen Tönen versehen. Dabei ist zu beachten, dass die Originaltöne natürlich kürzer werden (Abb. 1 g)),
- Verschiebung auf eine andere Zählzeit (Abb. 1 h)). Dies ist aus musikalischer Sicht trivial, aus algorithmischer hingegen nicht,
- Veränderte Fortführung, d. h. der Anfang ist zwar gleich, ab einem bestimmten Punkt ändert sich die Melodie aber (Abb. 1 i)).

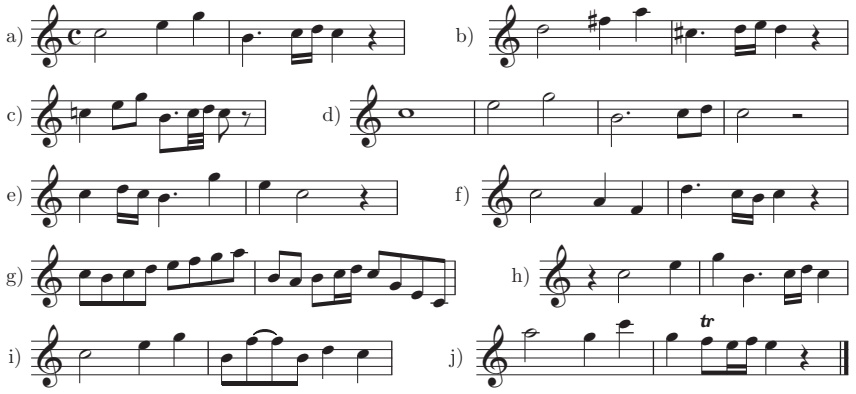


Abbildung 1: a) Eingangsmotiv aus Mozarts *Sonata Facile* (KV 545), b) Transposition, c) Streckung, d) Stauchung, e) Krebs, f) Umkehrung, g) Umspielung, h) Verschiebung, i) veränderte Fortführung, j) Mozarts Original-Variation

Diese Techniken sind problemlos algorithmisch zu erkennen, indem ein *Sliding Window* über die Noten fährt und jeweils zwei Noten überprüft. Dabei wird ein Entscheidungsbaum (siehe Abb. 2) abgearbeitet: Sind die ersten beiden Noten gleich, wird auf eine originale Wiederholung oder eine Umkehrung geprüft. Bei der Umkehrung ist der Sprung genauso weit, die Richtung jedoch verschieden, daher *Abs(Sprung)*. Sind die ersten Noten nicht gleich, wird der Unterschied festgestellt und dieselbe Veränderung (nach Tonlängen und

-höhen getrennt) auf die zweite Note angewandt. Erreicht man damit eine Gleichheit, wird die Veränderung auf das gesamte Motiv angewandt und dieses dann mit der Fundstelle verglichen. Der Code hierzu ist unter http://www.jannikarndt.de/media/docs/findVariation.py verfügbar.

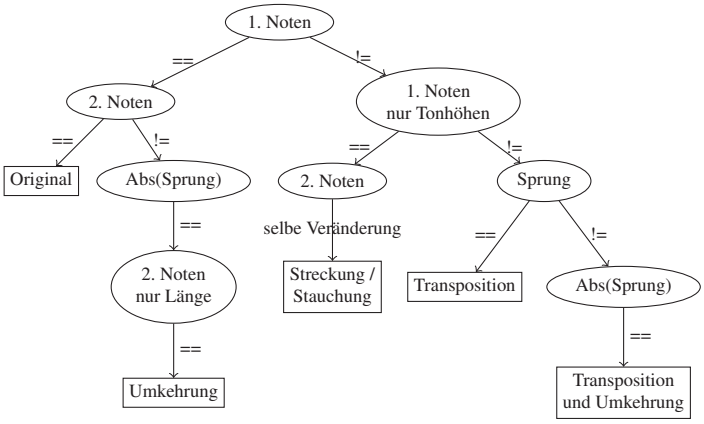


Abbildung 2: Entscheidungsbaum um anhand von zwei Noten Umkehrungen, Streckungen/Stauchungen und Transpositionen zu entdecken.

3 Erkennung unbekannter Variationen

Insbesondere aber um aus kleinen Motiven ganze Themen zu erstellen werden in der Regel andere Variationen genutzt, die nicht standardisiert und daher auch nicht einfach zu entdecken sind. Abb. 1 j) zeigt die Variation, die Mozart in seiner *Sonata Facile* (KV 545) gewählt hat. Sie benutzt keine der oben genannten Vorgehensweisen. Vergleicht man die Tonsprünge der beiden Motive, so findet man keine Ähnlichkeit. Trotzdem erkennt man beim Hören, dass die beiden Motive zusammengehören.

Die Musikpsychologie erklärt diese Fähigkeit anhand der Gestalt-Theorie [But92, PW04]. Ziel für einen Algorithmus ist es also

- nach dem *Gesetz der Prägnanz* Unterteilungen zu finden, die sich durch ein bestimmtes Merkmal abheben und dabei einfache Strukturen ergeben,
- nach dem *Gesetz der Nähe* nicht auf konkrete Intervallsprünge zu achten, sondern Bereiche zu entdecken, in denen ähnliche Arten von Sprüngen auftreten und
- nach dem *Gesetz der Kontinuität* die Richtung von Läufen (auf- oder absteigend) zu berücksichtigen.

Abb. 3 zeigt, wie diese Gesetze auf den Anfang und die Variation des Motivs aus der *Sonata Facile* angewandt werden können. Aus diesen Vorgaben kann folgender heuristischer Algorithmus aufgestellt werden:

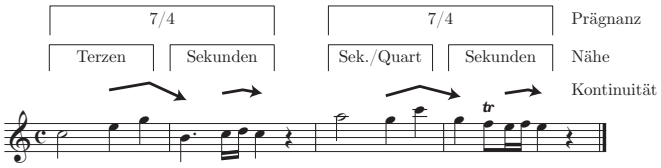


Abbildung 3: Die drei wichtigsten Gestalt-Gesetze im Anfang von Mozarts *Sonata Facile*.

Zunächst wird das Stück in *Kandidaten* unterteilt die eine ähnliche Länge wie das gesuchte Motiv haben. Nun werden für Teilbereiche dieser Kandidaten, z. B. ganze, halbe oder viertel Takte, die Intervalle berechnet und nur das am häufigsten vorkommende mit dem Motiv verglichen. Die Unterteilung kann aus dem Motiv gewonnen werden, indem das Doppelte des längsten Notenwertes gewählt wird. Im Beispiel von Abb. 3 ist der längste Notenwert eine Halbe, daher werden ganze Takte betrachtet. So ist sichergestellt, dass für jede Unterteilung mindestens ein Intervall existiert. Als dritter Schritt werden die Richtungen der Melodielinie berechnet. Da z. B. Umspielungen als Rauschen betrachtet werden, ist es sinnvoll, die Zählzeiten des Originalmotivs auf den Kandidaten zu übertragen und nur die Richtung zwischen diesen *Referenzpunkten* zu vergleichen.

4 Zusammenfassung

Variationen von Motiven und Themen sind ein zentraler Bestandteil der Musik. Es ist daher eines der grundlegenden Probleme des Music Information Retrieval, diese Variationen erkennen und in größeren Werken finden zu können. Für einfache Variationen, die nach bekannten Regeln erstellt werden, ist dies mit einer Kombination eines *Sliding Windows* mit einem Entscheidungsbaum einfach umsetzbar. Für komplexere Variationen muss jedoch auf Erkenntnisse der Musikpsychologie zurückgegriffen werden, welche als Erklärung für das menschliche Erkennungsvermögen die Gestalttheorie anführt. Anhand dieser kann ein heuristischer Algorithmus erstellt werden, der losgelöst von konkreten Notenwerten die größeren musikalischen Konzepte vergleicht. Es steht noch aus, diesen Algorithmus zu implementieren und seinen Erfolg in der Praxis zu zeigen.

Literatur

[But92] David Butler. *Musician's Guide to Perception and Cognition/Book and Disk*. Schirmer Books, 1992.

[LInR08] Pedro J Ponce De León, José M Iñesta und David Rizo. *Mining Digital Music Score Collections : Melody Extraction and Genre Recognition*. Number November. 2008.

[PW04] MT Pearce und GA Wiggins. Rethinking gestalt influences on melodic expectancy. *Proceedings of ICMPC8*, Seiten 367–371, 2004.

Generation of Training Data for Learning-to-Rank Processes in an Expert Seeking Application

Felix Beierle, Felix Engel, Matthias Hemmje

University of Hagen
Multimedia and Internet Applications
felix@beierle.de {felix.engel, matthias.hemmje}@fernuni-hagen.de

Type of work: Master's thesis
Advisors: Dipl.-Inf. Felix Engel, Prof. Dr.-Ing. Matthias Hemmje

Abstract: One of the most important assets in large companies are their experts. Studies in the field of *expertise seeking* have shown that while the specific topic of knowledge is the most relevant aspect in actual searches for experts, other factors regarding the context of the search have to be considered. Within a framework developed at the University of Hagen, *Learning to Rank* is used to learn a ranking function that ranks expertise seeking search results by relevance. This thesis addresses the semi-automatic generation of training data needed for the learning process, employing rules to express relevance patterns.

1 Introduction

Especially in large organizations, the task to find an expert for a specific field is not trivial. Existing applications for the search for experts often only consider topic-factors related to the actual field of expertise. Studies in the field of *expertise seeking* show that while the so-called *quality-related* factors are the most significant in terms of relevance when searching for an expert, there are other important factors, especially contextual factors regarding the relationship of searcher and potential expert, e.g., their familiarity [HBBR10]. In this master's thesis an existing framework developed at the University of Hagen within the SMART VORTEX project (http://www.smartvortex.eu) is used and enhanced. It can retrieve different kinds of information from a semantically annotated knowledge base. The information about the potential experts are stored in a *feature vector* configured by a domain expert. To learn a ranking function to rank the generated feature vectors by relevance, training data has to be provided for the used *Learning to Rank* (LTR) library (different LTR libraries can be used, e.g., RankLib, http://sourceforge.net/p/lemur/wiki/RankLib). Finding reliable training data is an expensive task; so far crowd sourcing, log analysis, or manual specification have been suggested [DFTM12]. This master's thesis follows the idea of [EJH13] to use a system of rules to semi-automatically generate training data for LTR processes. The specific contributions reported here are the analysis of expert seeking

parameters that can be considered in such a software (Section 2), and the elaboration and refinement of a system of rules for the comparison of feature vectors on multiple levels (Section 3). As a further part of the thesis, the existing framework is extended to support the retrieval of the needed feature values, and the rule system is implemented.

2 Retrieving Information from the Knowledge Base

Consider the following example: A user of the expert seeking application is assigned to a new project and needs an expert. He is searching for a programming expert to give a presentation on advanced programming techniques with Java and Perl. In this example, the searcher can give the skills 'Java,' 'Perl,' and 'Presentation' as the query.

When searching for an expert, different *expert seeking parameter*s have to be taken into account. *Quality*-related relevance factors are about the formal qualifications of the potential experts [WvdHS12]. ***Topic*** refers to direct links between a user and the asked skills. We use the term ***approach*** bundling aspects regarding the expert's perspective on the asked field of expertise. ***Up-To-Dateness*** refers to temporal information like the last time an asked skill was used for a project. ***Experience*** can include factors like the amount of time someone has been working for the company, years of work experience, number of projects, or the number of connections someone has in the semantically annotated company knowledge base, etc. Studies in expertise seeking come to the conclusion that the familiarity between searcher and expert is the most important relevance factor (with about 10-20%) in the *accessibility* category [WvdHS12, HBBR10]. We refer to space- and time-constraints with the parameter ***proximity*** and to other relational aspects with ***closeness***. In contrast to the quality-related factors, all of the accessibility-related factors depend not only on the expert, but also on the user that is performing the search.

The general approach is as follows: For every person in the knowledge base, a *feature vector* (or *feature value vector*) is constructed, consisting of several *feature*s (e.g. the number of finished related projects). Each feature is represented through a *feature value*. Each expert seeking parameter consists of a set of *relevance aspect*s. One relevance aspect can consist of a single feature (e.g. age). As motivated in [EJH13], there can be dependencies between features, and therefore, one relevance aspect can also consist of more than one feature (e.g. number of projects *and* years of work experience). The framework uses a pairwise LTR approach: Using a system of rules expressing relevance patterns, labeled training data consisting of pairs of feature vectors, along with the information which of the two is to be considered more relevant, is generated. Using the training data, LTR is applied to learn a ranking function by estimating the weight of every component of the feature vectors. The learned ranking model can be used to classify feature vectors from future searches.

Before the software can be used, a *domain expert* has to configure the system. Similar to the idea of an *application context* in [ADM06] we propose that a domain expert with knowledge about the ontology configures the search. In the *feature vector configuration*, he defines the features and how the feature values are calculated. He also defines rules

for all relevance aspects, for example: If a person A has completed more projects related to queried skills, and has more years of work experience than person B, then A is more relevant with respect to the relevance aspect 'work experience'. Once the configuration, e.g. for *skill*, of the vector and the rule system is completed, a user may enter a query for a particular set of skills and gets a list of the company's employees sorted by relevance with respect to the queried skills; this list is sorted by the previously learned ranking function.

3 Rule System for Relevance Labeling in a Pairwise LTR Approach

The comparison of two feature vectors $\vec{a}, \vec{b} \in \mathbb{R}^n_{\geq 0}$ can be done with respect to single-feature and multi-feature relevance aspects, and with respect to one or more expert seeking parameters, each consisting of a set of relevance aspects.

Single-Feature Relevance Aspect Comparison Most relevance aspects consist of one single feature (e.g. age). For comparing $\vec{a}$ and $\vec{b}$ with respect to such a single-feature relevance aspect, the corresponding feature values $a_i, b_i \in \mathbb{R}_{\geq 0}$ are compared (the index i indicating the position in the feature vector). Besides providing the two basic comparison operators, greater ($>$) and lesser ($<$), another requirement could be to consider only values that are higher than a certain *threshold* $t \in \mathbb{R}_{\geq 0}$. For instance, looking for an expert with the highest experience, the comparison operator is $>$. A threshold can be used to disregard employees that have not finished at least a certain number of projects. If a company wants to support their younger employees, the relevance aspect 'age' could be used with the comparison operator $<$, using a threshold to disregard employees that are too young.

The comparison yields either T (if a_i is considered more relevant than b_i), F (if b_i is more relevant than a_i), or 0 (neither of a_i, b_i is more relevant than the other one). Thus, the comparison function $c_\diamond(a_i, b_i, t)$ for single-feature relevance aspects with $\diamond$ being $>$ or $<$ and with threshold t has the target set $\{T, F, 0\}$ and is defined as follows:

$$c_>(a_i, b_i, t) = \begin{cases} T & a_i \geq t \wedge a_i > b_i \\ F & b_i \geq t \wedge a_i < b_i \\ 0 & \text{otherwise} \end{cases} \qquad c_<(a_i, b_i, t) = \begin{cases} T & a_i \geq t \wedge a_i < b_i \\ F & b_i \geq t \wedge a_i > b_i \\ 0 & \text{otherwise} \end{cases}$$

Note that both single feature value comparison functions are complementary in the first two arguments, i.e., $c_\diamond(a_i, b_i, t) = T \Leftrightarrow c_\diamond(b_i, a_i, t) = F$, and $c_\diamond(a_i, b_i, t) = 0 \Leftrightarrow c_\diamond(b_i, a_i, t) = 0$, for $\diamond \in \{<, >\}$ and for all non-negative values a_i, b_i, t.

Multi-Feature Relevance Aspects Comparison Following the example of 'work experience,' a person has to have both more years of work experience and a higher number of finished projects to be considered more relevant. In order to compare such multi-feature relevance aspects, several single feature value comparisons have to be taken into account. For this we introduce a three-value logic for the conjunction of two values in $\{T, F, 0\}$: $T \wedge T = T$ and $F \wedge F = F$ and all other conjunctions are evaluated to 0. The idea of this conjunction is that one feature vector has to be more relevant for all single comparisons of the multi-feature comparison to be more relevant with respect to the given multi-feature relevance aspect, for instance:

$$c_>(a_i, b_i, t_1) \wedge c_>(a_j, b_j, t_2) = \begin{cases} T & c_>(a_i, b_i, t_1) = T \wedge c_>(a_j, b_j, t_2) = T \\ F & c_>(a_i, b_i, t_1) = F \wedge c_>(a_j, b_j, t_2) = F \\ 0 & \text{otherwise} \end{cases}$$

Comparison with Respect to Sets of Relevance Aspects For an expert seeking parameter E, let x be the number of comparisons of relevance aspects in E that determine $\vec{a}$ more relevant and let y be the number of comparisons that determine $\vec{b}$ more relevant. If $x > y$, $\vec{a}$ is considered more relevant, if $y > x$, $\vec{b}$ is considered more relevant, otherwise they are considered equally relevant regarding that expert seeking parameter.

Feature Vector Comparison For the comparison of two feature vectors, there is one further level: the aggregation of the results of the comparison with respect to a set of expert seeking parameters. A value between 0 and 1 is assigned to each expert seeking parameter, signifying the percentage of relevance the parameter should take up. The percentages considering a feature vector more relevant are accumulated, and the person with the feature vector rated at a higher cumulated percentage value is labeled as the more relevant person for the given search.

4 Conclusion and Further Work

Within the framework used in this work, we addressed the semi-automatic generation of training data for LTR processes in an expert seeking application, thus avoiding the expensive and time-consuming process of manually generating such data. Future work includes an evaluation of the approach, especially regarding the dependencies between the amount of training pairs, the quality of the ranking model, etc. A further aspect that should be addressed is the implementation of online-learning, where the ranking function is updated through user feedback from previous searches.

References

[ADM06] Riccardo Albertoni and Monica De Martino. Semantic Similarity of Ontology Instances Tailored on the Application Context. In *OTM Conferences*, LNCS Vol. 4275, pages 1020–1038. Springer, 2006.

[DFTM12] Lorand Dali, Blaž Fortuna, Thanh Tran, and Dunja Mladenić. Query-independent Learning to Rank for RDF Entity Search. *The Semantic Web*, pages 484–498, 2012.

[EJH13] Felix Engel, Matthias Juchmes, and Matthias Hemmje. Expert search in semantic annotated enterprise data: integrating query- dependent and independent relevance factors. In *LWA 2013 - Lernen, Wissen & Adaptivität. Workshop Proceedings.*, pages 41–44, Bamberg, 2013.

[HBBR10] Katja Hofmann, Krisztian Balog, Toine Bogers, and Maarten de Rijke. Contextual factors for finding similar experts. *Journal of the American Society for Information Science & Technology*, 61(5):994–1014, 2010.

[WvdHS12] Lilian Woudstra, Bart van den Hooff, and Alexander P. Schouten. Dimensions of quality and accessibility: Selection of human information sources from a social capital perspective. *Information Processing & Management*, 48(4):618–630, 2012.

Classifying Incidents in Microblogs using Deep Belief Networks

Marco Ballhausen, Peter Felber, Thomas Klir, Ca Way Le

Department of Computer Science, Telecooperation
University of Technology Darmstadt
Hochschulstr. 10
D-64289 Darmstadt, Germany

{marco.ballhausen, thomas.klir, caway}@gmx.de, peter.felber@gmx.net

Kind of work: Project group | Supervisor: Dr. Benedikt Schmidt, Axel Schulz

Abstract: Microblogs such as Twitter[1] or Facebook[2] posts are a new phenomenon of today's society and gain raising importance in identifying emergency-related situations occuring in everyday life. The detection of such incident-posts is still a challenge as they are often composed of a mixture of everyday language and internet slang.

This paper focuses on classifying incident-related messages in the social network Twitter, which allows emergency-managements to filter and gather additional information automatically about incidents that occured. In our approach, we make use of Deep Belief Networks in combination with supervised machine learning to classify tweets. The evaluation result for our approach revealed an accuracy of 88.5% and a performance comparison between Natural Language Features and Deep Belief Network Features.

1 Introduction

The number of smartphone users has increased dramatically in the last years[3]. Combined with the also increasing propagation of social networks, people tend to post events and incidents, which they encounter in their everyday life. These information are valuable, because emergency managers will notice incidents and get a better overview of the situation on-scene. This leads to better and faster coordination. Ushahidi[4] is a social platform especially designed to be used for reporting incidents of any kind. This platform got popular during the Haitian earthquake or the terrorist attacks in Mumbai. Incident-related posts can be find in Twitter, too. These posts reveal mostly small-scaled incidents, like car crashes or shootings. In contrast Ushahidi is focused on large-scaled incidents, such as earthquakes. The following Tweet shows what a fire related Tweet can look like:

```
Eye Opener: Firefighters battle a fast-moving
  fire in Southern Calif. http://t.co/c7atcFcxq3
```

The challenge for someone interested in these information is to identify all incident related messages amongst the huge amount of 175 million Tweets per day[5].

[1] https://twitter.com/
[2] http://www.facebook.com
[3] http://statista.com/statistics/201182
[4] http://www.ushahidi.com
[5] http://www.briansolis.com/2012/02/the-state-of-the-twitterverse-2012/

Current approaches classify relevant messages by identifying features with Natural Language Processing (NLP) techniques and use them for machine learning [SRP13]. The researchers in [AVSS12, AHH+12, LLKC12] gathered and analysed Twitter data, whereas the first two focus on fire events. Lie et al. [LLKC12] introduce a system with a classifier, using Twitter specific features, such as hashtags, as well as spatial and temporal characteristics. In [VHSP10], the authors analyse microblogging in the course of two hazardous events to identify whether Twitter can contribute to situational awareness. The main challenge in this research topic is that the portion of incident related Tweets are rather small compared to the Tweet-traffic per day, thus making it hard to adapt machine learning techniques.

The approach in this paper will take advantage of Deep Belief Networks (DBNs) in order to find suitable features for supervised machine learning. The authors in [SHR11] showed that DBNs produce better classification results than other learning techniques, like Maximum Entropy and Boosting based classifiers. In this paper 'word2vec'[6] is the implementation of choice. The performance can be compared to Support Vector Machines. Thus, this method is considered to increase overall accuracy for this classification problem of tweets.

2 Dataset and Ground Truth

In order to realise Tweet-classification the approach is based on supervised machine learning. One requirement for this is a labeled dataset. Tweets are labeled with one of the following states: 'No Incident', 'Shooting', 'Fire', 'Crash' and 'Injured people'. This means that we consider a single label/multi-class problem. The approach's dataset consists of 5000 Tweets, which have been crawled from Seattle, for the time from July until August 2013. To achieve a ground truth the dataset has been labeled by four Master students in Computer Science to one of the possible states. The labels of each Tweet have been discussed to have objective labels and by that increase the quality of the labels. The outcome of this process forms the ground truth and is the base for the supervised machine learning. The distribution of labels amongst the ground truth can be found in Table 1.

Label	No Incident	Shooting	Fire	Crash	Injured people
Allocation	77,56%	9,1%	8,24%	4,78%	0,32%

Table 1: Distribution of labels types amongst the ground truth

3 Approach

The classification approach has been realised by implementing several Apache UIMA[7] pipelines in chain. The architecture of this chain of pipelines can be found in Figure 1.

Preprocessing Pipeline: includes different standard NLP components: Tokenization; Lemmatization; Stemming; Part-of-Speech tagging; Stopword Removal; Slang, emoticon and hyperlink replacement. The outcome is filtered text.

[6] https://code.google.com/p/word2vec/ [7] http://uima.apache.org/

Deep Belief Network: is used on the filtered text to build up word relation model. With that model, words are arranged in clusters by the word2vec-clustering algorithm.[6]

Feature Pipeline: extract features based on the data derived by the previous steps. These features will be compared in further steps and can be splitted into two kinds:

NLP-features: state of the art NLP-classification techniques:

- Word extraction: The occurence of a word as binary vector.
- TF-IDF: This metric reflects how important a word is to a document.
- Exclamationmarks: the number of exclamation marks used in a Tweet.
- No. of uppercases: The number of uppercase letters used in a Tweet.
- Person: A set of possible persons which occur in a Tweet (prefixed by '@')

DBN-features: are binary features, which are extracted from each cluster of the outcoming model in the DBN pipeline. For example, if a word of a Tweet is found in a cluster, the appropriate feature is set to 1.

Supervised Learning: The previously generated features serve as input for the supervised machine learning algorithm. The java-framework WEKA[8] has been used to accomplish this task. JRIP has been used as a rule-based learning approach in order to keep the opportunity to reproduce the learned model at a later time.

Evaluation Pipeline: uses the validation set as input and evaluate the quality of the extracted features and the learned model.

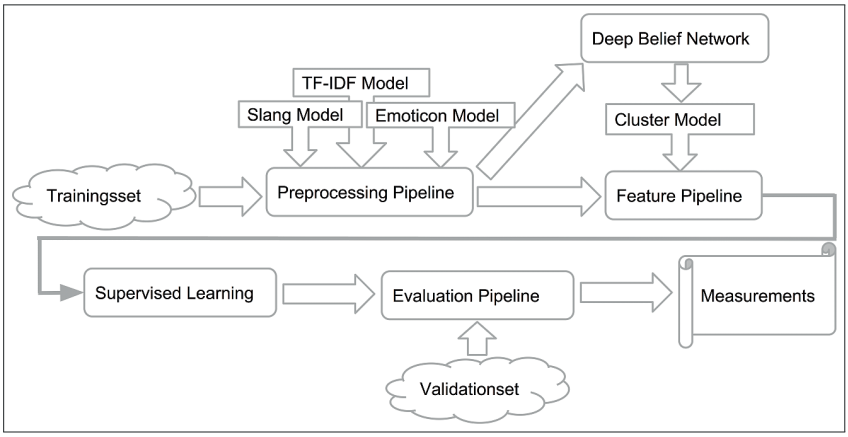


Figure 1: The implementation consists of several pipelines

4 Evaluation

The evaluation compares the metrics accuracy and f-measure of the DBN-features, NLP-features and both in combination. It turned out that both metrics performs best if used as standalone. This means by using the DBN-features without the NLP-features. This results in an accuracy of 88,5% and f-measure of 87,62% (see Figure 2 and Figure 3).

These results might be dependant on the selection of NLP features. This could be examined further by evaluating additional features, such as Character N-Grams. In the case

[8] http://www.cs.waikato.ac.nz/ml/weka/

of better results with other setups, differences between NLP-features and DBN-features has to be proven to be as significant as with the current setup. Different classifiers (e.g. Naive Bayes) and more training data should be used for a broader conclusion regarding the performance of features created from DBNs in supervised machine learning.

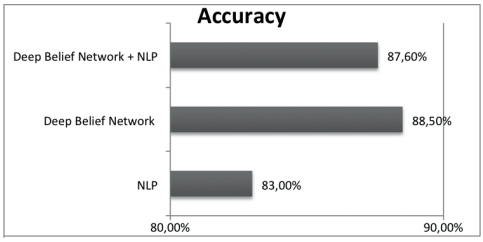


Figure 2: Accuracy for the different Feature sets

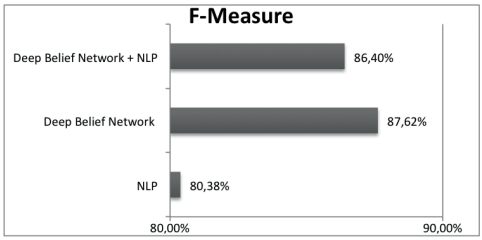


Figure 3: F-Measure for the different Feature sets

5 Conclusion and Outlook

This paper present an approach to find incident-related Twitter posts. Furthermore, the performance of DBN-features and NLP-features are compared. The evaluation shows that using features extracted from a DBN leads to 88,5% accuracy and by that about 5,5% better than features derived with NLP-methods. This means that there may be great potential in using DBN for extracting features and integrate them in supervised machine learning techniques. Features derived from DBN-clusters or word-to-word metrics could improve accuracy as well as reducing the amount of necessary training to identify incident-related Tweets. This is particulary useful in the area of identifying small-scale incidents in social media like Twitter. Further research in this topic could reveal more sophisticated features.

References

[AHH+12] Fabian Abel, Claudia Hauff, Geert-Jan Houben, Richard Stronkman, and Ke Tao. Twitcident: fighting fire with information from social web streams. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 305–308. ACM, 2012.

[AVSS12] Puneet Agarwal, Rajgopal Vaithiyanathan, Saurabh Sharma, and Gautam Shroff. Catching the Long-Tail: Extracting Local News Events from Twitter, 2012.

[LLKC12] Rui Li, Kin Hou Lei, Ravi Khadiwala, and KC-C Chang. TEDAS: a twitter-based event detection and analysis system. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1273–1276. IEEE, 2012.

[SHR11] Ruhi Sarikaya, Geoffrey E. Hinton, and Bhuvana Ramabhadran. Deep belief nets for natural language call-routing. In *ICASSP*, pages 5680–5683. IEEE, 2011.

[SRP13] Axel Schulz, Petar Ristoski, and Heiko Paulheim. I See a Car Crash: Real-Time Detection of Small Scale Incidents in Microblogs. In Philipp Cimiano, Miriam Fernández, Vanessa Lopez, Stefan Schlobach, and Johanna Völker, editors, *ESWC (Satellite Events)*, volume 7955 of *Lecture Notes in Computer Science*, pages 22–33. Springer, 2013.

[VHSP10] Sarah Vieweg, Amanda L Hughes, Kate Starbird, and Leysia Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1079–1088. ACM, 2010.

Data Mining beim Widerstandspunktschweißen: Vorgehensweise und erste Ergebnisse der Prognose von Punktdurchmessern

Benjamin Hoffmann, Josef Mögelin, Benjamin Arndt, Curtis Mosters

{hoffmanb, moegelin, arndbenj, mosters}@fh-brandenburg.de

Art der Arbeit: Semesterübergreifendes Masterprojekt

Betreuer der Arbeit: Dipl.-Ing. Christoph Großmann, Dipl.-Inform. Ingo Boersch

Abstract: Beim Widerstandsschweißen spielt der richtige Zeitpunkt des Elektrodenwechsels eine entscheidende Rolle für die Festigkeit der Verbindung und den Ressourcenverbrauch. Wegen einer latenten Verbindungsbildung kann der dafür wichtige Punktdurchmesser aber nicht direkt während des Schweißvorganges gemessen werden. Durch die Vorhersage der Schweißlinse bzw. des Punktdurchmessers mittels eines Prognosemodells könnte die Standmenge optimiert werden. Diese Arbeit beschreibt die Merkmalsextraktion, Merkmalsselektion und Modellerstellung an einer realen Datenmenge. Das finale Modell kann den Linsendurchmesser eines Schweißpunktes zerstörungsfrei in mehr als 92% der Fälle korrekt vorhersagen.

1 Einleitung und verwandte Arbeiten

Punktdurchmesser können beim Widerstandspunktschweißen nur unter großem Aufwand gemessen werden. Eine zuverlässige Vorhersage anhand der Beobachtung mehrerer Verlaufsgrößen des Schweißprozesses (z.B. Spannung, Strom) würde es ermöglichen, den Zeitpunkt eines Elektrodenwechsels und somit den Ressourcenverbrauch zu optimieren. Das Ziel der Arbeit besteht in der Auswahl eines geeigneten Lernalgorithmus (Setups) zur Prognose des Punktdurchmessers und eines ersten Modells mit Güteschätzung. Von einem Kooperationspartner wurden dazu 16 separate Versuchsreihen mit 20.012 Schweißpunkten bereitgestellt. Insgesamt sind 3.241 dieser Punkte klassifiziert. Dies stellt eine deutliche Steigerung gegenüber der in [AH13, MM13] verwendeten Datenmenge dar. Für jeden Schweißpunkt liegen sieben Verlaufsgrößen mit je 24.000 Werten vor.

Wissenschaftliche Arbeiten, die transparente Modelle ([BHS07]) (z.B. Modellbäume, siehe Wekas M5P [Qui92, WW97]) und mehr als 3.000 klassifizierte Punkte zur Vorhersage des Punktdurchmessers einsetzen, wurden unseres Wissens noch nicht betrachtet. In [PHLJ04] wird ein Bayessches Netz zur Vorhersage des Punktdurchmessers benutzt. Punktdurchmesser werden in drei Klassen eingeteilt. Die Merkmale werden mittels einer Klassifikation der Punkte anhand von Quartilen definiert. Als Kanäle (Verlaufsgrößen) dienen Spannung, Stromstärke und Druckkraft. Bei [HLJ+06] handelt es sich um eine Arbeit, die Data-Mining-Techniken auf Messdaten beim Widerstandsschweißen einsetzt, um den zu-

grundeliegenden Prozess zu identifizieren. Es werden verschiedene Selektionsverfahren zur Prozessidentifikation anhand gemessener Kanäle untersucht. Dabei werden 54 geometrische und statistische Merkmale aus den segmentierten Kurven extrahiert. Zusätzlich wird jeder Kanal in zehn gleichgroße Segmente unterteilt, von denen der Mittelwert berechnet wird. Unter Verwendung fünf verschiedener Merkmalsselektionsverfahren, eines 3-Nearest-Neighbour-Klassifikators und einer 2/3-1/3 Aufteilung in Trainings- und Testdaten wurde eine Erfolgsrate von 99,3% erreicht. Weitere Arbeiten verwenden Methoden des unüberwachten Lernens (selbstorganisierende Karten) zur Veranschaulichung von Abhängigkeiten ([HPLJ04]) oder Hopfield-Netze zur Qualitätsschätzung ([CR04]).

2 Vorgehen

Für jeden Punkt werden die beim Schweißen aufgezeichneten Verlaufsgrößen segmentiert. Aus den entstehenden Segmenten werden Merkmale extrahiert. Mittels eines Selektionsverfahrens werden die besten Merkmale ermittelt, die zusammen mit dem Punktdurchmesser als Eingabe für den Lernalgorithmus dienen. Dieser erstellt aus den Daten ein Modell zur Prognose des Punktdurchmessers.

Es werden zwei verschiedene Segmentierungsansätze verwendet. Für vier Kanäle werden die Schnittstellen anhand der jeweils zwei größten Minima/Maxima der geglätteten zweiten Ableitung bestimmt (in [AH13] werden jeweils drei Minima/Maxima benutzt). Für die restlichen Kanäle wird eine Schwellwertsegmentierung unter Betrachtung des Minimum und Maximum bzw. deren Differenz eingesetzt. Beide Ansätze liefern fünf nicht überlappende Segmente. Das mittlere Segment wird in zwei gleich lange Teile aufgesplittet.

Statistische und geometrische Merkmale werden auf jedem der sechs Segmente aller Kanäle berechnet. Zusätzlich zu den in [AH13] beschriebene Merkmalen wurde der erste und letzte Wert eines Segments, die Indizes des Minimums/Maximums, die Varianz und auf der geschätzten Autokorrelationsfunktion das Minimum/Maximum (mit Indizes), der Mittelwert, die Standardabweichung, die Varianz und die Quartile berechnet. Insgesamt entstehen dadurch 1.426 Merkmale je Schweißpunkt.

Bei der Merkmalsselektion wurden die besten n Merkmale dreier verschiedener Methoden untersucht: Pearson-Korrelationskoeffizient, Rangkorrelationskoeffizient und χ^2-Korrelation. In der Modellauswahl wurden überwiegend transparente (LR (Lineare Regression), M5P, M5Rules) und einige wenige nicht transparente Modelle (RF (Random Forest), SVM (Support Vector Machine), k-NN (k-Nearest Neighbor mit k=1)) evaluiert. Für eine ausführliche Erklärung der Verfahren wird auf [WF05] verwiesen. Jeder dieser Lernalgorithmen wurde mit jeder Merkmalsselektionsmethode mit den besten 10, 50, 100, 250, 400 Merkmalen bzw. ganz ohne Merkmalsselektion untersucht.

Um die beste Kombination von Merkmalsselektion und Lernalgorithmus (Setup) auszuwählen, wird der in [MM13] definierte Default-Prozess (Abbildung 1) eingesetzt. Bei diesem werden 10% der Daten vorenthalten (Holdout). Mit den restlichen 90% wird mittels einer Kreuzvalidierung (k=10) die Performanz des Setups (innerhalb der Kreuzvalidierung: Merkmalsselektion und Modellerstellung) anhand zweier Fehlermaße, dem Root Mean Squared Error (RMSE) und dem User-Defined-Error (UDE), berechnet. Ein Fehler im Sinne des

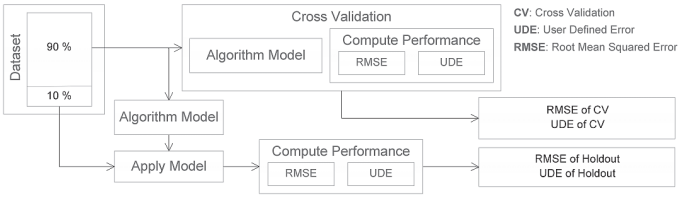


Abbildung 1: Setup-Evaluationsprozess (Standardprozess, vgl. [MM13])

UDE stellt eine mehr als 10% abweichende Prognose $\hat{x_i}$ vom echten Wert x_i dar:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{x_i} - x_i)^2}, UDE = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}0, & falls\, 0,9 \cdot x_i < \hat{x_i} < 1,1 \cdot x_i \\ 1, & sonst\end{cases}.$$

Mit den 90% der Daten wird ein Modell erstellt. Dieses wird auf die enthaltenen 10% der Daten angewendet mit anschließender Bestimmung des RMSE und des UDE des Holdouts. Insgesamt ergeben sich damit vier Fehlermaße: RMSE/UDE der Kreuzvalidierung und RMSE/UDE der Holdoutmenge. Anhand der Fehlerschätzung werden die besten Setups ausgewählt. Das beste Setup erzeugt abschließend aus 100% der Daten das finale Modell. Alle Analyseergebnisse sind aus den Originaldaten durch Ausführung von RapidMiner-Prozessen/R-Skripten reproduzierbar.

3 Ergebnis und Ausblick

Die Auswahl der besten Setups wurde mittels der RMSE-Mittelwerte der Kreuzvalidierung und den RMSE-Werten des Houldouts erstellt, da durch eine Diskretisierung, wie beim UDE, Unterschiede zwischen den Modellen weniger deutlich werden. Der Default-Lerner, der immer den Punktdurchmesser-Mittelwert der Daten zurückgibt, mit welchen er trainiert wurde, erreicht einen RMSE-Mittelwert in der Kreuzvalidierung von 0,574 (UDE: 40%) und einen RMSE von 0,549 beim Holdout (UDE: 36,9%). Der Punktdurchmesser wird nur in 60% korrekt vorhergesagt.

Random Forest liefert beim RMSE die besten Ergebnisse; eine Merkmalsselektion lieferte bei diesem Lernalgorithmus keine Verbesserung. SVM, M5P, M5Rules und LR besitzen leicht höhere Fehlerraten. Setups mit 1-NN als Lerner liefern im Durchschnitt schlechtere Ergebnisse als der Rest. Eine Merkmalsselektion verbessert das Ergebnis nur geringfügig. SVMs mit allen Merkmalen liefern nur Werte, die mit denen des Default-Lerners vergleichbar sind. Um das beste Setup auszusuchen, wurden aus der Paretomenge der transparenten Modelle (LR, M5P, M5Rules) vielversprechende Setups ausgewählt. Ein kleinschrittigeres Vorgehen bei der Optimierung der Merkmalsanzahl ergab drei finale Modelle (zwei in R, eins in RapidMiner), welche abschließend mittels einer Kreuzvalidierung (k=100) auf 100% der Daten evaluiert wurden. Die Ergebnisse (Setups und unsere Fehler) befinden sich in Tabelle 1.

Im Gegensatz zu [MM13] stellen diese Ergebnisse eine deutliche Verbesserung dar. Der Punktdurchmesser konnte in 92% der Fälle korrekt aus den Messdaten vorhergesagt werden.

Tabelle 1: Performanzschätzung ausgesuchter Modelle, Kreuzvalidierung (k=100), alle Daten

Setup	RMSE-Mittelwert	UDE-Mittelwert	Anzahl linearer Modelle (100%-Modell)
R: M5P, rangbasiert, 249 M.	0,240	7,2%	33
R: M5P, rangbasiert, 251 M.	0,242	7,4%	1
RM: M5P, Pearson, 85 M.	0,25	7,6%	36

Die 100%-Modelle liefern interessante Ergebnisse über positive und negative Einflüsse auf den Punktdurchmesser. Die Arbeit hat gezeigt, dass es möglich ist, den für die Festigkeit eines Schweißpunktes wesentlichen Durchmesser zerstörungsfrei mit Methoden des Data Mining aus Verlaufsgrößen des Schweißprozesses zu bestimmen. Als nächster Schritt ist eine Klassifizierung neuer Daten mit dem besten Modell geplant. Durch eine Performanzmessung unseres Modells an ungesehenen Daten kann überprüft werden, ob unsere Schätzung sich bestätigt. Andere Arten der Merkmalsselektion (Vorwärts-/Rückwärtsselektion) und eine verbesserte Segmentierung bzw. Merkmalsdefinition könnten die Performanz noch verbessern. Die Arbeit im Zuge eines Masterprojekts wurde unterstützt durch das Labor für künstliche Intelligenz der Fachhochschule Brandenburg.

Literaturverzeichnis

[AH13] Benjamin Arndt und Benjamin Hoffmann. Segmentierung und Merkmalsdefinition mehrkanaliger Messdaten zur Prognose bei einem punktförmigen Fügeverfahren. In *14. Nachwuchswissenschaftlerkonferenz ost- und mitteldeutscher Fachhochschulen*, 2013.

[BHS07] Ingo Boersch, Jochen Heinsohn und Rolf Socher. *Wissensverarbeitung: Eine Einführung in die Künstliche Intelligenz für Informatiker und Ingenieure*. Spektrum, 2007.

[CR04] Yongjoon Cho und Sehun Rhee. Quality estimation of resistance spot welding by using pattern recognition with neural networks. *IEEE T*, 53(2):330–334, 2004.

[HLJ+06] E. Haapalainen, P. Laurinen, H. Junno, L. Tuovinen und J. Röning. Feature Selection for Identification of Spot Welding Processes. In *The 3rd International Conference on Informatics in Control, Automation and Robotics*, Seiten 40–45, August 2006.

[HPLJ04] Junno H., Laurinen P., Tuovinen L. und Röning J. Studying the Quality of Resistance Spot Welding Joints Using Self-Organising Maps. In *Fourth International ICSC Symposium on Engineering of Intelligent Systems, Madeira, Portugal*, 2004.

[MM13] Curtis Mosters und Josef Mögelin. Merkmalsselektion und transparente Modellierung zur Prognose einer Zielgrösse bei einem punktförmigen Fügeverfahren. In *14. Nachwuchswissenschaftlerkonferenz ost- und mitteldeutscher Fachhochschulen*, 2013.

[PHLJ04] Laurinen P., Junno H., Tuovinen L. und Röning J. Studying the quality of resistance spot welding joints using Bayesian networks. Seiten 705–711. Artificial Intelligence and Applications, Innsbruck, Austria, 2004.

[Qui92] Ross J. Quinlan. Learning with Continuous Classes. In *5th Australian Joint Conference on Artificial Intelligence*, Seiten 343–348, Singapore, 1992. World Scientific.

[WF05] Ian H. Witten und Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

[WW97] Y. Wang und I. H. Witten. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*. Springer, 1997.

A Knight's path problem as an example to investigate human problem solving

Peter Treiber

TU Kaiserslautern
ptreiber@rhrk.uni–kl.de

Abstract: Complexity theory is a well–established discipline of computer science. With efficient algorithms and fast computers many problems can be solved with computers very fast and very good. But many everyday problems are still solved by hand, e.g. designing a book cover, creating world climate models etc. In complexity theory, problems are classified by hardness. What I use as measure for hardness, is actually the time needed to find a solution. What I want to know is, whether such a measure can be applied for every day problems. To investigate this further, I programmed a little game which I analyzed. My focus lay on the impact of different possible solutions on the hardness of the different levels.

1 Introduction

Almost 50 years ago, Moore predicted that the computational power will double every 18 months [Moo65]. And with the rise of smart–phones, one could think that problem solving is not needed anymore as an ability. Just ask your smart–phone, and it will solve the problem at hand. For some problems, such as navigating, it works already, but for others not. Problem solving is still considered as a significant human competence and is tested in different tests, for instance in the PISA–Test [KFRW01]. But these tests only find out how humans solve problems. What we wanted to know is: Which features of a solution indicate the difficulty of a problem? Human problem solving techniques, as described by Anderson et al. [And93], suggest, that difference reduction is one often used technique. That means that if a problem allows a solution, where in every step the difference is reduced, it should be easier solvable than a problem, which does not allow such a solution. Increasing the difference in one step of the solving process seems to be counter–intuitive and as such should increase the difficulty. To investigate this matter, I designed a small game, which is described in section 2. I then let some people play this game. The results can be seen in section 4. In section 5 I will give an overview about possible further research.

2 A Knight's path problem

The game I designed is a chess–like board–game. On a 12×12 chess–board, the goal is to move a knight from the starting position to a predefined goal field, using as few moves as possible. The knight moves via the "Rösselsprung" over the board. That is, he makes an "L"–shaped move: Two squares in one direction, then one square orthogonal. On some squares there are obstacles, thus hindering the knight to use that square as an intermediate position. I designed six different levels, that is different configurations of starting positions, the locations of the goal, and the number and position of the obstacles. A solution to any of the problems is a sequence of moves that lead the knight from the start to the goal. The length of a solution is defined as the number of moves. I defined two different characteristics of a possible solution: Whether it is monotone or not, and whether it is greedy or not. To understand these concepts in the context of the game, we need the notion of distance between two squares on the board. The distance is measured using the Euklidean distance of the coordinates of the squares. In a monotone solution, the distance to the goal gets smaller in each step, whereas in a greedy solution a square is chosen among all reachable squares which has the smallest distance to the goal. While this seems to be a small difference, the two characteristics are independent, as can be seen in Figure 1.

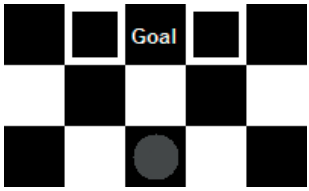


Figure 1: An example where a greedy solution is possible, but no monotone solution exists.

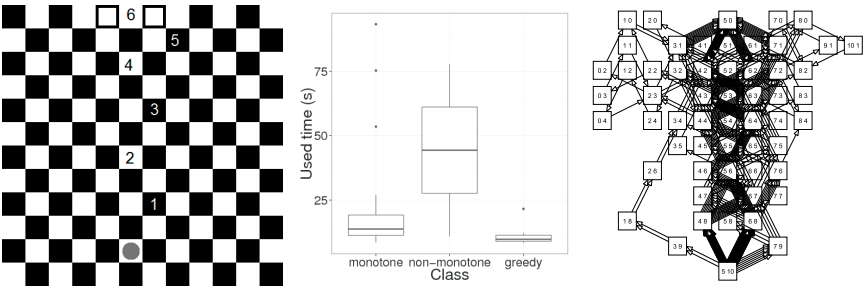
With these two characteristics, I had four classes of solutions: monotone solutions, non–monotone solutions, non–monotone but greedy solutions and monotone and greedy solutions. I designed the six levels in such a way, that each level had a shortest solution of length six, but they could belong to different classes. Since most of the levels were symmetric, there was more than one shortest solution in most levels. There could also be shortest solutions which belonged to different classes in one level. All of the four classes of shortest solutions were covered by at least one level.

3 Participants

Participants were recruited using the Amazon Mechanical Turk. They were paid one dollar to participate in our experiment. 21 started the experiment, 15 finished it.

4 The experiment: Hypotheses and the results

In the experiment, the participants played the six different levels each five times, in a random order. Our first hypothesis was that greedy and monotone solutions would be easier to find. Non–monotone solutions contain at least one move where the distance to the goal grows, so it seems to be a counter–intuitive move. We first tested the number of found solutions against the null–hypothesis that every shortest solution would be found with the same probability. We did a chi–squares test to determine whether our hypothesis was right. The result is different for all the levels: For levels 1 and 3, the probability that our hypothesis is right is nearly zero. For levels 2 and 5 it is between 85% and 95%, whereas for the levels 0 and 4 it is high enough to accept our hypothesis.



(a) Starting configuration of level 4. One sample solution is given by the numbers.

(b) The mean time needed for finding a solution, depending on the class of the solution in level 4.

(c) The found solutions composed into a graph.

Figure 2: Level 4 and the results

We want to elaborate more on the results of level 4. The configuration can be seen in Figure 2a. In this level, the greedy solutions are not monotone. Firstly, we just counted the number of found shortest solutions and compared it to the expected numbers, if all shortest solutions would have been found with the same probability. What we can see in Table 1 is that nearly five times as many greedy solutions as expected have been found and only half as many non–monotone solutions found as one would expect. In Figure 2c we see all found solutions composed in a graph. What we can see there is that there exists a clear corridor where most of the found solutions lie. This corridor contains the greedy solutions. So these solutions are clearly preferred by humans.

Since the numbers are not so high, we also compared the time needed for finding solutions. What we can see in Figure 2b also supports our hypothesis: It seems to be much easier to find solutions that are monotone (left) or greedy (right), since the time needed is much less than for the non–monotone (middle) solutions. What we can see from this level is that our hypothesis about the preferred solutions seems to be right. Whether we can apply this finding to other problems than on the game under examination, will be discussed in the next section.

Solution class	non–monotone	monotone	greedy
expected	4.2	34.28	1.52
found	2	31	8

Table 1: The expected number of solutions based on a uniform distribution in comparison to the number of found solutions.

5 Discussion and future research

In this paper we investigated a small board game to enhance our understanding of human problem solving. As we have seen in the example level, monotone and greedy solutions seem to be preferred by humans. But the results have not been that clear in all the levels, so further experiments are necessary to validate the results. The model of giving all the solutions the same probability seems too rough to get good results, so it would be interesting to see how a different model, e.g. assigning every single move a probability, would change the results. Since many solutions have moves in common, one would be able to better find single steps that make solving the game easier. When finding steps that are more likely to be chosen by humans in context of the game, we would also find steps that are taken less likely. It has been shown that playing video games can enhance one's problem solving ability[GLE13]. With our approach one could develop games which can train problem solving abilities in a way, that one would also consider less likely solutions, thus enabling one to think out–of–the–box.

6 Acknowledgments

The previous research has been done as my bachelor thesis at the University Heidelberg. I would like to thank my supervisors Prof. Dr. Katharina Zweig and Prof. Dr. Gerhard Reinelt for their help and advice.

References

[And93] J. R. Anderson. Problem solving and learning. *American Psychologist*, 48, Januar 1993.

[GLE13] I. Granic, A. Lobel, and R. Engels. The benefits of playing video games. *American Psychlogist*, 2013.

[KFRW01] E. Klieme, J. Funke, D. Leutnerand P. Reimann, and J. Wirth. Problemlösen als fächerübergreifende Kompetenz. *Zeitschrift für Pädagogik*, 2, 2001.

[Moo65] G. E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38, April 1965.

Klassifizierung einer SOA Applikation anhand des ESARC

Danielle Collenbusch, Fekkry Meawad, Patrick Kopf, Tim Kornherr
Vorname.Nachname@Student.Reutlingen-University.de

Masterstudiengang Services Computing
Hermann Hollerith Zentrum
Danziger Straße 6
71035 Böblingen

Abstract: **Durch den ESARC – Enterprise Service Architecture Reference Cube- wird eine Klassifikation der Architektur von SOA-Applikationen ermöglicht. Die Einordnung der SOA-Applikation WebAutoParts.com unter Anwendung der drei ESARC-Kerndimensionen Business and Information, Information Systems und Technology hat gezeigt, dass insbesondere die Information Systems Dimension abgedeckt ist und diese zudem direkte Auswirkungen auf die Technology Dimension besitzt.**

1 Einführung

Zur Klassifizierung der Architekturen von SOA-Applikation gibt es aktuell keine standardisierte Referenzarchitektur. ESARC bietet eine konsistente und vergleichbare Bewertung der Qualität und somit die Möglichkeit der Klassifizierung und Untersuchung von unterschiedlichen Architekturaspekten [Zi11]. Der Fokus des Referenzmodells bezieht sich auf die Evaluierung und Optimierung von SOA-Architekturen [ZZ11]. Er vereint bestehende Referenzmodelle und Architektur Muster wie zum Beispiel TOGAF[OG09]. Ziel ist die Einordnung bestehender Enterprise Architekturen, um eine Vergleichbarkeit zu schaffen. In der folgenden Ausarbeitung wird die SOA-Applikation WebAutoParts.com [Wi12] anhand des abstrakten Architekturreferenzmodells ESARC [ZZ11] klassifiziert. Dies soll es ermöglichen, eine Aussage über die Qualität der SOA-Applikation zu treffen, indem die Abdeckung der Kerndimensionen des ESARC festgestellt wird. Die SOA-Applikation beschreibt ein einfaches Szenario. Deshalb wird der Fokus auf die drei zentralen ESARC-Dimensionen gelegt. Hierzu zählen: *Business & Information Architecture*, *Information System Architecture* und *Technology Architecture*. Diese Dimensionen umfassen jeweils mehrere Schichten, deren Abdeckung untersucht wird. Auf eine Gewichtung der einzelnen Kerndimensionen zur Erfassung der Qualität wurde im Rahmen dieser ersten Untersuchung zunächst verzichtet.

In Kapitel 2 wird zunächst ein Überblick der SOA-Applikation gegeben. Weiterführend wird in Kapitel 3 eine Abbildung des ESARC auf die SOA-Applikation vorgenommen. Hierbei wird dargestellt, welche Schichten innerhalb der drei zentralen Dimensionen abgedeckt werden.

2 WebAutoParts.com

WebAutoParts.com ist ein hypothetischer Online-Autoteile-Shop basierend auf einer Service-orientierten Architektur. Im Rahmen dieser Untersuchung wird der Bestellprozess von Autoteilen über WebAutoParts.com betrachtet. Dieser ist in *Abbildung 1* dargestellt.

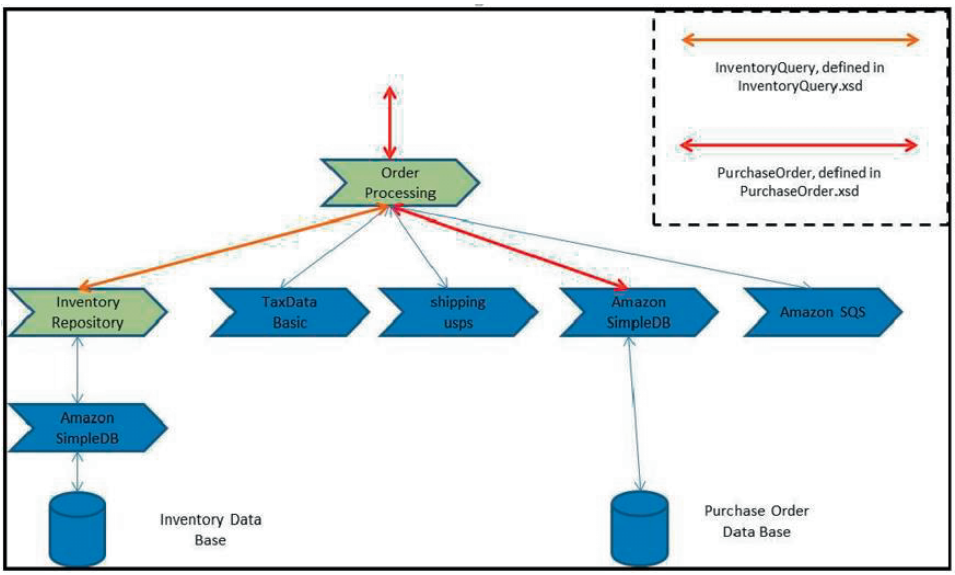


Abb. 1 Order Process WebAutoParts.com [Wi12]

Um den Bestellprozess umzusetzen, werden zwei lokale BPEL-Prozesse (grün) genutzt, die vier externe Webservices (blau) orchestrieren. Diese Vorgehensweise stellt eine agile Cloud Computing-Entwicklungsmethode dar [Wi12]. Aufgrund der grobgranularen Darstellung der WebAutoParts.com Applikation anhand eines Chevron-Diagramms, welches nur beschränkt Informationen preisgibt, müssen für die hier durchgeführte Klassifikation zusätzliche Annahmen getroffen werden.

3 Abbildung des ESARC auf WebAutoParts.com

In den folgenden Abschnitten werden die drei ESARC-Kerndimensionen beschrieben und eine Analyse von WebAutoParts.com vorgenommen. Die Kerndimensionen des ESARC sind in *Abbildung 2* schematisch dargestellt. Für eine vollständige Übersicht aller Dimensionen des ESARC wird auf die Beiträge [Zi11] und [ZZ11] verwiesen.

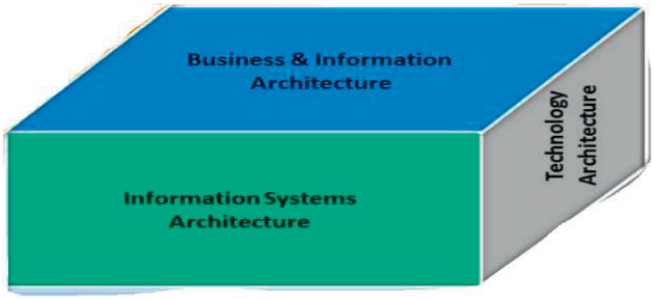


Abb. 2 - ESARC Kerndimensionen [Zi11]

Die *Business & Information* Referenzarchitektur definiert die Verbindung zwischen der Geschäftsstrategie und den unterstützenden, strategischen Informationssystemen [Zi11]. In der betrachteten SOA-Applikation wird die übergeordnete Schicht *Business Vision*, *Drivers*, *Goals*, *Objectives* nicht abgedeckt, da keine Anhaltspunkte zu Rahmenbedingungen wie Ziele und Visionen gegeben sind. Auch die Schicht in Bezug auf die Organisation und Standorte ist aus dem verwendeten Beispiel aufgrund fehlender Beschreibungen nicht ersichtlich. Dementsprechend ist die folgende Schicht der Ressourcen und Aufbauorganisation ebenfalls nicht abgedeckt. An dieser Stelle ist anzunehmen, dass in dem ausführbaren Prozess von WebAutoParts.com verschiedene Rollen hinterlegt sind. Die Ausrichtung des Geschäftsmodells auf Domänen innerhalb der *Business Domains and Capabilities* Schicht ist nicht abgedeckt. Obwohl die Ausrichtung und Aufbauorganisation nicht abgedeckt sind, lässt sich die Ablauforganisation, dargestellt in der *Business Processes, Workflows, Policies, Procedures* Schicht, direkt aus dem Beispiel ableiten. Die Applikation stellt den Prozess einer Bestellung, ausgelöst durch einen Kunden, dar. Es ist wahrscheinlich, dass die Geschäftsregeln (*Business Rules* Schicht) innerhalb des ausführbaren Workflows hinterlegt, jedoch nicht direkt im dargestellten Modell sichtbar sind. Weiterführend wird die *Business Information* Schicht partiell dargestellt. Es sind Datenbanken hinterlegt, woraus der Informationsfluss abgeleitet werden kann. Jedoch sind keine näheren Details über den Nachrichtenfluss bekannt. Aus den primären Schichten ist die verbleibende *Business Products and Services* ebenfalls in dem Modell dargestellt. Durch das im Prozessmodell verwendete *Inventory Repository* wird verdeutlicht, dass Produkte angeboten und bestellt werden können. Genaue Informationen zu der Produktkategorie ergehen hieraus nicht. Die unterstützenden Schichten *Configuration of Business & Information Demands* und *Business Measures and Controls* werden nicht abgedeckt. Somit wird nur die *Business Processes, Workflows, Policies, Procedures* Schicht detailliert dargestellt. Der Informationsfluss sowie Informationen zu Produkten und Service**s** können aus dem Modell nur oberflächlich abgeleitet werden. Alle weiteren Schichten werden als nicht abgedeckt klassifiziert.

Die *Information Systems Architecture* Dimension des ESARC beschreibt den abstrakten Entwurf der individuellen Lösungsarchitektur eines Anwendungssystems [Zi11]. Dabei enthält diese die wichtigsten anwendungsspezifischen Servicetypen und definiert deren Beziehung mittels eines Schichtenmodells [Zi11]. In erster Linie wird die *Information Services for Enterprise Data* Schicht durch den Geschäftsprozessschritt *Amazon SimpleDB* abgedeckt. Hierbei wird durch einen BPEL-orchestrierten Service eine

Datenbank-Abfrage ausgeführt, um Informationen über den aktuellen Lagerbestand zu erhalten. Die Schnittstelle zum Service eines externen Anbieters wird innerhalb der *AmazonSimpleDB.wsdl* definiert. Der orchestrierte Service ist innerhalb der *InventoryRepository.bpel*-Datei beschrieben. Im weiteren Verlauf findet sich erneut ein BPEL-orchestrierter Aufruf gegen die *Amazon SimpleDB*. Hierbei wird die aufgenommene Bestellung persistiert. Eine weitere, abgedeckte Schicht ist *Task Services*. Diese können interne als auch externe Services darstellen [ZZ11]. Als *Task Services* können Prozessschritte angesehen werden, die eine spezifische Aufgabe ausführen. In WebAutoParts.com zählen dazu beispielsweise die Schritte *TaxData Basic*, *shipping usps*, *Amazon SQS*, *Amazon SimpleDB* und *Inventory Repository* [WI12]. Abhängig vom Ergebnis des *Inventory Repository* Prozessschritts kommt der *Rule Service* zum Einsatz. Schließlich ist die *Process Services* Schicht zu nennen, welche die konsistente und korrekte Ausführung des aktuellen Prozesses überwacht. In der SOA-Applikation geschieht dies bei der Abfrage zur Verfügbarkeit von Artikeln (Prozessschritt: *Inventory Repository*). Dies bedeutet, dass eine später gestartete, jedoch möglicherweise schneller abgearbeitete Bestellung eine zuvor gestartete Bestellung nicht negativ beeinflussen darf. In der SOA-Applikation ist diese Koordinationsaufgabe dem übergeordneten Prozessschrittes *Order Processing* zuzuordnen. Zusammenfassend hat die Analyse der *Information Systems* Dimension ergeben, dass lediglich die vier oberen Schichten in WebAutoParts.com abgedeckt sind. Zu den anderen Schichten konnten keine passenden Artefakte gefunden werden, die eine Existenz der jeweiligen Schicht erahnen lassen.

Die *Technology* und *Information Systems* Dimensionen stehen in Abhängigkeit zueinander. Die *Technology* Dimension umfasst die Hardware und Software Ressourcen, welche zur Bereitstellung von Geschäfts-, Daten- und Applikationsservices notwendig sind. Dementsprechend umfasst diese Dimension Standards für die Bereiche Infrastruktur, Middleware, Netzwerk, Kommunikation und Verarbeitung. Die Schichten innerhalb der Technology Dimension beinhalten die Schichten Datenbanksystem, TP-Monitor und MQ System, Applikationsserver, Enterprise Service Bus (ESB), Rule-Server und Prozessserver. Weiterhin decken zusätzliche Schichten den Bereich für Kollaboration und Interaktionsdienste ab. Dazu gehören: Interaction Framework, Portal und Workflowserver, Kollaborationsframework und Choreographie-Framework. Auch eine Security-Schicht wird abgebildet [Zi11]. Bei der Abbildung dieser Dimension auf die SOA-Applikation wurde festgestellt, dass nicht alle Schichten abgedeckt werden. Da innerhalb von WebAutoParts.com auch intern Daten gespeichert werden müssen, wird eine Abdeckung der Datenbanksystemschicht angenommen. Auch der Applikationsserver wird als eine Schicht identifiziert, die abgedeckt ist. Durch den Applikationsserver wird eine lokale Laufzeitumgebung bereitgestellt. Über den gesamten Prozess hinweg werden verschiedene Webservices orchestriert. Demgemäß können auch die Schichten ESB und Prozessserver klassifiziert werden. Aufgrund der begrenzten Darstellung der SOA-Applikation in [Wi12] konnten keine weiteren eindeutig abgedeckten Schichten identifiziert werden. Es wird davon ausgegangen, dass auch eine Abdeckung der Schichten *Workflowserver* und *Security* besteht. Weiterhin wird angenommen, dass die *Workflowserver*-Schicht das Zusammenspiel zwischen dem Bestellungsprozess und weiteren internen Prozessen beschreibt. Da mit sensiblen Daten (Kundendaten) gearbeitet wird, wird eine Abdeckung der *Security*-Schicht vorausgesetzt. Folglich müssen gewisse Sicherheitsaspekte eingehalten werden, welche durch die *Security*-Schicht zentral

bereitgestellt werden können. Zu den verbleibenden Schichten der Technologie Referenzarchitektur konnten keine Annahmen zur Klassifizierung gemacht werden, somit werden diese als nicht abgedeckt angesehen.

4 Fazit

Die Klassifizierung der SOA-Applikation WebAutoParts.com wurde anhand der Beschreibung und des Chevron-Diagramms aus [Wi12] durchgeführt. Es konnte gezeigt werden, dass insbesondere innerhalb der *Information Systems* Dimension eine höhere Abdeckung der Schichten festgestellt werden kann. Für die Dimensionen *Business & Informations* und *Technology* mussten zur Klassifizierung teilweise Annahmen getroffen werden. Es wurde zudem eine bestehende Abhängigkeit zwischen den Dimensionen *Information Systems* und *Technology* gezeigt. Schichten, die in erst genannter Dimension ermittelt werden konnten, sind in der *Technology* Dimension durch entsprechende Technologien abgedeckt. Insgesamt betrachtet konnte trotz der abstrakten Darstellung der SOA-Applikation eine erste Klassifizierung mit Hilfe des ESARC vorgenommen werden.

Eine spezifischere Klassifizierung könnte durch eine detaillierte Beschreibung und Darstellung der Applikation und Geschäftsumgebung wie beispielsweise Strategie, Ziele und Organisation vorgenommen werden. In einem weiteren Schritt könnten den verschiedenen Dimensionen des ESARC eine Gewichtung zugeordnet werden, um eine detailliertere Klassifizierung der SOA-Applikation zu ermöglichen. Solche weiterführenden Untersuchungen ermöglichen es, die Qualität der SOA-Applikation zu erfassen und Verbesserungen konkret anzusetzen. Dies könnte zum Beispiel durch eine vergleichende Klassifizierung von anderen SOA-Applikationen geschehen.

Literaturverzeichnis

[Wi12] Wilde, N.: The WebAutoParts.com SOA Application. University of West Florida, Pensacola, 2012.

[Zi11] Zimmermann, A; Hans-Jürgen, Groß; Gunther, Piller; Helge, Buckow; Oliver F. Nandico; Karl, Prott: Capability Diagnostics of Enterprise Service Architectures using a dedicated Software Architecture Reference Model. IEEE - International Conference on Services Computing, Washington, 2011; S. 592-599.

[OG09] TOGAF "*The Open Group Architecture Framework*" Version-9, The Open Group, 2009.

[ZZ11] Zimmermann, A.; Zimmermann, G.: ESARC - Enterprise Services Architecture Reference Cube for Capability Assessments of Service-oriented Systems. IARIA - International Academy, Research, and Industry Association, Rom, 2011; S. 63-66.

Modellierung von Hardwareplattformen für die modellgetriebene Softwareentwicklung

Andreas Dann

adann@mail.uni-paderborn.de
Fraunhofer-Institut für Produktionstechnologie IPT
Projektgruppe Entwurfstechnik Mechatronik, Softwaretechnik
Zukunftsmeile 1, 33102 Paderborn

Abstract: Die Anzahl der Funktionen, die durch Software realisiert werden, steigt in mechatronischen Systemen stetig. Daher werden zur Softwareentwicklung für mechatronische Systeme modellgetriebene komponentenbasierte Methoden genutzt, die es erlauben, Funktionen in Softwarekomponenten zu kapseln. Zur Ausführung müssen die Softwarekomponenten zu Steuergeräten (ECUs) allokiert werden. Zur Allokation müssen die Eigenschaften einer Hardwareplattform bekannt sein, um zu bestimmen, ob Ressourcenanforderungen der Softwarekomponenten durch die ECUs erfüllt werden. Daher wurde im Rahmen einer Bachelorarbeit ein Hardware Plattform-Beschreibungsmodell (PDM) erarbeitet, das die hierarchische Modellierung einer Hardwareplattform ermöglicht und benötigte Informationen komponentenbasiert für die Allokation bereitstellt.

1 Einleitung

Mechatronische Systeme sind alle Arten von Systemen, welche mit ihrer Umwelt über Sensoren und Aktoren interagieren und häufig in sicherheitskritischen Umgebungen eingesetzt werden. Beispiele für sicherheitskritische mechatronische Systeme sind Fahrerassistenzsysteme im PKW, die mithilfe von verschiedenen Sensoren, Aktoren und Steuergeräten (ECUs), auf denen die Software ausgeführt wird, ihre Funktionen realisieren. Der innovationstreibende Faktor von mechatronischen Systemen ist die Software, wodurch deren Anteil stetig steigt. Aufgabe der Software innerhalb eines mechatronischen Systems ist die Organisation der Kommunikation verschiedener ECUs sowie die Durchführung von Steuerungs- und Überwachungsaufgaben mittels Sensoren und Aktoren unter sicherheitskritischen Anforderungen. Durch den steigenden Anteil der Software steigt die Komplexität der Softwareentwicklung.

Um der steigenden Komplexität zu begegnen, wird die Software für mechatronische Systeme modellgetrieben entwickelt. Bei der modellgetriebenen Softwareentwicklung [SVEH07] wird die Software in Form von formalen Modellen spezifiziert, die von der technischen Realisierung abstrahieren. Hierdurch wird die Komplexität der Modelle reduziert und deren Verständlichkeit erhöht [SVEH07]. Weiterhin erlaubt die Abstraktion von der technischen Realisierung die Verifikation der Modelle mittels ModelChecking [CGP06]. Die

verifizierten Modelle werden abschließend zur Generierung lauffähiger Software genutzt [SVEH07].

Eine Methode der modellgetriebenen Softwareentwicklung für mechatronische Systeme ist MECHATRONICUML [BBB+12]. MECHATRONICUML bietet einen Entwicklungsprozess, eine Modellierungssprache sowie Konzepte zur Verifikation und Simulation für die Softwareentwicklung mechatronischer Systeme. In MECHATRONICUML erfolgt die Modellierung der Software in Form von Softwarekomponenten, deren Verhalten und Kommunikation unter Beachtung von Echtzeitanforderungen spezifiziert werden. Anschließend erfolgt im Entwicklungsprozess von MECHATRONICUML die Verifikation des Verhaltens und der Kommunikation bezüglich sicherheitskritischer Anforderungen.

Zur Ausführung müssen die Softwarekomponenten zu ECUs allokiert werden. Allerdings stellen verschiedene Softwarekomponenten unterschiedliche Anforderungen an ECUs, beispielsweise bezüglich Speichergröße, Netzwerkgeschwindigkeit und Ausführungszeit, die abhängig von der Rechengeschwindigkeit der ECU ist. Jedoch setzt sich die Hardwareplattform eines mechatronischen Systems aus mehreren, zu Teilnetzwerken organisierten, ECUs mit unterschiedlichen Eigenschaften zusammen [WHW09]. Somit sind zur Bestimmung einer Allokation von Softwarekomponenten zu ECUs Informationen über die Hardware notwendig. Die benötigten Informationen über die Hardware des mechatronischen Systems werden durch Plattform-Beschreibungsmodelle (PDMs) spezifiziert. Daher wurde im Rahmen einer Bachelorarbeit im Bereich Softwaretechnik des Fraunhofer-Instituts für Produktionstechnologie ein Plattform-Beschreibungsmodell zur Modellierung von hierarchischen, verteilten Hardwareplattformen für MECHATRONICUML umgesetzt.

2 Konzept der Hardware Plattform-Beschreibung

Das erarbeitete PDM beinhaltet ein Konzept zur hierarchischen, parametrierbaren Modellierung einer Hardwareplattform und zur Spezifikation von Eigenschaften, die zur Allokation notwendig sind. Die Erarbeitung eines Konzepts und die Bestimmung der notwendigen Eigenschaften basiert auf den Arbeiten MARTE [OMG11], OMG D&C [OMG06], DeSi [MMBM04] und Zeller et. al [ZP12]. Im Unterschied zu den betrachteten Arbeiten wurde das Konzept der komponentenbasierten Softwareentwicklung zur Modellierung von Hardwareplattformen adaptiert und das PDM als eigenständiges Metamodel mithilfe des Eclipse Modeling Frameworks (EMF) umgesetzt[1]. Durch die Adaption der komponentenbasierten Softwareentwicklung, wird die flexible Wiederverwendung einmal modellierter Elemente der Hardwareplattform in unterschiedlichen Konfigurationen ermöglicht.

Im PDM wird zwischen Hardwareressourcen und Hardwareplattformen unterschieden. Hardwareressourcen repräsentieren logische Einheiten einer Hardwareplattform. Zu diesen Hardwareressourcen werden die Softwarekomponenten allokiert. Hardwareressourcen stellen entsprechend ihrer Semantik unterschiedliche Funktionen und Kapazitäten zur Ausführung von Software bereit, z.B. ECUs, die spezifische Speicher und Prozessoren besitzen. Hardwareplattformen setzen sich aus mehreren vernetzten Hardwareressourcen

[1] `https://trac.cs.upb.de/mechatronicuml/wiki/GI2014`

zusammen. Sie repräsentieren die Plattform eines Systems, z.B. die Plattform eines PKWs, die aus mehreren ECU-Netzwerken besteht.

Weiterhin beinhaltet das erarbeitete PDM ein Typen-/Instanzkonzept zur Parametrierung bereits modellierter Hardwareressourcen und Hardwareplattformen. Auf Typebene werden Kardinalitäten und Attribute für Hardwareressourcen bzw. Hardwareplattformen spezifiziert. Diese werden dann auf Instanzebene konkretisiert. Mithilfe der Kardinalitäten lässt sich die Variabilität einer Produktlinie modellieren, z.B. ein PKW, der in verschiedenen Ausstattungen eine unterschiedliche Anzahl an ECUs besitzt. Zur Trennung der verschiedenen Aspekte erfolgt die Plattform-Beschreibung aus vier unterschiedlichen Sichten (engl. Views). Die vier Views der Plattformmodellierung werden in Abbildung 1 anhand eines Netzwerks aus ECUs, wie es beispielsweise in PKWs zu finden ist, dargestellt.

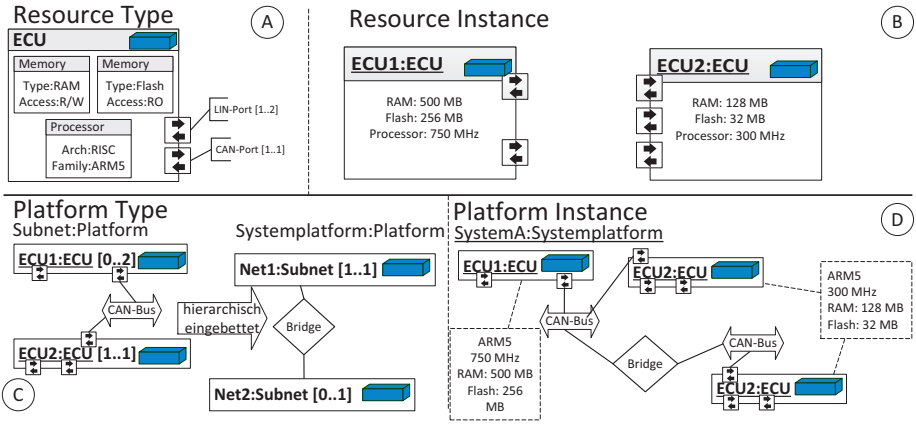


Abbildung 1: Konzepte des Hardware Plattform-Beschreibungsmodells

Die *View Resource Type* Ⓐ dient zur Beschreibung von Hardwareressourcen auf Typebene. So wird beispielsweise in der *Resource Type View* der Typ ECU modelliert, der sich aus den Ressourcen Flash-, Arbeitsspeicher und Prozessor zusammensetzt. Weiterhin erfolgt die Modellierung verschiedener Hardware-Ports, die eine Kardinalität besitzen.

In der *Resource Instance View* Ⓑ wird der Typ ECU zu verschiedenen ECUs instantiiert, die sich in Speicher- und Rechenkapazität unterscheiden. Weiterhin wird die Anzahl der Hardware-Ports der verschiedenen Ressourcen Instanzen entsprechend der in Ⓐ spezifizierten Kardinalität festgelegt.

Die *Platform Type View* Ⓒ dient der hierarchischen Modellierung einer Hardwareplattform. So können Teilnetzwerke der Hardwareplattform modelliert werden, die aus einer variablen Anzahl vernetzter ECU-Instanzen bestehen. So lassen sich beispielsweise unterschiedliche Ausstattungsvarianten eines PKWs mithilfe von Kardinalitäten modellieren. Das modellierte Teilnetzwerk wird schließlich als Teil einer hierarchischen Hardwareplattform modelliert. Diese Hardwareplattform setzt sich aus mehreren solchen Teilnetzwerken zusammen, die über Netzwerkbrücken verbunden sind. Entsprechend den spezifizierten Kardinalitäten erfolgt in der *Platform Instance View* Ⓓ die Instantiierung einer konkreten Hardwareplattform.

3 Zusammenfassung und Ausblick

Das erarbeitete PDM wurde anhand eines Netzwerks aus Mindstorms und Arduino Boards erprobt. Dabei zeigte sich, dass sich der logische Aufbau der Plattform widerspiegeln lässt. Ebenso ermöglicht die Umsetzung eines Typen-/Instanzkonzepts die Wiederverwendung bereits modellierter Elemente bei der Modellierung gleichartiger Plattformen. Allerdings wurde bei den Arduino Boards ersichtlich, dass sich der physikalische Aufbau insbesondere der Hardware-Pins nur begrenzt durch Hardware-Ports widerspiegeln lässt. Daher ist es notwendig, in späteren Arbeiten zur Bestimmung einer Allokation und zur Generierung von Code zu evaluieren, inwieweit die Beschreibung des physikalischen Aufbaus, z.B. Hardware-Pins, benötigt wird.

Weiterhin ist es in zukünftigen Arbeiten notwendig zu evaluieren, inwieweit das erarbeitete PDM erweitert werden muss, um benötigte Informationen zur Durchführung einer Schedulability Analyse und einer Worst-Case-Execution-Time Analyse des für eine Hardwareplattform generiertem Codes zu ermöglichen.

Danksagung

Mein Dank gilt meinem Erstgutachter Steffen Becker und meinem Betreuer Uwe Pohlmann, die es mir ermöglichten, meine Bachelorarbeit umzusetzen und mich durch zahlreiche Anregungen unterstützten.

Literatur

[BBB+12] Steffen Becker, Christian Brenner, Christopher Brink, Stefan Dziwok, Christian Heinzemann, Uwe Pohlmann, Wilhelm Schäfer, Julian Suck und Oliver Sudmann. The MechatronicUML Design Method – Process, Syntax, and Semantics. Bericht, Software Engineering Group, Heinz Nixdorf Institute University of Paderborn, 2012.

[CGP06] Edmund M Clarke, Orna Grumberg und Doron Peled. Model checking, 2006.

[MMBM04] Marija Mikic-Rakic, Sam Malek, Nels Beckman und Nenad Medvidovic. A Tailorable Environment for Assessing the Quality of Deployment Architectures in Highly Distributed Settings. In *Component Deployment*, Jgg. 3083 of *LNCS*, Seiten 1–17. Springer Berlin Heidelberg, 2004.

[OMG06] OMG. Deployment and Configuration of Component-based Distributed Applications Specification. Bericht April, OMG, 2006.

[OMG11] OMG. UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems. Bericht June, 2011.

[SVEH07] Thomas Stahl, Markus Völter, Sven Efftige und Arno Haase. *Modellgetriebene Softwareentwicklung*. dpunkt-Verlag, Heidelberg, 2. Auflage, 2007.

[WHW09] Hermann Winner, Stephan Hakuli und Gabriele Wolf. *Handbuch Fahrerassistenzsysteme*. Vieweg+Teubner, Wiesbaden, 2009.

[ZP12] Marc Zeller und Christian Prehofer. Modeling and efficient solving of extra-functional properties for adaptation in networked embedded real-time systems. *Journal of Systems Architecture*, Dezember 2012.

Integration von Informationen über die Bodenbeschaffenheit in das eNav-System

Masterarbeit

Dženan Džafić Dominik Franke

{dzafic, franke}@embedded.rwth-aachen.de

Abstract: Eine effiziente Nutzung der Akkukapazität ist für Fahrer von Elektrofahrzeugen erstrebenswert. Die Kalkulation des Energieverbrauchs führt besonders für Nutzer von Elektrorollstühlen zur Steigerung der Mobilität, da der Fahrer einen potentiellen Ausfall des Akkus aufgrund fehlender Energie verhindern kann. Für die Berechnung energieeffizienter Routen ist eine Vielzahl von Einflussfaktoren zu betrachten, die den Energieverbrauch beeinflussen. Anschließend an die Einbeziehung der Steigung, die bereits in das eNav Navigationssystem integriert ist, widmet sich diese Abschlussarbeit der Bodenbeschaffenheit der Fahrbahnen, indem der Rollwiderstand der Straßenbeläge mit in die Verbrauchsfunktion integriert wird. Abschließende Tests weisen im Durchschnitt eine Steigerung der Effizienz nach.

1 Einleitung

Stromsparende Maßnahmen sind aufgrund von erhöhten Energiepreisen sehr beliebt, da die Akkukapazität deutlich besser ausgeschöpft wird. Energiesparmodi schaffen es, den Verbrauch deutlich zu reduzieren. Ein Navigationssystem, das durch Berücksichtigung von Einflussfaktoren wie der Steigung, dem Bodenbelag oder der Temperatur einen Rückgang des Energieverbrauchs führt, folgt diesem Trend auf eine besondere Weise. Das Navigationssystem eNav bedient sich einer Verbrauchsfunktion, die von einer Modifikation des A*-Suchalgorithmus genutzt wird. Diese Arbeit erweitert die bestehende Verbrauchsfunktion, welche bereits die Steigungsinformationen einbezieht, um den Faktor Bodenbeschaffenheit. Das Ergebnis ist ein Navigationssystem, das dem Fahrer durch die Berechnung von energieeffizienten Routen assistiert und eine bessere Planung ermöglicht [DF13].

Üblicherweise beschränken sich Navigationssysteme bei der Berechnung auf den kürzesten oder den schnellsten Weg. Solange der Energieverbrauch vernachlässigt wird, reicht die Entfernung zwischen den Streckenpunkten aus. Die Berechnung energieeffizienter Routen erfordert allerdings weitere Faktoren wie Steigung und Bodenbelag. Openstreetmap (OSM) bietet frei erhältliches Kartenmaterial, welches über Schnittstellen für den Bodenbelag und Höheninformationen verfügt. Die damit verbundenen Informationen sind allerdings nicht besonders detailliert, weshalb eine weitere Datenquelle genutzt wird. Der Stadtbetrieb Aachen stellt für Forschungszwecke genauere Informationen über den Bo-

denbelag bereit. Diese liegen in einem anderen Format vor als das OSM-Kartenmaterial, weshalb eine Zuordnung der Straßenbeläge zu den Straßen nötig ist. Zusätzlich sind für die Rollwiderstände konkrete Werte festzulegen. Das Navigationssystem eNav benutzt eine mit Höhendaten und Bodenbelaginformationen angereicherte OSM-Datei. Die Höheninformationen werden zur Berechnung der Steigung verwendet und die Bodenbeläge legen den Rollwiderstand fest. Die Verbrauchsfunktion berücksichtigt diese Faktoren bei der Berechnung einer Route mit minimalem Stromverbrauch. Als Ergebnis liefert das Navigationssystem neben der kürzesten und der schnellsten Route eine energieeffiziente Route.

2 Grundlagen

Das Kartenmaterial für das eNav Projekt besteht aus mehreren Bestandteilen. OpenStreetMap (OSM) legt den Grundstein für das Navigationssystem eNav. Das Kartenmaterial, welches seit der Entstehung im Jahr 2004 kontinuierlich erweitert und verbessert wird, ist mit der Zielsetzung entstanden, die gesamte Welt digital abzubilden [RT10]. Die Nutzer können dabei jederzeit Informationen hinzufügen oder editieren. Eine enorme Datendichte schafft eine wunderbare Grundlage, die alle wichtigen Elemente wie Straßen, Gebäude und Points of Interests (POI) enthält. Es besteht zudem die Möglichkeit Informationen in Form eines Tags hinzuzufügen. Beispielsweise sind die Tags *alt* für die Höhe und *surface* für die Straßenoberfläche nützlich. Für eigene Zwecke kann das Kartenmaterial auch um weitere Tags und Informationen angereichert werden.

OpenTripPlanner (OTP) ist ein quelloffener, multimodaler Routenplaner, der alle Funktionalitäten eines herkömmlichen Navigationssystems mitbringt. Aufgrund der Tatsache, dass der zuvor verwendete WebService ORS kein Open-Source-Projekt ist, siehe [DF13], übernimmt OTP dessen Rolle. OTP bietet die Möglichkeit, Schnittstellen selbstständig anzupassen und ist bezüglich der Berechnung der Steigung flexibler [DFBK13]. Ein dreidimensionales Kartenmaterial lässt sich leicht erzeugen. Die Routenberechnung wird mit Hilfe des A*-Suchalgorithmus vollzogen, wobei einige vordefinierte Heuristiken angewendet werden. Diese lassen sich anpassen und für die Berechnung des Verbrauchs unter der Berücksichtigung des Bodenbelags erweitern.

Der A*-Suchalgorithmus berechnet den kürzesten Pfad von einem Startknoten A zu einem Endknoten B. Die Eingabe für den Algorithmus ist ein Graph, der aus Kanten und Knoten besteht und ein Straßennetz modelliert. Der A*-Algorithmus zeichnet sich besonders dadurch aus, dass die Suche vorzeitig terminiert, wenn eine obere Schranke, die durch eine Heuristik festgelegt wird, unterboten wird. In der Regel wird die Luftlinie als Parameter für die Heuristik angewendet. Ein Kriterium für die Heuristik ist, dass diese nicht durch einen reellen Wert unterboten werden darf.

3 Das Navigationssystem eNav

Das Navigationssystem eNav ist über zwei Schnittstellen erreichbar, nämlich über einen mobilen Client als Android-App und über ein auf OPT basierendes Webinterface. Beide Schnittstellen sind an die Bedürfnisse von eNav angepasst. Nach der Eingabe eines Start- und Zielknotens, eines Normalverbrauchs, der maximalen Geschwindigkeit und einer maximal zu bewältigenden Steigung lässt sich eine Routenberechnung initiieren. Der Normalverbrauch ergibt sich über die maximale Akkukapazität geteilt durch die maximale Reichweite, welche aus dem Handbuch des Elektrorollstuhls zu entnehmen sind. Der eNav Server nimmt die Parameter entgegen und startet die Berechnung mit einer Modifikation des A*-Algorithmus. Der Energieverbrauch wird über folgende Formel bestimmt:

$$\text{Verbrauch} = \text{Distanz} \cdot Normalverbrauch \cdot 1,15^{\text{Steigung}} \cdot \text{Reibungsfaktor}$$

Der Fokus dieser Arbeit liegt auf dem Reibungsfaktor, welcher mit den Parametern in der Verbrauchsfunktion multipliziert wird. Der Teilausdruck für die Steigung sowie die Heuristik sind ausführlich in [DFBK13, DF13] beschrieben. Die Heuristik bleibt für diese Arbeit unverändert, da der Reibungsfaktor auf Null gesetzt wird. Dies ist aufgrund der Tatsache zulässig, dass kein Reibungsfaktor eine senkende Wirkung auf den Energieverbrauchs hat. Somit wird die Korrektheit der Heuristik durch die Hinzunahme der Bodenbeschaffenheit nicht beeinträchtigt [DFBK13].

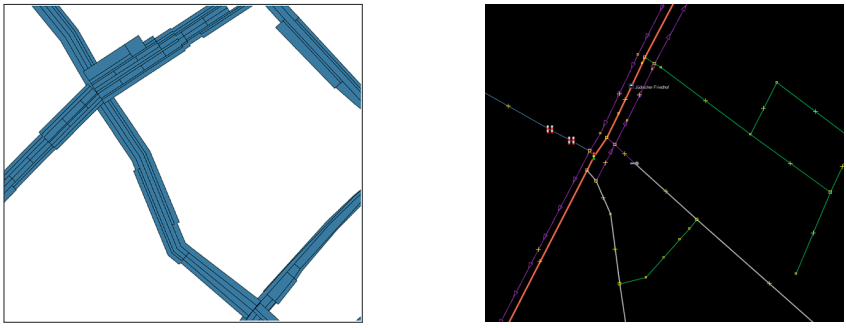


Abbildung 1: Ansicht QGIS (links) und JOSM (rechts)

Die Datenquelle des Stadtbetriebs Aachen liegt als Shapefile vor, das OSM-Kartenmaterial hingegen in Form einer XML-Datei. Ein Ausschnitt aus beiden Dateiformaten ist in Abbildung 1 zu sehen. Die beide Formate unterscheiden sich insofern, dass Straßen im Shapefile aus Polygonen mit mehreren Spuren bestehen, im OSM-Kartenmaterial jedoch aus Kanten. In OSM wird häufig ein Fahrradweg neben einer Straße als Zusatztag an einer Kante gespeichert. Somit ist eine Strategie erforderlich, welche aus den Spuren der Polygone den passenden Straßenbelag auswählt. Da keine Möglichkeit besteht, jede einzelne Spur in OSM zu übertragen, wird in einer Kante nur der beste Belag gespeichert. Die schlechteren Straßenbeläge werden verworfen. Aus diesem Grund muss davon ausgegangen werden, dass der Fahrer in der Lage ist, selbstständig den optimalen Belag zu befahren. Ein

Shapefile	OSM-Tag	Faktor
Asphalt/Beton	asphalt/concrete	1,00
Großer Natursteinpflaster	cobblestone	1,05
Kleiner Natursteinpflaster	cobblestone:flattened	1,02
Betonpflaster	paving_stones	1,01
Plattenbelag	concrete_plates	1,02
Unbefestigte Fläche	pebblestone	1,07

Tabelle 1: Zuteilung von Reibungsfaktoren

weiterer Schwerpunkt ist die Ermittlung und Zuordnung der auf einer Studie basierenden Reibungskoeffizienten [SS03]. Tabelle 1 listet die Faktoren auf, wobei Asphalt als Basis und somit mit dem Faktor 1 angenommen wird.

4 Fazit

Die bereits in das Navigationssystem eNav integrierten Einflussfaktoren Steigung und Bodenbelag zeigen, dass es möglich ist, durch diese zusätzlich zum gängigen Kartenmaterial hinzugefügten Informationen den berechneten Energieverbrauch zu senken. Ein Test über 100.000 zufällig generierte Routen ergibt in 42% aller Berechnungen, dass die effizienteste Route nicht der kürzesten Route entspricht, was einem beachtenswertes Ergebnis gleichkommt. Die Verbrauchsfunktion ist noch verbesserungswürdig und muss durch Testfahrten validiert werden. Weitere Einflussfaktoren, die es ermöglichen den Energieverbrauch zu minimieren sind zukünftig zu untersuchen. Die Ausweitung auf andere Elektrofahrzeuge ist ebenfalls ein vorgesehener Schritt, der besonders unter Berücksichtigung der Energierückgewinnung sehr ertragreich zu sein verspricht.

Literatur

[DF13] Dzenan Dzafic und Dominik Franke. Entwicklung und Evaluation eines Navigationssystems für Elektrorollstühle. In *Gesellschaft für Informatik Seminars, Informatiktage*, Seiten 185–188. Gesellschaft für Informatik e.V., 2013.

[DFBK13] Dzenan Dzafic, Dominik Franke, Danni Baumeister und Stefan Kowalewski. Modifikation des A*-Algorithmus für energieeffizientes 3D-Routing. In *Angewandte Geoinformatik 2013 - Beiträge zum 25. AGIT-Symposium (AGIT)*, Seiten 414 – 423. Wichmann Verlag, 2013.

[RT10] Frederik Ramm und Jochen Topf. *OpenStreetMap - Die freie Weltkarte nutzen und mitgestalten*. Lehmanns, dritte. Auflage, 2010.

[SS03] Tobias Schmidt und Dirk Schlender. Untersuchung zum saisonalen Reifenwechsel unter Berücksichtigung technischer und klimatischer Aspekte. Bericht, Bergische Universität Wuppertal, 2003.

SOA Technologie Architektur am Beispiel Open Source JBoss

Dimitrios Buzungidis, Andreas Etüs, Dominik Kurz, Tobias Wankmüller

Hochschule Reutlingen
Fakultät Informatik
Architecture Reference Lab
Alteburgstraße 150
72762 Reutlingen

Dimitrios.Buzungidis | Andreas.Etues | Dominik.Kurz | Tobias.Wankmueller
@student.reutlingen-university.de

Abstract: Im ersten Schritt geht es um die Serviceorientierte Technologie Architektur am Beispiel des führenden Produktes WebSphere einer kommerziellen Lösung von IBM. Hierbei werden die Bestandteile Process Server, Enterprise Service Bus, Messaging und Persistenz erläutert und zu einer Gesamtheit abgestimmt. Nachfolgend wird die Open Source Lösung JBoss mit all ihren Komponenten sowie dessen Kommunikationsablauf erklärt und ebenfalls grafisch dargestellt.

1 Einführung

Um die einzelnen Komponenten von Serviceorientierter Architektur (SOA) in der Theorie besser zu verstehen, ist es im ersten Schritt sinnvoll eine kommerzielle Produktfamilie im Detail genauer zu betrachten, da diese sich in der Literatur qualitativ durchgesetzt hat, bevor die Herausforderung sich an einer Open Source Lösung zu orientieren, durchgeführt wird. Dafür sollte ein einheitliches Verständnis für den Begriff SOA geschaffen werden. Unter SOA wird ein Architekturmuster der Informationstechnik aus dem Bereich der verteilten Systeme verstanden. Dabei wird SOA als Methode angesehen, die z. B. Datenbanken, Server und Websites als einzelne, unabhängige Komponenten verwaltet und deren Kommunikation untereinander gewährleistet. Darüber hinaus erfolgt über die Orchestrierung die Zusammenfassung der Dienste. Hierbei wird eine flexible Kombination mehrerer Services zu einer Komposition verstanden. Diese beschreibt abschließend einen ausführbaren Geschäftsprozess. Zu guter Letzt wird anhand der flexiblen Kombinations- und Austauschmöglichkeiten von SOA eine langfristige Senkung von Kosten in der Softwareentwicklung sowie das Erreichen einer höheren Flexibilität von Geschäftsprozessen erzielt.

2 SOA Komponenten

2.1 WebSphere Process Server

Wie bereits in der Einführung geschildert, wird im ersten Schritt das kommerzielle Produkt untersucht, um die einzelnen Bestandteile der technischen Seite von SOA in ihrer Theorie zu verstehen. Das Unternehmen IBM hat die Produktfamilie WebSphere auf den Markt gebracht. Dabei stellt der WebSphere Applikationsserver das Grundgerüst für die Funktionalitäten des eigentlichen Process Servers dar.

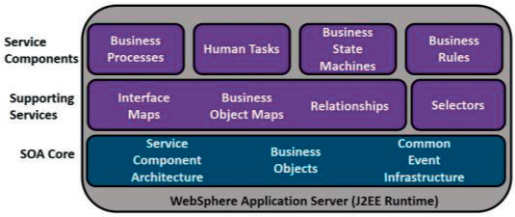


Abbildung 1: WebSphere Process Server

Dieser ist in drei Schichten untergliedert. Die unterste Ebene stellt dabei den SOA Kern dar, in welchem die Spezifikationen des SOA-Modells beschrieben werden. Des Weiteren werden innerhalb dieser Schicht die Implementierungen des Interfaces für externe Schnittstellen vorgenommen. Die darauf aufbauende Supporting Services Schicht verwaltet dabei die Verbindung zu den heterogenen Komponenten und sorgt für die Vereinheitlichung der Kommunikationssprache, sodass eine einheitliche Kommunikation stattfinden kann. Abschließend befasst sich die Service Components Schicht mit der Verwaltung und der Vernetzung von Geschäftsprozessen [1].

2.2 Enterprise Service Bus

Beim kommerziellen Produkt der Firma IBM beinhaltet die Supporting Services Schicht die Funktionalitäten und Aufgaben eines Enterprise Service Buses (ESB). Aus diesem Grund wird versucht die Theorie nicht anhand des WebSphere Produktes zu erläutern, sondern durch eine weitere Recherche in der Literatur. Mit ESB wird in der Informationstechnik eine Kategorie von Softwareprodukten beschrieben, welche die Integration von verteilten Diensten in einem Unternehmen organisieren. Dabei stellt der ESB einen gemeinsam genutzten Kommunikationsbus dar, der die Kommunikation von Punkt-Zu-Punkt-Verbindungen zwischen Clients überflüssig macht und von allen Dienstanbietern und Nutzern zentral genutzt wird. Die Hauptaufgabe besteht darin, als Adapter zwischen den unterschiedlichen Clients zu wirken und die verschiedenen Nachrichten zu empfangen, zu übersetzen und zu verschicken. Abgerundet kann gesagt werden, dass der ESB mit dem Schlagwort „Kommunikationsinfrastruktur“ gleichgesetzt werden kann.

2.3 Messaging und Datenhaltung

Unter Messaging wird die Art der Kommunikation der einzelnen Komponenten mit dem ESB verstanden. Hierbei gibt es unzählige, gängige Ausprägungen, wohingegen sich der Java Messaging Service (JMS) im internen Unternehmensalltag, aufgrund dem hohen Maß an Kompatibilität durchgesetzt hat. Weitere Vorteile für die Durchsetzung von JMS sind der asynchrone Support, die Möglichkeit der Übertragung von großen Datenpaketen sowie der ausgeprägten Unterstützung von Middlewares. Für die Interaktion mit externen Unternehmenspartnern hat sich aufgrund der großen Bekanntheit und der einfachen Client-Implementierung der HTTP-Standard durchgesetzt. Die SOA-Architektur stellt einen eigenen, integrierten Speicher zur Datenhaltung zur Verfügung, um den Verlust von Daten vorzubeugen, da im Regelfall eine Datenamnesie herrscht. Eine SOA ohne diese Komponente funktioniert nur auf der untersten Ebene. Hierbei gehen eine Fülle an Daten und Metadaten verloren mit denen die Entwicklung einer Gesamtlösung vereinfacht hätte werden können.

3. Open Source JBoss

Nachdem nun der theoretische Ansatz von SOA anhand des kommerziellen Produkts WebSphere verstanden wurde, kann die Recherche einer Open Source (OS) Lösung in diesem Bereich umgesetzt werden. Aufgrund einer hohen Anzahl von unterschiedlichen OS Lösungen erfolgte eine Differenzierung bzw. Fokussierung auf eine Komplettlösung. Das Produkt JBoss der Firma RedHat repräsentiert solch eine Komplettlösung und ist dabei folgendermaßen aufgebaut. Die Registry, die Orchestration sowie die Rules befinden sich in der obersten Schicht der JBoss Enterprise SOA Plattform. Wie bereits beim WebSphere veranschaulicht, stellt die zweite Schicht den ESB dar. Dieser dient hier ebenfalls als Kommunikationsinfrastruktur. In der untersten Ebene befindet sich der JBoss Applikationsserver (AS), welcher ein System zur Verwaltung von Objekten und Komponenten für verteilte Anwendungen darstellt. Die großen Stärken des JBoss sind die Verbergung der Implementierungsdetails, die Sicherung der Persistenz von Komponenten sowie die Unterstützung von verteilten Transaktionen. Der Applikationsserver bietet ebenfalls ein hohes Maß an Flexibilität, sodass die Persistenz je nach Wunsch von der eigenen Bean und damit dem Administrator oder direkt vom Container und damit dem AS verwaltet wird. Ebenfalls ist ein gleiches Maß an Flexibilität innerhalb der Transaktionsbehandlung gegeben. Der Client kann beim JBoss im Namensverzeichnis (JNDI) eine Objektreferenz ermitteln und sich im Anschluss am eigentlichen Server anmelden. Im Anschluss bekommt der Client eine Interface Objektinstanz zurück, um mit diesem via Interface kommunizieren zu können. Die zur Verfügung gestellten Methodenaufrufe werden über die Services direkt ausgeführt. An dieser Stelle wird die Serviceorientierte Architektur besonders deutlich und kann in der nachfolgenden Abbildung visuell betrachtet werden [2].

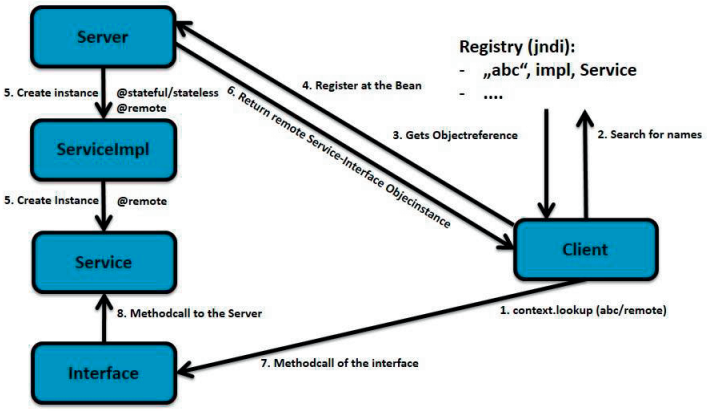


Abbildung 2: Kommunikationsablauf am Beispiel JBoss

Abgerundet wird durch das separat im Foliensatz aufgeführte Konfigurationsbeispiel gezeigt, dass die JBoss Lösung alle benötigten Komponenten einer Serviceorientierten Architektur beinhaltet und dem Anwender eine Allround-Lösung für private und kommerzielle Zwecke zur Verfügung stellt.

4. Zusammenfassung und Ausblick

Betrachtet man das Potenzial der Open Source Lösung hinsichtlich des kommerziellen Produkts der Firma IBM, so wird sehr schnell deutlich, dass die Open Source Lösung bereits zu einem sehr hohem Maß ausgereift ist, sodass eine OS Lösung aufgrund der enormen Kostenersparnis oftmals eine gelungene Alternative zur kommerziellen Lösung darstellt. Ebenfalls muss bei der OS Lösung auf keine elementare Komponente verzichtet werden, sondern bietet im Gegenzug ebenfalls die Möglichkeit den JBoss um weitere OS Komponenten zu erweitern. Im Großen und Ganzen muss es nicht immer die teurere, kommerzielle Lösung für eine Serviceorientierte Architektur für das Anbieten von Services für Clients sein.

5. Literaturverzeichnis

[1] http://www.ibm.com/developerworks/websphere/library/techarticles/0509_kulhanek/0509_kulhanek.html (Zugriffsdatum: 20.12.2013)

[2] https://docs.jboss.org/author/display/AS7/Getting+Started+Developing+Applications+Guide (Zugriffsdatum: 22.12.2013)

Projekt-Reviews am Beispiel von IT-Projekten kleinerer Unternehmen

Simon Flaiz, Stefan Geiselhart, Marc Prokop, Alexander Schlegel und Thomas Wiest
(vorname.nachname@student.reutlingen-university.de)

Projektarbeit an der Hochschule Reutlingen
Fakultät Informatik

Betreuer der Arbeit: Prof. Dr. rer. nat. Alfred Zimmermann

Abstract: Diese Ausarbeitung setzt sich mit den Qualitätssicherungsmaßnahmen vor Projektstart und dem Monitoring während eines Projekts auseinander. Anhand eines Soll-Ist-Vergleiches an einem real durchgeführten Projekt werden die Elemente aus der Ist-Situation eruiert und konsolidiert. Erkannte Defizite und suboptimale Elemente des Projekts werden inspiziert und durch theoretische Ansätze und Modelle versucht zu eliminieren. Ziel ist eine projektoptimale Konstruktion eines Soll-Konzepts, welches auf zukünftige analoge Projekte adaptiert werden kann.

1 Einleitung

Einer im Juni 2012 von Gartner durchgeführten Studie zufolge, sind rund die Hälfte aller aufgetreten Fehler in IT-Projekten unabhängig von der Projektgröße. Die aufgetretenen Fehler lassen sich in zwei Segmente untergliedern. Zum einen können substantielle Verspätungen durch eine fehlerhafte Zeitplanung oder unzureichend durchgeführte Risikoanalysen des Projekts entstehen. Ein anderer Aspekt sind Funktionsfehler, die aufgrund von Missachtung der Qualitätssicherungsmaßnahmen entstehen können [G12].
Es stellt sich daher die Frage, welche Präventionen diesbezüglich getroffen werden können. In den nachfolgenden Kapiteln wird ein real durchgeführtes IT-Projekt mittlerer Größe inspiziert und analysiert. Das IT-Projekt wurde bei der Feil, Feil & Feil GmbH mit Sitz in Ludwigsburg durchgeführt. Das Unternehmen ist eine Full-Service-Internetagentur, die derzeit zwanzig Mitarbeiter beschäftigt. Zu den Kernaufgaben des Unternehmens gehören unter anderem die Entwicklung von Webanwendungen, App-Entwicklung auf verschiedenen etablierten mobilen Betriebssystemen, sowie das Screen-/Print-Design. Der Fokus dieser Ausarbeitung liegt dabei auf der Durchführung eines Soll-Ist-Vergleiches, mit der Berücksichtigung der Modelle und theoretische Ansätze aus relevanten Literaturen. Bei der Analyse werden die Aspekte der Qualitätsmaßnahmen vor Projektstart und die Verfolgungsmöglichkeiten während des Projektes eruiert und analysiert, inwiefern diese eingehalten wurden und an welcher Stelle Verbesserungspotenzial besteht. Die Schlussbetrachtung der gewonnenen Informationen gewährt einen möglichen Ausblick, wie die entdeckten Defizite oder suboptimalen Elemente in zukünftigen analogen Projekten vermieden werden können.

2 Verwandte Arbeiten

Ein Qualitätsplan umfasst die Bereitstellung von Qualitätszielen und stellt zugleich ein Framework für die Erreichung beziehungsweise Kontrollmechanismen dieser Ziele bereit [W90]. Demnach impliziert der Einsatz eines Qualitätsplans zugleich dessen Kontrolle und Prüfung, was beispielsweise in Form von Reviews geschieht. [R90] beschreiben ein Review als einen Prozess oder ein Meeting in welchem Artefakte oder Produkte einem bestimmten Empfängerkreis präsentiert und mit diesem diskutiert werden. Die einschlägige Literatur [G04] beschreibt vielfältige Review-Methoden und -Typen, die alle kontext- und projektabhängig eingesetzt werden.

3 Ist-Analyse und Soll-Konzept

Im Rahmen unserer Untersuchung des vorliegenden Projekts konnten wir die Anwendung verschiedener Qualitätsmanagementmethoden feststellen. Einige der Pre-Project-Methoden wurden hierbei nur teilweise oder überhaupt nicht angewandt. Vor allem im Bereich des Project-Review wurden substantielle Defizite festgestellt.
Im folgenden Abschnitt wird das Verbesserungspotential in den Bereichen Pre-Project und Project-Review aufgezeigt, sowie ein optimierter methodischer Ansatz aufgeführt.

3.1 Review-Methodik

Im betrachteten Projekt wurden innerhalb von ad-hoc-Meetings funktionale und vertraglich festgelegte Kundenanforderungen getestet und abgenommen. Die Review-Meetings wurden flexibel nach Bedarf und ohne vordefinierte zeitlich und inhaltlich festgelegte Meilensteine abgehalten.
Bei näherer Betrachtung der Meetings aus einer analytischen Perspektive fallen hinsichtlich einer adäquaten Review-Methodik [W11] negative Aspekte auf. Es fehlen unter anderem eine strukturierte Form der Reviews, zeitlich festgelegte Abläufe, Entwicklungsstandards und die Verbindung zu erforderlichen Dokumenten, wie dem Design-Document oder einem Pflichtenheft. Somit liefern die bisher angewandten ad-hoc-Meetings dem Projekt keine optimale Tracking- beziehungsweise Kontrollfunktion. Daraus ergibt sich die Notwendigkeit eine neue Review-Methodik für das untersuchte Projekt zu entwickeln. Diese Methodik soll für zukünftige Projekte mit vergleichbarem Projektcharakter adaptierbar sein.
Im Folgenden wird das Konzept der angepassten Review-Methodik des Projekts erläutert. In der Pre-Project Phase werden zeitlich und inhaltlich verbindliche Meilensteine definiert und in die Perioden 1 bis n eingeplant. In diesen Perioden werden jeweils Reviews abgehalten, in denen alle Review-Aktivitäten durchgeführt werden. Der Fokus wird anhand dieser Methodik auf die frühzeitige Identifikation der Abweichungen von vorab definierten Entwicklungsstandards gelegt. Zudem soll eine frühzeitige Fehlererkennung eingeführt werden, die als Tracking- beziehungsweise Kontrollfunktion [W11] für das Projekt dient.

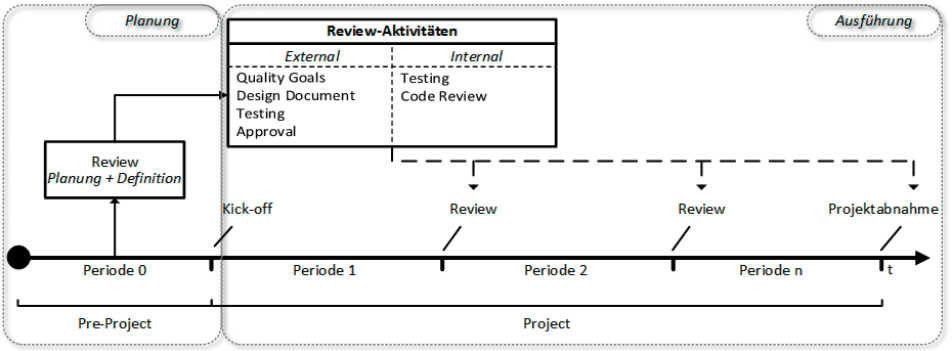


Abbildung 1: Review-Konzept

Im Folgenden werden selektierte Elemente aus dem Qualitäts- und Entwicklungsplan in repetitiven Reviews nach der zuvor dargestellten Methodik behandelt und deren Verbesserungspotential aufgezeigt. Durch die Symbiose mit der Review-Methodik können diese Elemente eine deutliche Verbesserung der Qualität innerhalb des aufgezeigten Softwareprojekts bewirken.

3.2 Qualitätssicherungsaktivitäten/Review-Aktivitäten

Das untersuchte Projekt berücksichtigte lediglich kundenspezifische Anforderungsfunktionen, wohingegen definierte Qualitätsziele fehlten. Diese sind nötig um einen frühzeitigen Indikator für Abweichungen zu erhalten und Gegenmaßnahmen einzuleiten. Qualitätsziele werden nach [G04] als Quality Goals verstanden. Durch sie ist es möglich, frühzeitig und mithilfe gezielter Anwendungen Qualitätsstufen zu testen und diese zu implementieren.
Solche Anforderungen, Zeiteinheiten und Vorgehensweisen für die individuelle Softwareprojektentwicklung sollten in einem grundlegenden Dokument erfasst sein. Das Dokument wird nach [G04] als Design-Document betitelt und als fundamentales Begleitdokument der Review-Meetings und des Softwareentwicklungsprozesses abgehandelt. Neben den detaillierten Anforderungen des Kunden sollten auch die Projektplanung mit Meilensteinen, Vorgehensweisen und Review-Terminen beinhaltet sein, um angestrebte Qualitätsebenen zu halten beziehungsweise zu erreichen.
Im Projektverlauf fanden ausschließlich manuelle Softwaretests statt und Code-Reviews wurden nicht durchgeführt. Des Weiteren wurden keine internen Programmierstandards definiert. Um diesen Defiziten entgegenzuwirken, werden systematisierte Tests, definierte und vorgegebene Programmierstandards, sowie kontinuierliche Code-Reviews eingeführt. Dadurch erfolgt eine frühzeitige Identifikation und Eliminierung von Fehlern mit simultaner und signifikanter Steigerung der Softwarequalität [G04]. Die zeitnahe Einführung eines Continuous-Integration-Systems ist daher empfehlenswert. Mit Hilfe dieses Systems kann eine automatisierte und kontinuierliche Überprüfung der definierten Programmstandards stattfinden. Darüber hinaus können automatisierte Softwaretests ausgeführt werden. Die aus dem System resultierenden Ergebnisse dienen den Reviewern als wertvolle Ausgangsinformationen für ein Code-Review.

Für den weiteren Projektfortschritt sind innerhalb jeder Review-Session Abnahmen durchzuführen, die für den weiteren Verlauf entscheidend sind. Hierbei stehen Funktionalität und Umfang der für die entsprechende Session festgelegten Workpackages im Vordergrund, welche vom Kunden angenommen oder abgelehnt werden. Je nach Entscheidung sind im Nachgang Korrekturmaßnahmen beziehungsweise Überarbeitungen notwendig. Andernfalls wird nach erfolgreicher Abnahme die planmäßige Fortführung der Entwicklung und des Projekts eingeleitet.

4 Schlussbetrachtung

Nach Beleuchtung und Untersuchung der eingesetzten Qualitätsmethoden und -aktivitäten vor und während dem Projekt, konnte auf dieser Basis zuerst deren Bewertung und nachgehend die Optimierung in Form eines Soll-Zustandes vollzogen werden. Den erkannten Defiziten und fehlenden Bestandteilen wurde auf diese Weise gezielt entgegengewirkt. Ein für nahezu alle Softwareprojekte sehr pervasives Steuerungs- und Kontrollelement, die Review-Methodik, wurde als elementarer Bestandteil neu modelliert und in eine für das behandelte Projekt effiziente und passende Ziel-Review-Methodik transformiert.
Bei unserem Modell steht planerische Sicherheit sowie Abstimmung und Relevanz der Review-Inhalte im Mittelpunkt. Dies wollen wir bereits in der Pre-Project Phase initialisieren. Der gesamte Projektverlauf wird hinsichtlich der Review-Sessions zeitlich und inhaltlich verbindlich definiert und geplant. Ungeplante und unzureichend vorbereitete ad-hoc-Meetings mit dem Review-Team werden hiermit vermieden. Review-Sessions werden auf Basis von JIRA-Daten strukturiert durchgeführt und gesteuert. JIRA dient zunehmend als Input-Lieferant hinsichtlich Entwicklungsständen einzelner Softwaremodule, dem Gesamtprojektfortschritt als auch den Ergebnissen von Softwaretests. In Zukunft sind diese essentiell wichtigen Inhalte zentral in JIRA aufzufinden und wesentlicher Bestandteil sämtlicher Review-Sitzungen im Projekt.

Literaturverzeichnis

[G12] Mieritz L.: Survey Shows Why Projects Fail, 01.07.2012, https://www.gartner.com/doc/2034616/survey-shows-projects-fail, Zugriff am 21.01.2014

[W90] Westland, J.: The Project Management Lifecycle, Kogan Page Limited, 1. Auflage, London, 2006; S. 74

[R90] Radatz, J.: IEEE Standard Glossary of Software Engineering Terminology, The Institute of Electrical and Electronics Engineers,New York1990; S. 64

[G04] Galin, D.: Software Quality Assurance, Pearson Education Limited, 1. Auflage, Essex, Great Britain, 2004

[W11] Wallmüller, E.: Software Quality Engineering – Ein Leitfaden für bessere Software-Qualität, 3. Auflage, München, 2011, Hanser Verlag; S. 86, 98

Eine DSL zur Modellierung von Tests für Automatisierungsanwendungen

Matthias Jurisch, Michael Pötz

Labor für Verteilte Systeme, Hochschule RheinMain
Unter den Eichen 5, 65195 Wiesbaden
matthias.jurisch@gmail.com, michaelpoetz@gmail.com

Art der Arbeit:	Master-Projekt (Matthias Jurisch), Master-Thesis (Michael Pötz)
Betreuer:	Prof. Dr. Reinhold Kröger

1 Einleitung

Kleine und mittelständische Unternehmen (KMU) der Automatisierungsbranche wollen für ihre Software schlanke Entwicklungsprozesse verwenden. Häufig geht dies mit proprietären Modellen einher, die über Jahre gewachsen sind. So ist es auch bei der Eckelmann AG (Wiesbaden). Dort verwendet man sogenannte *Sequential Function Tables* (SFT) zur Beschreibung des Verhaltens von Automatisierungskomponenten. Sie stellen Automaten in einer tabellarischen Notation dar. Auf einem Automatisierungsknoten können mehrere SFT-Instanzen ausgeführt werden. Verteilte Anwendungen verwenden mehrere Knoten, deren ggf. hierarchisch strukturierte SFT-Instanzen miteinander kommunizieren. Bei der Codegenerierung wird ein Skelett der Automaten generiert. Bedingungen, Aktionen und Kommunikationsbeziehungen zwischen Automaten muss der Entwickler händisch umsetzen. Dieser Vorgang ist fehleranfällig und muss daher, insbesondere im Rahmen von Regressionstests nach Programmänderungen, getestet werden. Die Wirtschaftlichkeit der Entwicklung erfordert ein möglichst hohes Maß an automatisierten Abläufen. Die Automatisierung des Testprozesses erfolgt durch eine Adaption des Testframeworks TPTP[1] und ist nicht Gegenstand dieser Arbeit.

Die vorliegende Arbeit wurde als ein in ein Forschungsprojekt der Hochschule integriertes studentisches Projekt durchgeführt. Zur Beschreibung der Tests wurde von den Autoren eine domänenspezifische Sprache (*Domain Specific Language* – DSL) entwickelt, in der sich der Aufbau der Tests und das Soll-Verhalten von verteilten Automaten sowohl für ein plattformunabhängiges Automatenmodell als auch für SFTs modellieren lässt. Die mit der DSL modellierten Testfälle werden über Modelltransformationen in eine generische Repräsentierung, die für TPTP ausführbar ist, umgewandelt. Diese Repräsentierung orientiert sich an UTP [OMG05], um eine Wiederverwendbarkeit in anderen Tools zu gewährleisten.

[1] Eclipse Test & Performance Tools Platform

Zur Laufzeit validiert ein generiertes Testorakel das tatsächliche Verhalten (Ist-Verhalten) des *System Under Test* (SUT) gegen das in den Testfällen spezifizierte gewünschte Verhalten (Soll-Verhalten).

Um die Wirtschaftlichkeit der Testmodellierung sicherzustellen, sollen die Systemmodelle beim Entwurf der Tests weiterverwendet werden. Gängige Standards wie TTCN-3 [ETS13] bieten diese Möglichkeit nicht. Auch spezialisiertere Ansätze aus der Automatisierung basieren nicht auf formal über Automaten beschriebenen Systemen [WFS12], [BK08], oder betrachten keine verteilten Automatensysteme [LMNS05].

Im Abschnitt 2 wird das Konzept der erstellten DSL vorgestellt sowie die Funktionsweise des Testorakels erläutert. Die Implementierung des Konzepts und die verwendeten Technologien sind im Abschnitt 3 dargestellt. Abschnitt 4 beschreibt den aktuellen Stand der Arbeiten und gibt einen kurzen Ausblick.

2 Konzept

Im Testframework werden alle am Test beteiligten SFTs im *Deployment-Modell* verwaltet. Für jede SFT werden auch ihre Zustände und Transitionen dort angegeben. Neben dem Deployment-Modell existiert für jeden Testfall eine *Testfallbeschreibung*, die das Soll-Verhaltensmodell beinhaltet. Dieses referenziert die SFTs, Zustände und Ereignisse aus dem Deployment-Modell. Deployment-Modell und Testfallbeschreibung werden in der entwickelten DSL erstellt.

Die Soll-Verhaltensmodelle können verschiedene Aspekte umfassen. Als wesentlicher Aspekt kann Soll-Verhalten als *Pfad* über Zuständen einer SFT in der DSL beschrieben werden. Die Beschreibungssprache umfasst Möglichkeiten zur Modellierung von einfachen Zustandsfolgen, Alternativen, Verboten und Wildcards. Getrennt davon kann das Soll-Verhalten durch Prädikate über Automaten, Zuständen und Ereignissen beschrieben werden, um auf diese Weise zeitliche Bedingungen oder Häufigkeiten von Eintritten in Zustände zu spezifizieren. Ein Pfad über mehrere Automaten wird als *Globaler Pfad* bezeichnet, auch zeitliche Bedingungen über mehrere SFTs können ausgedrückt werden. Ein Beispiel für einen globalen Pfad mit zwei SFTs *Tuersteuerung_1* und *Schalter_1* ist im folgenden Listing zu sehen.

```
PathAssert Schalter_Alt {
  Schalter_1:Gedrueckt ->
    (Tuersteuerung_1:Schliessend -> Tuersteuerung_1:Geschlossen) or
    (Tuersteuerung_1:Oeffnend -> Tuersteuerung_1:Offen)
}
```

Das Zusammenspiel der verschiedenen Komponenten ist in Abbildung 1 dargestellt. Als Testorakel werden aus der Beschreibung des Soll-Verhaltens nichtdeterministische Automaten generiert, mit denen das Systemverhalten validiert wird. Diese Testautomaten erhalten als Eingabe die beobachteten Zustandsübergänge des SUT. Dabei werden alle Zustandsübergänge der SFTs als *Trace-Events* von einem Trace-Interface übertragen. Es kann angenommen werden, dass diese vollständig und in der korrekten Reihenfolge vom

SUT an das Testframework geliefert werden.

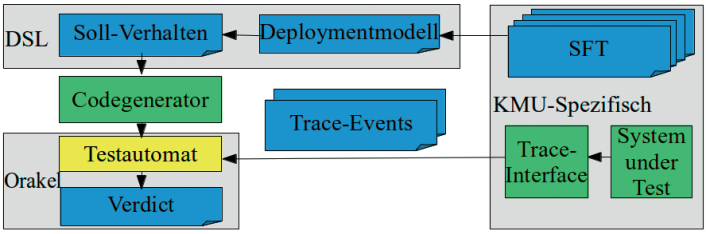


Abbildung 1: Ablauf der Ausführung

Um die Testautomaten auszuführen, wird ein Algorithmus von Ken Thompson [Tho68] zur Verarbeitung von regulären Ausdrücken [Kle51] verwendet. Zu Beginn werden alle Zustände vorgehalten, in denen sich der nichtdeterministische Automat bei Testanfang befinden kann. Diese Menge wird dann als Grundlage für den nächsten Berechnungsschritt verwendet. Beim Lesen eines Zustandsübergangs des SUT wird eine neue Menge errechnet, die alle Zustände enthält, in denen der Automat nach diesem Zustandsübergang sein kann. Enthält diese Menge nach dem Abarbeiten aller erfolgten Zustandsübergänge einen akzeptierenden Zustand, ist der Test erfolgreich.

3 Implementierung

Das beschriebene Konzept wurde prototypisch umgesetzt. Zur Modellierung der SFTs wurde ein Metamodell mit Hilfe des *Eclipse Modeling Frameworks* (EMF) entwickelt [SBPM09]. Weiterhin umfasst dieses Metamodell Elemente zur Modellierung der Testfallbeschreibungen. Durch die Verwendung dieses Frameworks kann mit QVT-Modelltransformationen [OMG08] sehr einfach von proprietären Automatenmodellen verschiedener Hersteller in ein generisches internes Automatenmodell transformiert werden. Zur Implementierung der DSL wurde XText verwendet, da es zur Repräsentierung von Modellinstanzen ebenfalls EMF nutzt und somit eine Integration zwischen KMU-spezifischen Automatenmodellen und Beschreibungen des Soll-Verhaltens möglich ist.

Die in der DSL beschriebenen Modelle werden durch die in XText integrierte Code-Generierung in Java-Klassen umgewandelt, welche die Testautomaten repräsentieren. Diese Klassen bieten zur Testauswertung eine Methode zum Weiterschalten der Testautomaten mittels Trace-Events des SUT. Weiterhin existiert eine Methode, die überprüft, ob der Testautomat noch in einen akzeptierenden Zustand gelangen kann. So kann der Test abgebrochen werden, wenn ein akzeptierender Endzustand nicht mehr erreicht werden kann.

Zur Anbindung an das SUT wurde von der Eckelmann AG ein Trace-Interface bereitgestellt. Das Trace-Interface erlaubt es, alle Zustandsübergänge der am SUT beteiligten SFTs zu abonnieren.

4 Aktueller Stand und Ausblick

Die erste Version der DSL wurde dem Projektpartner vorgestellt und mit den Testern diskutiert. Erste Verbesserungsvorschläge wurden bereits eingearbeitet. Die kooperative Vorgehensweise verspricht eine gute Anpassung an die Bedürfnisse der Tester und eine hohe Akzeptanz.

Als nächster Schritt wird das Testframework an den Projektpartner übergeben, der damit betriebliche Praxistests durchführt. Die Ergebnisse dieser Praxistests werden im weiteren Verlauf des Gesamtprojekts ausgewertet und die sich daraus ergebenden Veränderungen in das Testframework eingearbeitet.

Der vorgestellte Ansatz hat den Vorteil, dass er durch die DSL firmeninterne Modellierungsformen berücksichtigt, das Kernsystem aber trotzdem wiederverwendbar ist. Insbesondere die Möglichkeit, die gleichen Modelle bei der Entwicklung und beim Testen verwenden zu können, ist ein positiver Aspekt, da die Tester sich nicht mit einem vollständig neuen Modell vertraut machen müssen. Außerdem muss durch die Wiederverwendung der Systemmodelle das System beim Erstellen der Tests nicht neu modelliert werden, was unter anderem in wirtschaftlicher Hinsicht Vorteile verspricht. Daher bietet dieses Vorgehen auch Vorzüge für andere Unternehmen, in denen Systeme mithilfe von verteilten Automaten modelliert werden. Der besprochene Ansatz eignet sich allerdings nicht dazu, Fehler im Systemmodell selbst zu finden, dies ist aber auch nicht Ziel dieser Arbeit.

Literatur

[BK08] Eckard Bringmann und A Kramer. Model-based testing of automotive systems. *1st International Conference on Software Testing, Verification, and Validation*, 2008.

[ETS13] ETSI. ES 201 873-1 The Testing and Test Control Notation version 3, 2013.

[Kle51] SC Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, 1951.

[LMNS05] Kim G. Larsen, Marius Mikucionis, Brian Nielsen und Arne Skou. Testing real-time embedded software using UPPAAL-TRON. *Proceedings of the 5th ACM international conference on Embedded software - EMSOFT '05*, Seite 299, 2005.

[OMG05] OMG. UML Testing Profile, 2005.

[OMG08] OMG. Meta Object Facility (MOF) 2.0 Query/View/Transformation, v1.0, 2008.

[SBPM09] David Steinberg, Frank Budinsky, Marcelo Paternostro und Ed Merks. *EMF: Eclipse Modeling Framework 2.0*. Addison-Wesley Professional, 2nd. Auflage, 2009.

[Tho68] Ken Thompson. Programming Techniques: Regular expression search algorithm. *Communications of the ACM*, 11(6):419–422, Juni 1968.

[WFS12] Michael Wahler, Ettore Ferranti und Robin Steiger. CAST: Automating Software Tests for Embedded Systems. *Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference on*, 2012.

Architektur eines Cockpits zur interaktiven Analyse von Enterprise Architectures auf Basis von Viewpoints

Anja Kirchner, Sascha Scheurer, Christian Weber, Anke Wiechmann

Betreuer: Dierk Jugel

Hochschule Reutlingen, Fakultät Informatik, Studiengang Wirtschaftsinformatik
vorname.nachname@student.reutlingen-university.de

Abstract: Die Analyse von Enterprise Architectures ist für die stetige Weiterentwicklung derselben in Unternehmen von essentieller Bedeutung. Zur Unterstützung der Stakeholder bei der Analyse stellen wir eine Architektur für ein interaktives Cockpit vor, dass auf dem Konzept von Viewpoints basiert. Die Kombination mehrerer, miteinander verbundener Viewpoints ermöglicht die Analyse von Problemstellungen.

1 Motivation

Unternehmen stehen vor der Herausforderung, ihre umfangreiche Enterprise Architecture (EA), bestehend aus Geschäftsprozessen, Applikationen, Technologien, der zugrunde liegenden Betriebsinfrastruktur und weiteren Architekturartefakten, zu verstehen und zu beherrschen. Die Architekturartefakte stehen in mannigfaltigen Beziehungen zueinander. Somit fehlt es häufig an Überblick und Transparenz. Mit Hilfe der Management-Disziplin Enterprise Architecture Management (EAM) werden die beschriebenen Probleme gelöst. Neben der organisatorischen Einbettung von EAM werden zur Unterstützung EAM-Werkzeuge genutzt, um die EA zu dokumentieren und in Richtung des idealen Zustands zu planen [Ha13]. Die erfassten Daten können dann durch Visualisierungen in Form von beispielsweise Projektportfolio- oder Bebauungsplangrafiken aufbereitet und dargestellt werden. Zur Weiterentwicklung von EAs sind Analysen notwendig, um Optimierungspotentiale in der Anwendungslandschaft aufzudecken und diese für die Architekturentscheidungen notwendigen Informationen aufzubereiten. Bei Architekturentscheidungen sind verschiedene Stakeholder beteiligt, die über differenziertes Expertenwissen verfügen und somit unterschiedliche Informationen für die Beantwortung von Fragestellungen bezüglich eines Sachverhalts benötigen. Für jede Fragestellung werden spezifische Informationen benötigt. Der ISO Standard 42010 [IS11] definiert das Prinzip von Views und Viewpoints vor dem Hintergrund der Beschreibung von Architekturen. Ein View entspricht dabei einer konkreten Darstellung (z. B. ein Bebauungsplan), wohingegen der Viewpoint die Konstruktion, Interpretation und Nutzung von Views definiert. Ein Viewpoint ist an einen oder mehrere Concerns geknüpft, welchen den Interessen der Stakeholder entsprechen und unterschiedliche Fragestellungen derselben adressieren. Das Konzept eines Cockpits findet in anderen Domänen, bspw. der Überwachung von Kernkraftwerken oder des Bahnverkehrs Anwendung. Dies besteht aus einem Raum mit mehreren Visualisierungsflächen für die Darstellung verschiedener Blickwinkel, die einen

Sachverhalt konkretisieren. Wir möchten das cockpitartige Konzept auf EAM anwenden und die Praxistauglichkeit anhand eines Prototyps nachweisen. Die Architektur des Prototyps ist Gegenstand dieses Aufsatzes. In Abschnitt zwei werden vorhandene Konzepte vorgestellt, die als Grundlage zur Entwicklung des Architektur-Cockpits dienen. Darauf aufbauend wird im dritten Abschnitt die Architektur des Cockpits vorgestellt und dessen prototypische Umsetzung erläutert. Im letzten Abschnitt werden die Ergebnisse reflektiert, die erlangten Erkenntnisse zusammengefasst und ein Ausblick weiterer Ausbaustufen des Architektur-Cockpits gegeben.

2 Stand der Forschung

[Er06] formuliert einen Ansatz, der es ermöglicht, durch Modeltransformationen automatisiert Visualisierungen von Anwendungslandschaften zu erzeugen. Dieses Prinzip wird bereits in dem EAM Werkzeug iteraplan angewendet. [Br11] beschreibt ein Kontrollraumkonzept zur Überwachung von Geschäftsprozessen. Hierbei stehen Geschäftsprozesse und die dafür verwendeten Informationssysteme im Fokus. [Da06] erläutert den Ansatz des Management Cockpit War Rooms, der zur Führung und Steuerung von Unternehmen dient. Dieser besteht aus vier verschiedenen Wänden, die unterschiedliche Blickwinkel auf relevante Sachverhalte zur Entscheidungsunterstützung darstellen. [Ju13] zeigt auf, wie die durch [Er06] vorgestellte Modelltransformation Verwendung finden kann. Dabei werden auf Basis eines Metamodells und Models mit dem Instrumentarium der Softwarekartographie Softwarelandkarten oder Diagramme generiert und dabei zu einem Analyseinstrumentarium weiterentwickelt. Er definiert die Fähigkeit, Stakeholder umfassend bei der Entscheidungsfindung durch ein interaktives Analyseinstrumentarium zu unterstützen, als *Corporate Intelligence*. Weiterhin greift er die Idee des Kontrollraumkonzepts von [Br11] auf und stellt ein Konzept eines EAM-Cockpits vor, das z. B. Funktionen, wie den semantischen Zoom besitzen soll. Während ein herkömmlicher Zoom Ausschnitte vergrößert und verkleinert, ändert der semantische Zoom nicht die Größe, sondern den Detailierungsgrad und die Sichtbarkeit der angezeigten Information, abhängig vom Kontext [Mo97]. Grundlage für das Verständnis der dargestellten Views innerhalb eines Cockpits im Gesamtzusammenhang mit Stakeholdern und der zugrundeliegenden Architektur bildet der ISO Standard 42010 [IS11].

3 Ansatz und Umsetzung

In diesem Abschnitt beschreiben wir eine Architektur zur Umsetzung eines Prototyps für den in [Ju13] beschriebenen Ansatz. Die Betriebsinfrastruktur, die für den Entwurf und die prototypische Umsetzung des EAM-Cockpits benötigt wird, bietet das bereits existierende Management Cockpit an der Hochschule Reutlingen, welches physisch aus 12 Monitoren und einem Smartboard als Hauptmonitor besteht.[1] Es wurde ursprünglich konzipiert, um betriebswirtschaftliche Zusammenhänge eines Unternehmens aus mehreren Blickwinkeln darzustellen und Simulationen für eine bestimmte Fragestellung (z. B. welchen Einfluss eine Erhöhung des Marketingbudgets auf die Absatzzahlen hat) durchzuführen. Abweichend von

[1] Siehe dazu http://www.inf.reutlingen-university.de/studienangebot/studienangebot-master/wi-master/wi-projekte/managementcockpit.html

diesem Konzept möchten wir das Cockpit nicht für Kennzahlen nutzen, sondern für die Darstellung von Zusammenhängen innerhalb einer EA, um den Stakeholdern die Entscheidungsfindung zu erleichtern. Das EAM-Cockpit eignet sich aufgrund der Möglichkeit zur übersichtlichen Darstellung mehrerer Views und der Smartboard-Funktionalität bestens um Architekturanalysen in mehreren Sitzungen vorzunehmen.

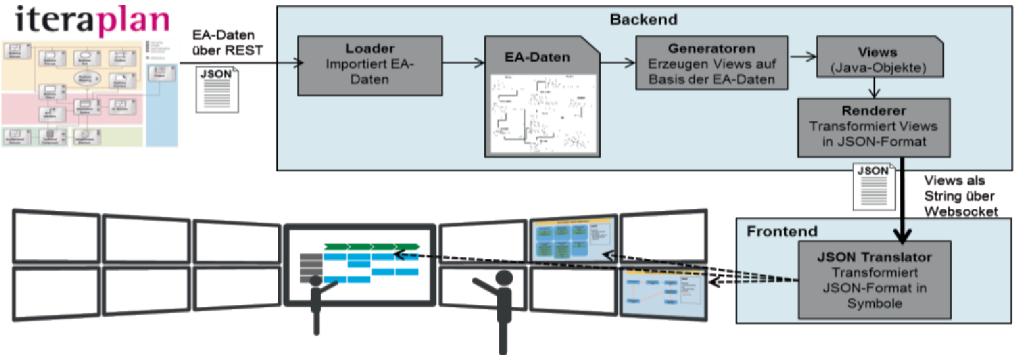


Abbildung 1: Architektur des EAM-Cockpits

Die zugrundeliegende Architektur des EAM-Cockpits ist Abbildung 1 zu entnehmen und besteht aus Backend und Frontend. Die Datenquelle bildet das EAM-Werkzeug iteraplan[2], da es eine Open Source Lösung ist, ein kompaktes Metamodell beinhaltet und Codefragmente für die Generierung von Views wiederverwendet, bzw. auf unsere Bedürfnisse adaptiert werden können. Die Loaderkomponente im Backend ist für das Importieren der in iteraplan abgelegten EA-Daten verantwortlich. Diese können mit Hilfe der von iteraplan zur Verfügung gestellten REST API in Form einer JSON Beschreibung abgerufen werden. Die importierten EA-Daten entsprechen einem Modell, dass die EA eines Unternehmens beschreibt. Intern wird im Backend ein flexibler Metamodellansatz verwendet, sodass verschiedene Datenquellen genutzt werden können. Um später die Views im EAM-Cockpit darstellen zu können, muss definiert werden, wie das EA Modell grafisch repräsentiert werden soll. Die Generatorkomponente führt die in [Er06] beschriebene Modelltransformation ausgehend vom EA Modell in Symbole eines Visualisierungsmodells durch. So soll z. B. ein Informationssystem als Rechteck mit einer Bezeichnung dargestellt werden. Weiterhin gibt es verschachtelte Strukturen wie z. B. die in einer Applikation verbauten technischen Komponenten. Die Applikation wird als Rechteck dargestellt, welches wiederum mehrere Rechtecke enthält, die die verwendeten Technologien repräsentieren. Ein Beispiel wäre die Applikation SAP ERP, welche wiederum die Technologien ORACLE 11g und SAP Netweaver nutzt. Die verschiedenen Objekte des EA Modells (bspw. Applikationen) können hierbei in verschiedenen Views enthalten sein und sollen sich später bei getätigten Interaktionen (bspw. das Hervorheben einer Applikation) in jedem View automatisch anpassen. Dieser Sachverhalt führt dazu, dass auf die beschriebenen Viewpointstrukturen aus [IS11] zurückgegriffen werden muss, um die einzelnen Objekte zu kapseln. Der Renderer nimmt die vom Generator erzeugten Views entgehen und überführt diese in eine JSON Beschreibung, um sie anschließend mittels eines Sockets an das Frontend zu schicken. Wir nehmen eine Trennung von Backend und Frontend vor, da in dem in [Ju13] beschriebenen Konzept eine über

[2] Siehe dazu http://www.iteraplan.de

mehrere Cockpits hinweg verteilte Kollaboration ermöglicht werden soll. Im Anschluss übersetzen wir die Elemente der JSON Beschreibung in Symbole und stellen diese im Cockpit dar. Die Views werden mittels des Grafikframeworks WPF dargestellt. Die Frontend-Komponente ist lose zum Backend-Model gekoppelt, um auch in einem Ausbauszenario mehrere Cockpits zu unterstützen, die auf das gleiche Backend zugreifen. Die Frontend-Komponente verteilt die, zu verschiedenen Views zusammengesetzten, Visualisierungen auf den Bildschirmen des Cockpits.

4 Zusammenfassung und Ausblick

Es wurde eine Architektur für einen Prototyp eines EA-Cockpits beschrieben, welcher im Stande ist, Analysen im Bereich EAM zu ermöglichen. Die Implementierung des Prototyps stellt einen ersten Schritt in Richtung der Umsetzung des Konzepts von [Ju13] dar. Die Architektur soll in den nächsten Schritten verfeinert und durch die Implementierung weiterer Features ausgebaut werden. Die Wahl von JSON als Austauschformat und die Nutzung von Sockets ermöglicht eine plattformunabhängige Implementierung, sodass z. B. flexibel weitere Frontend-Technologien einsetzbar sind. Zu einem späteren Zeitpunkt können so mobile Applikationen entwickelt werden, die zum einen die Steuerung des Cockpits, als auch die Teilnahme an räumlich verteilten Cockpit-Sitzungen ermöglichen. Weiterhin sollen Durchstiche von der Strategie-Ebene, bis hin zu der physischen Hardware, mit der Funktionalität eines Drill-Throughs realisiert werden. Dadurch können z. B. Impacts von Architekturänderungen simuliert werden. Die einzelnen Views sollen annotiert werden können, sodass z. B. bei zeitlich auseinander liegenden Architektur-Sitzungen der gleiche Sachverhalt wieder aufgegriffen werden kann. Dies sieht vor, dass bei Architekturänderungen zwischen Sitzungen, nicht nur die Annotationen, sondern auch die Architektur zum damaligen Zeitpunkt persistiert werden muss.

Literaturverzeichnis

[Br11] Brückmann, T.; Gruhn V.; Pfeiffer, M.: Towards Real-Time Monitoring and Controlling of Enterprise Architectures Using Business Software Control Centers. In ECSA'11 Proceedings of the 5th European conference on Software architecture, Springer, Essen, 2011, S. 287–294.

[Da06] Daum, J.: Management Cockpit War Room – Ziele, Funktionsweise und Zukunftsperspektiven eines (noch) ungewöhnlichen, aber hocheffektiven Managementinstruments. In Controlling, Heft 6, 2006; S. 311-318.

[Er06] Ernst, A.; Lankes, J.; Schweda, C.; Wittenburg, A.: Using Model Transformation for Generating Visualizations from Repository Contents – An Application to Software Cartography. Technische Universität München, 2006.

[Ha13] Hanschke, I.: Strategisches Management der IT-Landschaft : Ein praktischer Leitfaden für das Enterprise Architecture Management. Hanser, München, 2013.

[Ju13] Jugel, D. et. al.: Von der Softwarekartographie zur Corporate Intelligence. In Lecture Notes in Informatics – Informatik 2013, S. 1393-1407.

[IS11] ISO/IEC/IEEE 42010: Systems and Software engineering - Recommended practice for architectural description of software-intensive systems, 2011.

[Mo97] Modjeska, D.: Navigation in Electronic Worlds: A Research Review. Bericht, Computer Systems Research Group, University of Toronto, 1997.

Stepwise Back-in-time Debugging

Vasily Kirilichev[a], Eric Seckler[a], Benjamin Siegmund[a],
Michael Perscheid[b], and Robert Hirschfeld[b]
Hasso Plattner Institute, University of Potsdam, Germany
[a]{firstname.lastname}@student.hpi.uni-potsdam.de,
[b]{firstname.lastname}@hpi.uni-potsdam.de

Abstract: To fully understand how observable failures come into being, back-in-time debuggers provide access to past executions. However, the required run-time analysis is often associated with an inconvenient overhead that renders current tools impractical for frequent use. Potentially large execution histories are expensive to collect and include much data that needs to be analyzed.

Our previously presented stepwise run-time analysis speeds up this analysis by dividing it into multiple steps according to user interaction: A high-level analysis of the method call tree followed by on-demand refinements of object states. This paper advances our approach with a statement-level refinement that allows developers to *step* through large execution histories in forward and backward direction. The corresponding extension of PathFinder, our lightweight back-in-time debugger, provides for instant access to relevant run-time data without collecting needless data up front.

1 Introduction

To better comprehend what causes failures, back-in-time debuggers [Lew03] can help developers by providing access to all required execution details. These debugging tools include every information for describing what happened before observable failures. However, traditional dynamic analysis techniques such as post-mortem debuggers [Lew03] are typically inefficient, time-consuming, and impractical for frequent use. Most approaches capture comprehensive information about the entire execution up-front so that required run-time analysis is often associated with an inconvenient overhead.

Our stepwise run-time analysis [PSH+10] as basis for our lightweight back-in-time debugger named PathFinder [Per13] enables an experience of immediacy that current tools are missing by capturing run-time data only when needed. Low cost can be achieved by dividing the program analysis into multiple runs according to user interaction[1]. Developers immediately retrieve a shallow overview of the execution history that is expanded with user-relevant object states step by step.

Even if our approach enables immediacy characteristics during exploring execution histories, so far it is limited to the method-level. For that reason, this paper presents an extension to our stepwise run-time analysis that enables *stepping* at the statement-level on

[1] We leverage test cases as deterministic entry points into run-time behavior in order to reproduce arbitrary points in a program execution.

demand. Similar to symbolic debuggers, developers can step into and over a statement as well as to its return point, but also back to the previous statement and to the sender. With our PathFinder extension, developers now experience a complete back-in-time debugger which still includes the immediacy characteristics of our original approach.

The contributions of this paper are as follows[2]:

- An extension of our previous stepwise run-time analysis [PSH+10] that allows developers to immediately *step* through statements in execution histories in both forward and backward direction.
- An implementation of this approach as part of our lightweight back-in-time debugger called PathFinder [Per13].

2 Stepping Back and Forth in an Execution History

We describe how we can apply the stepping functionality known from symbolic debuggers to execution histories, providing developers with a familiar interface to navigate the runtime behavior. We present five different stepping actions to developers: *step into*, *step over*, *step return*, *step back*, and *step to sender*. Our approach uses a call tree data structure as representation of the execution history. It represents calling relationships between executed methods. In it, each method call appears as one node. For the stepping functionality, we keep track of the active statement within one of these method call nodes, which can then be highlighted accordingly in the visual representation of the call tree.

The *step into*, *step over* and *step return* functionality is very similar to conventional debuggers. If the currently active statement in the call tree is a call to a method, *step into* sets the first statement in the node of this method call as active statement. *Step over* evaluates statements as a unit and steps to the statement succeeding the active statement in its node. If no further statements exist in the active node, the effect of *step over* equals that of *step return*. *Step return* jumps back to the parent node of the active node and into the statement in the parent node that follows the method call statement corresponding to the active node.

For execution histories, it is possible to add two additional stepping operations. *Step back* is the exact opposite to *step over*. It steps to the statement preceding the active statement. In the case that no further preceding statements exist in the active node, its effect equals that of *step to sender*. *Step to sender*, in turn, is the opposite to *step into*, and is similar to *step return*, but jumps back to the exact method call statement in the parent node of the active node that corresponds to the active node.

[2] A screencast can be found at: `http://www.youtube.com/watch?v=5FtSaZtbUXg#t=210`

3 Stepwise Run-time Analysis at Statements

Our extension of the stepwise run-time analysis follows the same strategy as our original work [PSH+10]. We collect the information necessary for the execution of the stepping actions described before incrementally and on demand. In order to execute the stepping actions, we need to know the sequence of statements executed in the respective call nodes as well as the child nodes that these are corresponding to. Then, we can identify the statement preceding or following another one (necessary for *step over* and *step back*), the child node a method call statement is corresponding to (for *step into*), and the statement in a parent node corresponding to a specific child node (for *step return* and *step sender*).

PathFinder analyzes the execution history by instrumenting the code using method wrappers. We extend PathFinder with a new method wrapper, which is activated only for the refinement analysis necessary to gather the information mentioned above, and only for the method call under investigation. Instead of simply executing the wrapped method, the wrapper starts a simulated interpretation of the Smalltalk method source. This way, it is able to record the order of the individual statements actually executed. Additionally, we wrap called methods and record which statement is currently active in their calling method. So, we establish the relationship between the executed statements in the parent node and its child nodes. The execution of other methods in the call tree is not affected by this selective analysis. Whenever we step into a new method that we did not perform this analysis before, we immediately perform our analysis again for the new node in background.

4 Using the Stepping Mechanisms in PathFinder

Figure 1 shows our PathFinder back-in-time debugger displaying the call tree of a failing test case in the DicThesaurusRex Hunspell library[3]. The assertion in *testSpell* fails because of a bug in the method *DTRCamelCaseParser>>readString*. We use this screenshot to explain the stepping functionality we added to PathFinder.

The stepping analysis can be initialized from any method in the call tree via the *step into call node* button (f). This button initializes the active stepping statement to the first statement in the selected node. The user can then use the buttons in the toolbar to *step over* (a), *step into* (b), *step return* (c), *step back* (d) and *step to sender* (e). The target statements of these actions are illustrated with arrows labeled with the respective upper case letters.

We highlight the active stepping node visually by changing its background color to a darker blue; the active statement is highlighted through text selection (lighter blue background). Additionally, we also highlight the child node corresponding to the active statement (if one exists) with a lighter blue background. In the figure, this is the *readString* call node.

During our initial informal user studies, we could validate that the incremental stepping analysis does not cause any interaction delays noticeable by user. We have been using the feature in a number of existing Smalltalk projects without any limitations.

[3] https://www.hpi.uni-potsdam.de/hirschfeld/trac/SqueakCommunityProjects/wiki/dicThesaurusRex

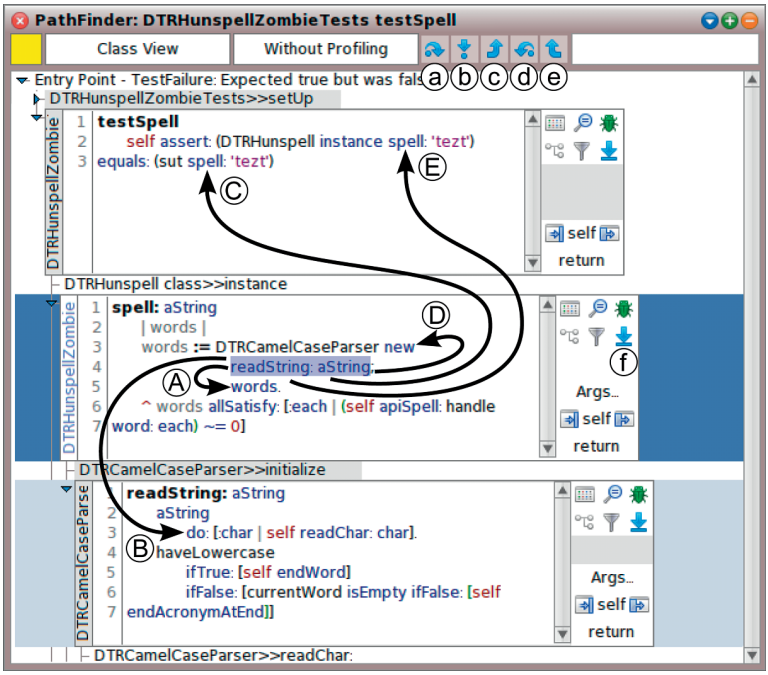


Figure 1: Stepping actions in PathFinder.

5 Conclusion

We have extended our stepwise run-time analysis to support stepping through execution histories. Using an incremental dynamic analysis, we are able to process large execution data step by step, while preserving the immediacy characteristics for developers. This approach makes our back-in-time debugger practical for frequent use. One possible area of future work is to allow developers to change methods directly in the call tree. Therefore, we will investigate adding support for *hot recompilation* to PathFinder.

References

[Lew03] B. Lewis. Debugging Backwards in Time. In *AADEBUG*, pages 225–235, 2003.

[Per13] M. Perscheid. *Test-driven Fault Navigation for Debugging Reproducible Failures*. PhD thesis, Hasso-Plattner-Institute, University of Potsdam, 2013.

[PSH+10] M. Perscheid, B. Steinert, R. Hirschfeld, F. Geller, and M. Haupt. Immediacy through Interactivity: Online Analysis of Run-time Behavior. In *WCRE*, pages 77–86, 2010.

CAELUS: Cloud Architecture for Enabling Mobile Multimedia Services

Dejan Kovachev, Ralf Klamma, Matthias Jarke

Information Systems and Databases
RWTH Aachen University
Ahornstr. 55
52056 Aachen
kovachev@dbis.rwth-aachen.de
klamma@dbis.rwth-aachen.de
jarke@dbis.rwth-aachen.de

Abstract: Mobile multimedia services in the cloud represent a complex interplay of challenging issues including computing models, software architecture, scalability, context, mobility, media-centric operations, user and community oriented design. This paper presents ways to efficiently apply the concepts of the emerging cloud computing paradigm in the design, development and delivery of mobile multimedia services. It describes an information systems architecture called CAELUS (Cloud Architecture for Enabling Mobile Multimedia Services) which includes both conceptual models and a concrete software platform. The contributions comprise a design view, platform and abstraction levels that lower the barrier for mobile multimedia services to leverage the clouds.

1 Introduction

Although many cloud-based products and prototypes have shown the potential to significantly change the IT world, cloud computing benefits are far from being achieved for mobile applications. Similar motivations that had driven cloud computing, are also driving the adoption of mobile multimedia cloud computing, but many new research challenges must be overcome. For example, mobile applications in the cloud involve a trade-off in terms of what should run on the device and what in the cloud, which is contingent to the application type, the device capability, data locality and the operating environment (network bandwidth, delay, cloud availability). Moreover, the traditional server/client programming models fail to provide seamless cloud execution in volatile mobile networks. Furthermore, distant cloud data centers induce prohibitive latency for certain classes of interactive mobile applications such as 3D games and augmented reality. Currently, mobile multimedia cloud computing is at a stage where some technology exists, but there are many opportunities for innovation and for turning the mobile cloud computing into a fruitful paradigm.

This paper presents research work aimed at the integration of mobile and cloud environments where multimedia artifacts play central role. An information system architecture based on

cloud computing principles which facilitates mobile multimedia services is being proposed. The architecture has been realized through several research prototypes. System engineering aspects have been evaluated and the approach has been validated in several domains such as technology-enhanced learning, digital documentation in cultural heritage and human-computer interaction. The rest of the paper is structured as follows. Section 2 provides a requirements overview. In Section 3 the CAELUS architecture is briefly described. The final section concludes this paper.

2 Key Requirements

In general, requirements define the functions of a system, constraints of its operations and specifications of system properties. The key requirements for CAELUS include:

Adoption of the cloud computing paradigm: The whole architecture must follow cloud principles. Moreover, mobile constraints need to be included, too. Computing and storage functions need to be provisioned as a utility-like service model. The minimum criteria for this requirement is the ability of the architecture to elastically and automatically scale with the workload and assigned service policies. In addition, complex setup configurations of software and hardware must be hidden from the cloud service users.

Holistic service-oriented application architecture: The architecture needs to realize a systematic functional decomposition of multimedia applications into reusable cloud services. This requirement must be achieved through a holistic approach that covers end-to-end multimedia life cycle across heterogeneous mobile and Web platforms. The services should follow the service-oriented architecture principles and cloud delivery types.

Development support: The architecture needs to provide development support at different levels in order to embrace different levels of expertise, application flexibility or desired lead time. This requirement is accomplished by providing frameworks, core general services, programming abstraction and models.

Mobile and Web integration: The architecture must be designed around mobile devices from scratch. Mobile clients must not be taken as yet another additional end point, but as primary element around which the cloud services are build. Moreover, mobile clients are not isolated islands, but are interwoven in everyday activities of end-users. Thus, inevitably need to be integrated with the other Web services which also accessed through stationary clients.

Content and metadata management: The cloud platform of the architecture has to provide means for acquisition, sharing, transformation and delivery of multimedia content in various formats and over different networks. Equally important is to support mechanisms for enrichment, contextualization and adaptation of the content via metadata.

3 Mobile Multimedia Cloud Architecture

In addition to the requirements from previous section, the architecture also considers several actors (i.e. stakeholders) including cloud providers, content providers, service providers, end users and communities of practice. For example, service providers need development support in building applications that are usable and suit the needs of communities of practice. The choices of a cloud platform have considerable implications on the design of mobile information systems.

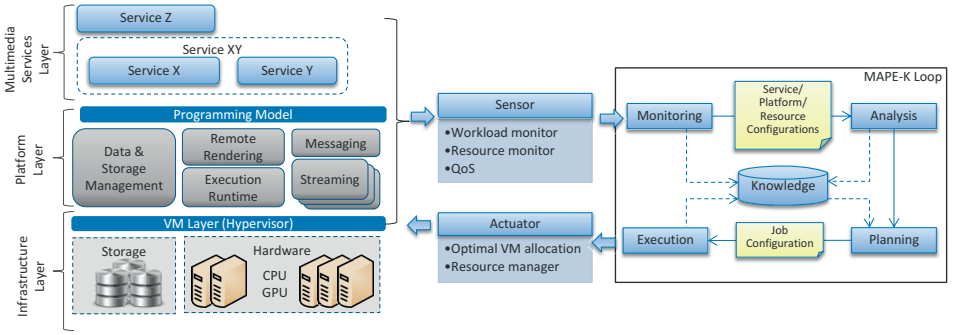


Figure 1: A cloud self-automated management model inspired by the MAPE-K loop

The purpose of any infrastructure is to offer an accessible collection of technologies that serve as foundation for other systems. Therefore, CAELUS was designed to reduce technological complexity for different levels of expertise and flexibility needs of service providers. The CAELUS test bed operates at three main layers: infrastructure, platform and multimedia. These layers, exposed to services providers via APIs, allow full access to the virtualized machines and storage, ability to compose desired applications out of existing components and mash up pre-defined services. The CAELUS approach of offering several abstraction depths to trade-off complexity and flexibility achieves the gentle-slope of complexity [Ber04]. Users need to be able to create small changes in a simple way, whereas more complicated changes should involve only incremental increase in complexity.

CAELUS at the infrastructure level handles the management of large-scale data, adapts to variable workloads, supports dynamic configurations of hybrid cloud applications, etc. At platform level, it employs multimedia specific libraries and complex configurations for various multimedia operations. Figure 1 gives an overview of CAELUS. The virtualized computing and storage infrastructure enables scalable and highly-available multimedia-centric services with easy development. Cloud interoperability achieved with Deltacloud API [Del]. Some of the functionalities of the test bed at this level are exposed as platform services which the application developers can use for more flexible control of the execution environment. The multimedia services layer considers the issues from the mobile multimedia facet. By using the multimedia services, developers can build scalable (mobile) multimedia applications that reflect the user and community requirements.

The different aspects of automatic management multimedia services on cloud architecture can be considered with help of the MAPE-K loop [IBM03] defined by IBM – a widely applied concept in automatic computing [CDTD+12]. It defines control loops in a system to achieve self-configuring, self-healing, self-optimizing and self-protecting features, which are also very relevant in cloud-based systems. The MAPE-K loop consists of the monitor, analyze, plan, execute and knowledge elements.

The right part of Figure 1 depicts a model for automatic management of cloud managed resources for multimedia applications. The model relies on the MAPE-K loop philosophy. Diverse sensors for estimating the system state can be used, e.g. workload, resource usage, and QoS policies. Data and events from sensors at all layers feed the monitoring element which provides service, platform and resource configurations to the analysis element. The analysis element evaluates the actual configuration of all layers involved in the application execution. The findings from this element are forwarded to the planning element and to the knowledge element to be used in subsequent decision processes. The planning element decides on job configurations that give highest utility values. These values are calculated with a utility function defined by some domain expert. In addition, the planning element uses a form of performance model to estimate the service/application execution. For example, a simple form of performance model is one based on historic execution logs. The execution element configures each layer according chosen configurations. For example, it can reserve VM instances, configure them with libraries and initialize them for service execution. It also evaluates the execution which is stored in the shared knowledge element.

4 Conclusions

The CAELUS approach helps service providers to change their development practices by providing means to design and deploy large-scale multimedia applications with less efforts. CAELUS and its core multimedia services realize many of the considerations from the system and multimedia perspectives. Next, the design and system architecture are tested in several mobile scenarios to validate the CAELUS approach.

References

[Ber04] Joerg Beringer. Reducing Expertise Tension. *Communications of the ACM*, 47(9):39–40, 2004. 3

[CDTD+12] Antonin Chazalet, Frederic Dang Tran, Marina Deslaugiers, Alexandre Lefebvre, Francois Exertier, and Julien Legrand. Adding Self-scaling Capability to the Cloud to meet Service Level Agreements. *International Journal On Advances in Intelligent Systems*, 4(3 and 4):180–187, 2012. 4

[Del] Delta Cloud API. [Online] `http://deltacloud.apache.org/`, last accessed: October, 2013. 3

[IBM03] IBM. An Architectural Blueprint for Autonomic Computing. Technical report, 2003. 4

Modellierung und Generierung von Testdaten für Datenbank-basierte Anwendungen

Nikolaus Moll*
Christian Baranowski, Thomas Fox, Jürgen Wäsch
Seerhein-Lab, Konstanz† (www.seerhein-lab.org)
stu@dev.nikolaus-moll.de

Abstract: In dieser Arbeit wird ein Ansatz zur Vereinfachung der Spezifikation von Testdaten für Datenbank-basierte Anwendungen vorgestellt. Dies beinhaltet eine DSL zur einfachen und übersichtlichen Beschreibung von Daten und deren Beziehungen sowie einen Generator zur automatischen Erzeugung von Testdaten.

1 Problemstellung und Ansatz

Softwaretests sind ein wichtiger Baustein für die Qualitätssicherung von Softwareprojekten. Für Tests von Datenbank-basierten Anwendungen müssen u.a. Testdaten für die Datenbank spezifiziert werden, auf deren Basis das Verhalten der zu testenden Software geprüft werden kann. Die Spezifikation dieser Testdaten ist leider augenblicklich sehr umfangreich und komplex und somit aufwändig und fehleranfällig. Die Komplexität ergibt sich v.a. aus der Beschreibung der Beziehungen zwischen den einzelnen Entitäten. Diese unterliegen einer Menge komplexer fachlicher Regeln, die sich aus dem Domänen-Modell und der Geschäftslogik der Anwendung ergeben.

Übergreifendes Ziel der hier beschriebenen Arbeit [Mol13] war es, die Spezifikation von Testdaten für Datenbank-basierte Java-Anwendungen zu vereinfachen. Hierzu wurde zum einen eine geeignete Domänen-spezifische Sprache (DSL) für Testdaten entwickelt. Zum anderen wurde ein Generator zur automatischen Erzeugung von Testdaten implementiert. Basis der Entwicklungsarbeiten war die Java-Bibliothek Simple Test Utils for JUnit & Co. (STU) zur Vereinfachung von Unit-Tests für Java-Anwendungen. STU steht unter der Apache License 2.0 und wird federführend von der SEITENBAU GmbH entwickelt.

Abb. 1 gibt einen Überblick über den gewählten, modellgetriebenen Ansatz. Ausgangspunkt ist eine formale Beschreibung des relationalen Datenbankschemas (Details siehe [Mol13]). Diese kann mittels eines Tools (nicht dargestellt) manuell erstellt bzw. aus einer existierenden Datenbank extrahiert und ergänzt werden. Aus der Schema-Beschreibung wird die Schema-abhängige Testdaten-DSL generiert. Diese DSL kann dann von den Testern genutzt werden, um verschiedene Testdaten-Sets zu beschreiben und diese mittels STU in ihre Unit-Tests einzubinden. Die Testdaten-Sets werden bei den Tests durch das STU-Framework automatisch in die Datenbank eingespielt. Auf Basis der Schema-Beschreibung können auch in der DSL beschriebene Testdaten-Sets generiert werden. Die generierten Testdaten können ggfls. vor Verwendung noch angepasst werden.

*Bis 31.12.2013 Student im Master-Studiengang Informatik / akademischer Mitarbeiter der Hochschule Konstanz (HTWG); seit 15.01.2014 tätig bei der PENTASYS AG in München.

†Das Seerhein-Lab ist eine Kooperation der HTWG Konstanz und der Firma SEITENBAU GmbH.

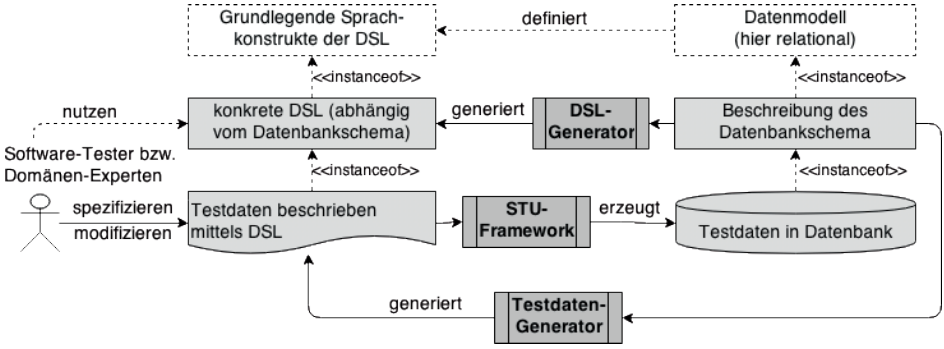


Abbildung 1: Überblick über den gewählten Ansatz.

2 Testdaten-DSL

Es wurden verschiedene Ansätze zur Entwicklung einer DSL für Testdaten untersucht. Der Fokus lag u.a. auf der Fachlichkeit der Datenstruktur, der typsicheren Beschreibung der Testdaten und der einfachen Spezifikation von Beziehungen zwischen Entitäten. Untersucht wurden verschiedene XML-basierte Darstellungen, wie z.B. in DbUnit benutzt, programmatische Spezifikationen und tabellarische Beschreibungsformen. Nach einer Evaluation wurde eine tabellarische Beschreibungsform gewählt. Diese Art der Testdatenmodellierung ist übersichtlich und syntaktisch einfach. Die grundlegende Idee stammt vom Testframework Spock [N+12]. Die EBNF der DSL ist in [Mol13] zu finden.

Listing 1 zeigt beispielhaft die Testdaten-DSL für eine Bücherverwaltung (Datenbankschema siehe Abb. 2 oben). In der tabellarischen Darstellung (`tables`) enthält die erste Zeile die Spaltennamen der Tabelle, die anderen Zeilen enthalten die einzufügenden Daten. Die erste Spalte einer Datenzeile enthält jeweils einen symbolischen Namen (`REF`) für den Tabelleneintrag, der zur Referenzierung und somit Spezifikation von Beziehungen (`relations`) zwischen Datensätzen genutzt werden kann.

Die Implementierung der Testdaten-DSL basiert auf der dynamischen Programmiersprache Groovy und verwendet Laufzeit-Metaprogrammierung in Verbindung mit Operator-Überladen [Mol13]. Die DSL kann eingebettet zusammen mit Java in den Tests (z.B. mit JUnit) genutzt werden und integriert sich sehr gut in gängige IDEs wie Eclipse. Die Spaltennamen sind in der DSL definiert, so dass Autovervollständigung unterstützt wird. Über die REF-Namen können Beziehungen typsicher modelliert und konkrete Werte abgefragt werden. Details zur Implementierung, zur Generierung der DSL für ein Datenbankschema und zur Nutzung der DSL in Softwaretests mit STU sind in [Mol13] zu finden.

3 Generierung von Testdaten

Zielsetzung war die Generierung von möglichst kleinen Testdaten-Sets, die für möglichst viele fachliche Tests verwendet werden können, d.h. eine hohe Testabdeckung bieten. Eine umfassende Literaturanalyse und die Evaluation existierender Werkzeuge ergab, dass bisher keine passende Lösung für diese Anforderung existiert. Aus diesem Grund wurde ein neuer Algorithmus zur Testdatengenerierung entworfen. Anleihen konnten dabei aus [HTW06] gezogen werden. Idee ist, durch Nutzung von Äquivalenzklassen und das gezielte Generieren von Grenzfällen bei Beziehungen eine hohe Testabdeckung zu erreichen.

```
class BookDatabaseGroovyDataSet extends BookDatabaseBuilder {
  def tables() {
    buchTable.rows {
      REF             | name
      CLEANCODE       | "Clean Code"
      EFFECTIVEJAVA   | "Effective Java"
      DESIGNPATTERNS  | "Design Patterns" }
    verlagTable.rows {
      REF             | name
      PRENTICE        | "Prentice Hall International"
      ADDISONWESLEY   | "Addison-Wesley" }
    autorTable.rows {
      REF         | vorname      | nachname
      UNCLEBOB    | "Robert C."  | "Martin"
      BLOCH       | "Joshua"     | "Bloch"
      GAMMA       | "Erich"      | "Gamma"
      HELM        | "Richard"    | "Helm"
      JOHNSON     | "Ralph"      | "Johnson"
      VLISSIDES   | "John"       | "Vlissides"  }
  }
  def relations() {
    PRENTICE.verlegt(CLEANCODE)
    ADDISONWESLEY.verlegt(EFFECTIVEJAVA, DESIGNPATTERNS)
    CLEANCODE.geschriebenVon(UNCLEBOB)
    EFFECTIVEJAVA.geschriebenVon(BLOCH)
    DESIGNPATTERNS.geschriebenVon(GAMMA, HELM, JOHNSON, VLISSIDES)
  }
}
```

Listing 1: Beispiel eines mittels DSL beschriebenen Testdaten-Sets.

Der Algorithmus betrachtet das Datenbankschema als Graph. Tabellen stellen Knoten, Beziehungen (Fremdschlüssel) stellen Kanten dar. Da assoziative Tabellen ebenfalls Beziehungen ausdrücken, werden diese als besondere Kanten behandelt. Ausgehend von einem beliebigen Startknoten werden die Kanten und die damit verbundenen Knoten rekursiv traversiert. Der Algorithmus erzeugt zu jeder Kante Entitäten der beteiligten Tabellen und mindestens Beziehungen für alle Kombinationen der unteren und oberen Grenzwerte, entsprechend den spezifizierten Unter- und Obergrenzen. Gleichzeitig wird versucht, die Anzahl der generierten Entitäten und Beziehungen zu minimieren. Aus diesem Grund werden auch nicht alle, über mehrere Kanten hinweg, mögliche Kombinationen berücksichtigt, da dies zu einer kombinatorischen Explosion führen würde.

Der Algorithmus soll an dem Datenbankschema aus Abb. 2 (UML-Darstellung mit Multiplizitäten) veranschaulicht werden. Als Startknoten wird im Beispiel *Buch* verwendet. Von hier aus werden alle Kanten besucht, hier angefangen mit der Kante (1..1:0..*) zum Knoten *Verlag*. Um möglichst alle Grenzfälle bzw. Äquivalenzklassen abzudecken, wird erzeugt: (1) eine Verlags-Entität, die keine Bücher verlegt, (2) ein Verlag, der genau ein Buch verlegt und (3) ein Verlag, der viele Bücher veröffentlicht[1]. Der Knoten *Verlag* hat keine nichtbesuchten Kanten, weshalb die Traversierung in *Buch* fortgesetzt wird mit der Kante zum Knoten *BuchAutor* (assoziative Tabelle, die eine 0..*:1..*-Assoziation zwischen *Buch* und *Autor* ausdrückt). Der Algorithmus sieht vor, die vier möglichen min/max-Kombinationen zwischen *Buch* und *Autor* zu generieren. Jede Beziehung zwischen einem *Buch* und einem *Autor* resultiert in einer Entität in der Tabelle *BuchAutor*. Existierende Entitäten in *Buch* und *Autor* werden für diese generierten Beziehungen soweit möglich wiederverwendet, bei Bedarf werden neue Entitäten in *Buch* und *Autor* generiert. Die Traversierung des Graph endet nun, da jede Kante besucht wurde. Allerdings sind einige der bis hierhin erzeugten *Buch*-Entitäten noch ungültig, da sie noch nicht zu einem *Verlag* gehören. Solche Entitäten

[1] Für * wird ein konfigurierbarer Wert verwendet, im Beispiel 4.

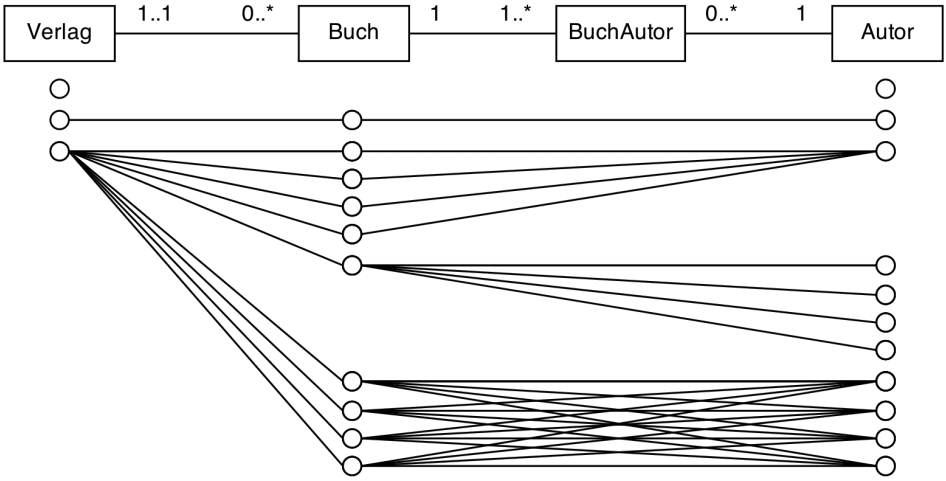


Abbildung 2: Beispiel-Datenbankschema und daraus generierte Entitäten und Beziehungen.

werden im letzten Schritt des Algorithmus behandelt. Alle Entitäten werden auf Gültigkeit bzgl. ihrer Beziehungen überprüft und bei Bedarf werden weitere Beziehungen und weitere Entitäten erzeugt. Abb. 2 stellt das im Beispiel generierte Testdaten-Set (Entitäten und ihre Beziehungen) grafisch dar.

Details zum Algorithmus (Pseudocode) und zur Java-Implementierung sind in [Mol13] zu finden. Evaluationen haben gezeigt, dass die generierten Testdaten unabhängig von der Reihenfolge der Traversierung und bzgl. des Datenbankschemas immer gültig sind. Zur Erzeugung der Werte von Entitäten wurden verschiedene Wertegeneratoren implementiert.

4 Fazit

Die in dieser Arbeit vorgestellten Konzepte und die daraus resultierende Software wurden in das vorhandene STU-Framework integriert. Der Code steht unter der Apache License 2.0 unter `https://github.com/seitenbau/stu/` zur Verfügung.

Die entwickelte Testdaten-DSL wurde bereits in der Qualitätssicherung von mehreren produktiven Softwareprojekten eingesetzt. Der Spezifikationsaufwand und die Fehlerrate konnte im Vergleich zur früheren Vorgehensweise deutlich reduziert werden. Der Testdatengenerator wurde dabei auf Datenbankschemata mit teilweise mehr als 80 Tabellen angewandt. Der Testdatengenerator konnte in jedem Fall einen konsistenten, übersichtlichen Testdaten-Set erzeugen, der eine sehr gute Startbasis für die Anwendungstests ergab.

Literatur

[HTW06] Kenneth Houkjaer, Kristian Torp und Rico Wind. Simple and realistic data generation. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, 2006.

[Mol13] Nikolaus Moll. Testdaten-Modellierung und -Generierung für Datenbank-basierte Anwendungen. Masterthesis, HTWG Konstanz / SEITENBAU GmbH, Oktober 2013. `http://nikolaus-moll.de/Masterthesis.pdf`.

[N+12] Peter Niederwieser et al. *Spock - the enterprise ready specification framework*, 2012. `http://spockframework.org/`.

Elastizitätsszenario von Cloud-Architekturen mit Node.js

Stefan Höfler, Saskia Rettenmeier, Marcel Weiß

vorname.nachname@student.reutlingen-university.de

Fakultät Informatik
Hochschule Reutlingen
Alteburgstraße 150
72762 Reutlingen

Art der Arbeit: Projektarbeit
Betreuer der Arbeit: Michael Falkenthal M.Sc.

Abstract: Im Kontext von Cloud Computing taucht heute ein Begriff immer wieder auf – Elastizität. Elastizität beschreibt generell den Aspekt, dass Anwendungen und IT-Systeme Rechnerressourcen nach Bedarf provisionieren bzw. deprovisionieren können. Viele Anwendungen haben eine monolithische Struktur und ermöglichen keine Elastizität in einem cloudbasierten Umfeld. In diesem Artikel wird beschrieben wie eine Anwendungsarchitektur aufgebaut sein kann die Elastizität ermöglicht. Ein mit Node.js umgesetzter Prototyp dieser Anwendungsarchitektur stellt die Machbarkeit dieses Ansatzes dar.

1 Einleitung

Cloud Computing ist ein neues Paradigma in der IT, bei dem Rechnerressourcen aber auch Softwarefunktionalitäten nach Bedarf konsumiert werden. Die am weitesten verbreitete Definition der fünf essenzielle Charakteristiken von Cloud Computing ist die des National Institute of Standards and Technology. Der „On demand Self-Service“ beschreibt die Provisionierung der Ressourcen. Während sich das „Resource Pooling“ mit der Bereitstellung von Ressourcen in einem Pool beschäftigt. Innerhalb der „Rapid Elasticity“ werden Services schnell und elastisch zur Verfügung gestellt. Der „Measured Service“ ist für die Messung und Überwachung der Ressource zuständig. „Broad Network Access“ bedeutet, dass Services mit Standard-Mechanismen im Netz verfügbar sind ohne an einen bestimmten Client gebunden zu sein [NIST]. An das System und die Anwendungsarchitekturen werden damit neue Anforderungen gestellt, um die definierten Charakteristiken zu implementieren. Cloud Computing führt einige neue Konzepte ein, welche die Art des Aufbaus und der Bereitstellung von Anwendungen verändern. Dazu gehören die Anforderung der Skalierbarkeit und die Charakteristik Elastizität von Anwendungen, die auf verteilten und netzorientierten Infrastrukturen laufen, sowie die Möglichkeit die Anwendung als Dienst im Netz zu betreiben.

2 Motivation

Die Migration von bestehenden Anwendungen, sogenannten Legacy Systemen, ist meistens nicht ohne Weiteres möglich, da traditionelle Anwendungsarchitekturen monolithische Komponenten besitzen, die nur durch hohen Aufwand eine Portierung der Anwendung in eine Cloud Umgebung ermöglichen. Zudem wurden diese Anwendungsarchitekturen nicht auf wechselnde Ressourcenverfügbarkeit ausgelegt. Anwendungen können damit nur unzureichend mit verändertem Ressourcenbedarf umgehen. Damit Anwendungen die Prinzipien von Cloud Computing ausnutzen können, muss deshalb deren Architektur umgestaltet werden, um die Charakteristik wie Elastizität zu erfüllen. Elastizität bedeutet, dass sich zur Verfügung stehende Serverressourcen, wie Rechenleistung, Storage und Netzwerkbandbreite möglichst optimal dem aktuellen Bedarf, d.h. der aktuellen Arbeitslast, angleichen. Eine Anwendung muss daher in der Lage sein aufgrund ihrer aktuellen Auslastung neue Ressourcen durch eine horizontale Skalierung bereitzustellen. Das bedeutet, dass automatisch neue Instanzen der Anwendung gestartet und gestoppt werden können, auf die sich der Workload verteilt. Genau hier muss ein Ansatz gefunden werden, welcher automatisch die neuen Instanzen der Anwendung startet und stoppt, auf die sich der Workload verteilt.

3 Anwendungsarchitektur

In Abbildung 1 ist ein Architekturansatz für einen Webshop dargestellt, der eine horizontale Skalierung im Sinne einer cloudfähigen Architektur zulässt. Der Architekturansatz geht dabei von zwei wesentlichen Komponenten aus. Zum einen der Global Master. Er ist die globale Managementzentrale für die Anwendung und besteht im Wesentlichen aus einer Service Registry und Load Balancer. In der Service Registry werden die aktiven Instanzen verwaltet während der Load Balancer http-Requests engegennimmt und an die Anwendung mittels eines Round Robin Verfahrens weiter routet. Zum anderen die eigentliche Webanwendung, die entsprechend der Auslastung horizontal Skalieren kann. Damit die Anwendung skalieren kann gibt es zusätzlich eine lokale Serverinstanz (Master), als Teil des Skalierungsmechanismus, die dafür zuständig ist Workerinstanzen zu starten und zu beenden.

Ankommende http-Requests werden anwendungsübergreifend vom Load Balancer engegegengenommen. Dieser schlägt in der Service Registry nach, welche Instanzen hinterlegt sind und routet die http-Requests an die entsprechende Workerinstanz weiter. Werden nun zusätzliche Instanzen benötigt kann der Global Master über eine im lokalen Master implementierten REST API zugreifen und neue Instanzen auf Server A starten und beenden. Gestartete Workerinstanzen registrieren sich automatisch in der Service Registry und stehen ab dann für den Load Balancer zur Verfügung. Stehen auf Server A keine Ressourcen mehr zur Verfügung wird dies an den Global Master kommuniziert. Dieser kann nun einen neuen Server (Server B) mit freien Ressourcen ansprechen auf dem weitere Instanzen gestartet werden können.

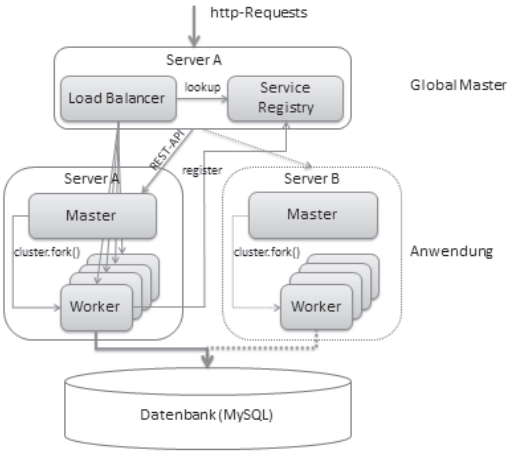


Abbildung 1: Anwendungsarchitektur Webshop.

4 Prototypische Umsetzung mit Node.js

Node.js ist ein Framework zur Entwicklung von skalierbaren Netzwerkanwendungen in Java Script. Konkret handelt es sich um ein eventgetriebenes non-blocking I/O-Framework auf Basis der "V8"-Engine von Googles [GOOG]. Die Besonderheit von Node.js liegt darin, dass es single Threaded ist. Anstatt zur Verarbeitung für jede eingehende Anfrage einen eigenen Thread zu verwenden, werden sämtliche Anfragen von einem einzigen Thread nacheinander verarbeitet, der eine endlos laufende Ereignisschleife, die „Event-Loop“ implementiert. Empfängt diese „Event-Loop“ nun Events von Programm- oder Nutzeraktionen werden diese in Callback-Funktionen umgesetzt. Wird von einem Programm eine externe Ressource, wie beim Auslesen einer Datei oder einer Datenbankabfrage benötigt, registriert dies die „Event-Loop“ und startet die Interaktion mit der externen Ressource. Danach wird die Interaktion beiseitegelegt, bis eine Antwort von der Ressource (z.B. Datenbank) in Form eines Callbacks vorliegt. In der Zwischenzeit kümmert sich die „Event-Loop“ um die in ihrer Queue zur Verarbeitung eingetragenen Events und verarbeitet diese bis zu deren ersten Interaktion mit einer externen Ressource. Ist die Interaktion abgeschlossen wird dies der „Event-Loop“ gemeldet und sobald die aktuellen den „Event-Loop“ blockierenden Funktionen abgeschlossen sind die beiseitegelegte Anfrage fortgesetzt entweder bis zur nächsten Interaktion mit einer externen Ressource oder bis zu deren Abschluss [NO12].Nachfolgender Abschnitt beschreibt, welche Node Module verwendet werden, um eine elastische Skalierung innerhalb der prototypischen Umsetzung zu ermöglichen. Der Global Master setzt sich aus zwei Node.js Modulen zusammen. Zum einen das Seaport Modul, welches eine Service Registry für die Verwaltung von Host/Port Tabellen beinhaltet. Dabei wird eine semantische Versionierung (name@server) verwendet. Für die Anwendung bedeutet dies, dass sich neue Workerinstanzen im Seaport Modul registrieren, die daraufhin vom Load Balancer abgefragt werden können[GITE]. Zum anderen das http-proxy Modul, welches eine Implementierung

des Load-Balancer zulässt. Durch diese Komponente kann ein Routing und eine Lastenverteilung der ankommenden http-requests an die entsprechende Workerinstanz der Anwendung durchgeführt werden [GITH]. Der Webshop wird mittels des Cluster Moduls und des Express Frameworks erstellt. Dabei ermöglicht das Cluster Module die Ausführung und Überwachung von mehreren Node.js Instanzen. Es findet eine Unterscheidung zwischen Master- und Workerinstanzen statt. Die Workerinstanz kann es mehrfach geben. Eine Masterinstanz wird durch den Aufruf der Anwendung gestartet. Ihre Aufgabe ist das Ausführen und Überwachen der Workerinstanzen, die dann die eigentliche Arbeit durch Business Logik und Websitengenerierung übernehmen [NODE]. Für die Kommunikation zwischen Global Master und Cluster Modul wurde eine REST-API implementiert. Diese ermöglicht es dem Global Master neue Instanzen zu starten und zu beenden, entsprechend der durch den Workload (z.B. Request pro Zeiteinheit) benötigten Ressourcen. Das Starten und Beenden der Workerinstanzen erfolgt über eine vordefinierte Route (z.B. workers/new). Stehen keine weiteren Ressourcen (z.B. in Form von CPU) zu Verfügung, wird über ein SSH-remote Command, auf einer Server-Instanz eine Version der Anwendung aus einem SVN Repository geladen und anschließend ausgeführt [GITS].

5 Zusammenfassung und Ausblick

Der Prototyp hat gezeigt, dass eine Anwendung spezifisch auf ein cloudbasiertes Umfeld ausgerichtet werden kann, um die Vorteile die Cloud Computing bietet auch nutzen zu können. Damit die Ressourcen der Cloud nach Bedarf genutzt werden können, muss die Anwendung Elastizität ermöglichen. Hierfür ist neben dem Loadbalancing auch eine Service Registry von Bedeutung. Wichtigster Bestandteil der Architektur ist aber ein Cluster von Instanzen der Anwendung. Innerhalb dieses Clusters können dann entsprechend der Auslastung der Anwendung Instanzen hinzugefügt oder entfernt werden. Die Implementierung des Webshops mittels Node.js hat gezeigt, dass mit Hilfe der genannten Elemente, Elastizität aus Anwendungssicht möglich ist. Das entwickelte Szenario bietet auch in Zukunft noch Ansatzpunkt für Folgeprojekte. In den Bereichen Sessionhandling, verteilte Transaktionen und Datenbankkonsistenz könnte das Projekt fortgeführt werden.

Literaturverzeichnis

[NIST] NIST Cloud Computing Reference Architecture, Recommendations of the National Institute of Standards and Technology, Liu, Tong, Mao, Bohn, Messina, Badger, Leaf, Gaithersburg, 2011.

[NO12] Node.js & Co., Skalierbare, hochperformante und echtzeitfähige Webanwendungen professionell mit JavaScript entwickeln, Roden, Heidelberg, Auflage 1, 2012.

[GITH] Github, https://github.com/nodejitsu/node-http-proxy, letzter Zugriff 25.01.2014.

[GITE] Github, https://github.com/substack/seaport, letzter Zugriff: 25.01.2014.

[GITS] Github, https://github.com/mscdex/ssh2, letzter Zugriff 25.01.2014.

[GOOG] Google, https://code.google.com/p/v8/, letzter Zugriff 25.01.2014.

[NODE] Node.js, http://nodejs.org/api/cluster.html, letzter Zugriff 25.01.2014.

Qualitätssicherung in agilen Teams – eine Mehrfachfallstudie

Holger Schmeisky

holger@schmeisky.com

Abstract: Gerade bei der Entwicklung von Webplattformen sind schnelle Releasezyklen essentiell. Traditionelle Qualitätssicherungsmethoden sind dafür zu langsam und agile Methoden versprechen Abhilfe. Ich habe untersucht, wie agile Teams Qualitätssicherung betreiben und habe herausgefunden, dass dort die Entwickler viel Verantwortung tragen und umfassende Aufgaben von Planung über Entwicklung, Qualitätssicherung, Veröffentlichung und Betrieb haben.

Gerade bei der Entwicklung von Webplattformen wird es immer wichtiger, schnell zu entwickeln. Kunden erwarten, dass Fehler innerhalb weniger Tage repariert werden und die Geschäftsseite will im Sinne von *Lean Startup* in möglichst kurzen Zyklen das Produkt anpassen ([Rie12]). Traditionelle Methoden der Qualitätssicherung (QS) mit nachgelagerten Testprozessen funktionieren hierbei nicht mehr, da sie die Zeit zwischen Entwicklung Veröffentlichung verlängern. Agile Methoden versprechen Abhilfe, es gibt aber Zweifel ob sie effektiv die Qualität sichern können.

In der Literatur gibt es nur wenige Untersuchungen zur Qualitätssicherung in agilen Teams. Es wird nur immer wieder betont, dass agile Methoden sich schlecht mit traditionellen Qualitätssicherungsmethoden vereinen lassen, es werden aber wenig Antworten darauf geliefert, wie diese ersetzt werden können. Insbesondere die Rolle des separaten Testers ist dabei interessant, da sie in traditionellen Methoden eine zentrale Rolle spielt, meist aber gar kein Teil von agilen Methoden ist.

Das Problem

Traditionelle Qualitätssicherungsmethoden sind weit verbreitet in der Softwareentwicklung. Agile Methoden widersprechen ihnen teilweise, daher gibt es Zweifel ob sie gut funktionieren. Untersuchungen zu agilen Teams deuten darauf hin ([KKTS10], [TKHD06]), dass agile Methoden gut funktionieren können, es aber schwierig ist sie umzusetzen.

Traditionelle Qualitätssicherungsmethoden werden definiert als solche, die allgemein akzeptiert und geschätzt sind im Software Engineering. Testen ist hierbei die wichtigste Aktivität, die Myers et al. definieren als “the process of executing a program with the intent of finding errors.” ([MSB11, p.11]). Dazu benötigt man eine *destruktive* Haltung, weshalb Tester *unabhängig* von den Entwicklern sein sollten ([MSB11, p.17]). Am Besten geht

das mit einer geschriebenen Spezifikation, wie in planbasierten Entwicklungsprozessen üblich. Dabei ist Testen meist eine *separate Phase* nach der Entwicklung.

Agile Prozesse beschreiben Qualitätssicherung meistens nur teilweise oder gar nicht. SCRUM z.B. hat zwar konstruktive QS-Praktiken, aber nur wenige analytische ([SS13]). Der kurze Sprintzyklus führt mit traditioneller, nachgelagerter QS allerdings zu Problemen. Extreme Programming hat viele QS-Praktiken durch Entwickler und die traditionellen, unabhängigen Tester fehlen ganz.

Es gibt nur wenige wissenschaftliche Untersuchungen dazu, wie QS in der Praxis gemacht wird. Talby et al. ([TKHD06]) führten eine Fallstudie bei einem XP-Team durch, das ein Enterprise Informationssystem für die Israelische Luftwaffe entwickelte. Die Qualität war sehr gut und Vorteile gegenüber vorherigen Prozessen waren, dass sie Fehler früher fanden und schneller beheben konnten.

Kettunen et al. ([KKTS10]) führten eine Fallstudie der Testprozesse und Aktivitäten in 12 Organisationen, die agil Software entwickeln durch. Sie fanden heraus, dass es zwar schwierig ist, parallel zu Testen und zu Entwickeln, aber die Flexibilität bei Änderungen erhöht wird und verhindert wird, dass man zu wenig Zeit und Ressourcen für das Testen plant. Außerdem ist das Feedback durch das Testen wertvoller, weil es früher kommt. Ein Problem, das sie entdeckten war, dass in Organisationen ohne dedizierte Tester (wo Entwickler testen), die Entwickler schnell das Testen überspringen, weil es als weniger wichtig angesehen wird als Entwickeln.

Die Studie

Ich habe meine Studie als Masterarbeit ([Sch14]) bei Prof. Lutz Prechelt von der AG Software Engineering an der Freien Universität durchgeführt. Da es nur wenige empirische Daten zur Qualitätssicherung in agilen Teams gibt, lautet die Forschungsfrage:

> Wie und unter welchen Bedingungen funktioniert agile Qualitätssicherung?
> Wenn sie gut funktioniert, warum funktioniert sie?

Es ist eine Mehrfachfallstudie von 2 Entwicklungsteams, eins bei SoundCloud, eins bei Immobilienscout24. Ich habe einen Entwickler der Teams jeweils 3 Tage bei der Arbeit begleitet und semi-strukturierte Interviews mit verschiedenen Teammitgliedern geführt. Fallstudien eignen sich besonders, um Fragen zu untersuchen, bei denen Kontext und Phänomen nicht klar getrennt werden können. Das ist hier der Fall: es ist nicht klar, welchen Einfluss Technologie, Produkt, Team oder Firma auf die Qualitätssicherung haben. Um aus einer Fallstudie objektive, belastbare Erkenntnisse zu erhalten sind vor allem verschiedenartige Datenquellen zum selben Betrachtungsgegenstand wichtig (*Triangulation*). Meine wichtigsten Triangulationen sind, dass ich viele Daten aus den Interviews in den Beobachtungen bestätigen konnte und in den Interviews verschiedene Rollen (Entwickler, Product Owner, SCRUM Master, ... soweit vorhanden) übereinstimmende Aussagen bekam.

Da die Firmen Webplattformen betreiben, entwickeln die Teams für Anforderer aus derselben Firma und veröffentliche ihre Software als Teil der Webplattformen der Firmen. Agile Prozesse sind in beiden Firmen "in Mode", allerdings gibt es keinen konkreten vorgeschriebenen Prozess, sondern jedes Team soll (u.U. mit Hilfe) einen Prozess finden, der zum Kontext und zum Team passt.

Ergebnisse

Die Teams bei SoundCloud und bei Immobilienscout24 sind sich sehr ähnlich: Beide haben einen, bzw. mehrere, unabhängige Dienste, die sie vollständig selber kontrollieren können. Diese Teams haben einen flußbasierten Prozess, wo Aufgaben nach Bedarf geplant werden und die Entwickler sind bei der Planung früh und stark involviert. Die Entwickler schreiben selber automatisierte Tests für die Software und es gibt eine klare Vorstellung davon, was einen guten Test ausmacht: Schnell, leicht verständlich, aussagekräftig (d.h. findet Fehler) und verlässlich. Die Tests sind in einer Pyramide strukturiert mit vielen kleinteiligen Tests, die viele Teile der Software nur simulieren (*mocken*), weniger Integrationstests die weniger mocken und sehr wenigen Webtests, die per Browser die gesamte Software testen. Nach der Abnahme durch den Product Owner werden die Änderungen innerhalb kurzer Zeit mithilfe von Deploymenttools per Knopfdruck/Befehl veröffentlicht. Die laufende Software wird durch die Entwickler selber überwacht, damit sie reagieren können, wenn sie nicht mehr funktioniert (z.B. viele Fehlerantworten ausgeliefert werden).

In beiden Firmen hat sich also ein sehr integrierter Prozess herausgebildet, in dem Entwickler viel Verantwortung tragen und viele Verantwortlichkeiten haben. D.h. sie entwickeln nicht nur, sondern haben auch Aufgaben, die traditionell eher anderen Rollen wie Qualitätssicherung oder Operations gehören. Der Prozess funktioniert gut, was man daran erkennt dass die Product Owner sehr zufrieden sind mit den Teams, weil sie Features zügig umsetzen können, es nur selten Bugs gibt und wenn doch, diese meistens innerhalb eines Tages behoben werden können.

Die Entwickler sind auch sehr zufrieden mit diesem Prozess und sind hoch motiviert. Sie können ihre Aufgaben mitgestalten und sind (im Normalfall) nicht auf andere Teams angewiesen, um etwas abzuschließen. Sie erhalten sehr schnell Feedback auf ihre Arbeit: vor allem durch die Tests, aber auch durch den Product Owner und schließlich von den Benutzern durch die zeitnahe Veröffentlichung.

Die hohe Zufriedenheit und Motivation lässt sich durch das "*Job Characteristics Model*" ([Wei04]) aus der Arbeitspsychologie erklären, das besagt, dass Arbeiter motiviert sind, wenn sie: (1) Vielfältige Aufgaben haben, (2) ihre Arbeit ein erkennbar zusammenhängendes Stück darstellt, (3) wissen, wie wichtig sie ist für den Firmenerfolg, (4) ihre Arbeit autonom gestalten können und (5) vor allem klare und direkte Rückmeldung über Erfolg und Wirksamkeit ihrer Arbeit bekommen. Diese Faktoren sind in den beiden Teams besonders ausgeprägt. Da sie ihre Aufgaben schnell abschließen können, müssen sie weniger gleichzeitige Aufgaben verwalten.

Schluss

Qualitätssicherung in agilen Teams kann gut funkionieren. Ich habe zwei Teams beobachtet, in denen Entwickler die Verantwortung für die Software von Planung über Entwicklung, Veröffentlichung und Betrieb haben. So können sie die schnellen Releasezyklen bei hoher Qualität gewährleisten. Die Entwickler müssen dabei durch Automatisierung (Tests, Deployment, Überwachung) unterstützt werden, damit die anderen Aufgaben nicht überhand nehmen. So zu arbeiten ist für die Entwickler sehr motivierend und psychologisch vorteilhaft, weil man nur wenige Aufgaben gleichzeitig hat.

Neu ist an diesen Erkenntnissen, dass diese Teams ganz ohne die traditionelle Rolle des Testers auskommen, der nach oder parallel zur Entwicklung die Qualität des Produkts bestimmt. In der Literatur wird immer von der Existenz eines Testers ausgegangen. Bei Talby et al. testet dieser parallel zu den Entwicklern. Bei Kettunen et al. wird davon ausgegangen, dass Tester immer noch zusätzlich zu den Entwicklern testen und es wird berichtet, dass es Probleme ohne diese Tester geben kann.

In diesen Teams übernehmen die Entwickler die Verantwortung für die Qualität und es ergeben sich Vorteile für die Arbeitsweise, dadurch dass die Entwickler die komplette Verantwortung von Planung bis Veröffentlichung übernehmen. Da die beobachteten Teams gut damit fahren, keine separaten Tester zu haben, stellt sich die Frage, in welchen Kontexten separate Tester überhaupt mehr Nutzen bringen als sie Probleme erzeugen.

Literatur

[KKTS10] V. Kettunen, J. Kasurinen, O. Taipale und K. Smolander. A study on agility and testing processes in software organizations. In *Proceedings of the 19th international symposium on Software testing and analysis*, Seiten 231–240. ACM, 2010.

[MSB11] Glenford J Myers, Corey Sandler und Tom Badgett. *The art of software testing*. Wiley, 2011.

[Rie12] Eric Ries. *Lean Startup: schnell, risikolos und erfolgreich Unternehmen gründen*. Redline Wirtschaft, 2012.

[Sch14] Holger Schmeisky. Qualitätssicherung in agilen Teams. Freie Universitt Berlin, Noch nicht abgegeben, 2014.

[SS13] Jeff Sutherland und Ken Schwabere. The Scrum Guide. www.scrum.org, 2013.

[TKHD06] D. Talby, A. Keren, O. Hazzan und Y. Dubinsky. Agile software testing in a large-scale project. *IEEE Software*, 23(4):30–37, 2006.

[Wei04] Ansfried B Weinert. Organisations- und Personalpsychologie. 2004.

FASL 1.0: Eine Skriptsprache zur Programmierung mobiler Geräte

Christian Cardello, Christian Dietrich, Philipp Eichhorn, Christian Eichler,
Marc Rosenbauer, Daniel Schmidt, Victor Simon
fasl@i2.cs.fau.de

Art der Arbeit: Projektarbeit („Mobile Application Development“)
Betreuer der Arbeit: Dipl. Inf. D. Brinkers, Dipl. Inf. T. Werth, A. Kumlehn, M.Sc.

Abstract: *FASL* ist eine Skriptsprache, die speziell für den Einsatz auf mobilen Geräten mit Android- oder iOS-Betriebssystem entwickelt wurde. Die Skripte können über eine umfangreiche API Gerätefunktionen nutzen, selbst erstellte graphische Oberflächen anzeigen, durch bestimmte Events ausgeführt und über einen Scriptstore geteilt werden.

1 Ziele

Die meisten herkömmlichen Programmiersprachen sind nicht an die Eigenheiten mobiler Geräte angepasst und bieten daher nur wenig Programmierkomfort. Eine mobile Skriptsprache muss mit den Gegebenheiten eines Smartphones, z.B. einem relativ kleinem Display und einer fehlenden Hardware-Tastatur, umgehen können.

```
import "Strings" as str
import "System" as sys
import "Http" as h

func main()
  res = h.get("http://ifconfig.me/ip")
  if(res.success)
    ip = str.trim(res.text)
    sys.showToast("your ip is: "+ip)
  else
    sys.showToast("cannot retrieve ip")
  end
end
```

Abbildung 1: Auslesen der IP via ifconfig.me

Die Sprache *FASL* ist auf eine geringe Anzahl an Schlüsselworten beschränkt und bietet somit eine optimale Grundlage für eine benutzerfreundliche mobile Oberfläche. Als Ausgleich dazu verfügt *FASL* über eine Vielzahl an Bibliotheken, um trotz der Limitierung komplexe Abläufe umsetzen zu können. Die Bibliotheken können bei Bedarf geladen werden, z.B. für den Internetzugriff, zur Darstellung von graphischen Elementen oder zur Bearbeitung von Zeichenketten (vgl. Abb. 1).

Das vorgestellte Projekt basiert auf den Arbeiten des Vorjahres [GRK13], welche im Laufe eines Semesters um eine Vielzahl an Eigenschaften erweitert wurden. Im Weiteren wird deshalb eine Auswahl der Neuerungen vorgestellt. Konkret werden das Laufzeitsystem, die Entwicklungsumgebung, die Möglichkeit zur Anbindung graphischer Benutzeroberflächen, der Scriptstore sowie die Portierung der Android-Applikation auf die iOS-Plattform beschrieben. Nach einem kurzen Überblick über verwandte Arbeiten endet der Beitrag mit einer Zusammenfassung.

2 Laufzeitumgebung

FASL verfügt über einen iterativen Interpreter mit Werte- und Befehls-Stack, wobei letzterer direkt aus dem abstrakten Syntaxbaum befüllt wird. Somit kann der aktuelle Zustand eines sich in Ausführung befindlichen Skriptes jederzeit gespeichert werden. Der Interpreter läuft unabhängig vom Rest der Applikation in einem Hintergrund-Dienst. Mehrere Skripte werden quasi parallel ausgeführt, wobei nach jeweils 1000 Instruktionen das ausgeführte Skript gewechselt wird (vgl. Zeitscheiben im präemptiven Multitasking).
Um mit der *FASL*-App möglichst mächtige Skripte schreiben zu können, benötigt diese fast alle von Android vorgesehen Berechtigungen. Aus Sicherheitsgründen kann ein neues Skript diese jedoch nicht direkt nutzen. Für jedes Skript werden die tatsächlich benötigten Berechtigungen gespeichert und bei Änderungen am Quellcode erneut eingefordert. Die in einem Skript verwendeten Bibliotheksfunktionen deklarieren per Annotationen die vorausgesetzten Berechtigungen, so dass der Parser diese schon zur Übersetzungszeit sammelt und vom Anwender vorab bestätigen lässt.
Die *FASL*-Laufzeitumgebung ist unabhängig von Android entworfen und kann daher auch lokal auf einem PC ausgeführt werden. Dabei können Aufrufe von Bibliotheksfunktionen, die zwingend ein Smartphone benötigen, z.B. SMS verschicken, ein Bild aufnehmen, etc., zur Ausführung per USB an ein Smartphone gesendet werden.
FASL stellt seine Laufzeitumgebung auch anderen Applikationen zur Verfügung. Diese können beliebige Skripte ausführen und ihre eigenen Bibliotheken mitbringen. Über Berechtigungen wird wiederum sichergestellt, dass der Anwender die Kontrolle behält.
FASL verfügt über sog. Schedules, mit denen Skripte automatisiert ausgeführt werden. Deren Ausführung kann durch diverse Funktionen ausgelöst werden. Dazu gehören neben den typischen Wecker-Funktionen unter anderem: Erreichen eines bestimmten Ortes, Eingehen eines Anrufs, einer SMS, Herstellen einer WLAN-Verbindung, Änderungen am Zustand des Akkus oder eine beliebige Kombination der obigen Auslöser mittels boolescher Algebra.

3 Entwicklungsumgebung

Die Entwicklungsumgebung ist ein wichtiger Teil des *FASL*-Projektes. Sie ermöglicht es dem Nutzer Skripte direkt auf einem mobilen Gerät zu schreiben. Hierzu gehört ein Editor mit Syntax-Highlighting, automatischer Einrückung, Undo, Redo und Hervorhebung von Fehlern im Programm. Die Skripte können wahlweise mit der normalen Tastatur des Touchscreens, einer Hardware-Tastatur oder einer speziellen token-basierten Tastatur, die häufig benötigte Token der Skriptsprache direkt einfügt, entwickelt werden.
Zusätzlich zur vereinfachten Eingabe von Skripten bietet *FASL* durch ein Vorschlagssytem eine weitere Hilfestellung zur Programmierung von Skripten. Während der Bearbeitung im Editor durchläuft ein Skript nach jeder Änderung die Stufen Lexer, Parser und Typinferenz. Aus der Typinferenz werden sehr viele Kontextinformationen abgeleitet, um passende Vorschläge generieren zu können. Diese Möglichkeit bleibt vielen anderen Editoren durch die klassische, strikte Trennung von der Übersetzungseinheit verwehrt.
Das System schlägt an geeigneten Stellen, neben Variablen und Funktionen des passenden Datentyps, Schlüsselwörter vor, z.B. `import` am Anfang eines Skriptes oder `func` und

`rec` außerhalb von Funktionen- oder Record-Definitionen. Innerhalb von Funktionen wird hingegen `while`, `foreach` und `if` angeboten.
Zusätzlich bietet *FASL* einen Debugger inkl. Anzeige der aktuellen Variablenwerte zur Fehlersuche an.

4 Views

Views dienen in *FASL* zur Erstellung einer graphischen Oberfläche für ein Skript durch den Benutzer. Als Ausgangspunkt werden bestehende graphische Elemente des Android-SDK (Unterklassen von *android.view.View*) verwendet. Views können dabei mit dem zugehörigen *FASL*-Skript kommunizieren (und umgekehrt). Dabei werden Skript-Funktionen eines Skripts direkt aus der View heraus aufgerufen (durch Rückruffunktionen, z.B. nach Tastendruck); skriptseitig kann die View durch Bibliotheksfunktionen verändert werden.
Zur Erstellung von Views liefert *FASL* einen entsprechenden Editor, in dem die Elemente der View hinzugefügt, positioniert und einzeln bearbeitet werden können. Darüber hinaus erfolgt die Verknüpfung zwischen Skript und View auch in diesem Editor.
Mögliche Elemente einer View sind Bilder, Texte, Checkboxen etc., aber auch Webviews (z.B. zur Programmierung eines Mini-Browsers in *FASL*). Des Weiteren können Views rekursiv andere Views einbinden, was den Aufbau komplexer Oberflächen ermöglicht.

5 Scriptstore

Der Scriptstore bietet dem Anwender die Möglichkeit, selbsterstellte Skripte hochzuladen, um sie so für andere Nutzer zugänglich zu machen. Der Scriptstore ist nicht nur direkt in der *FASL*-App integriert; es existert auch eine Web-Oberfläche, die von jedem regulärem Browser aus erreichbar ist.[1]
Neben einer Auflistung der am besten bewerteten Skripte, und Filterung nach Kategorien und Eigenschaften ist es auch möglich Skripte, nach Stichworten zu durchsuchen. Informationen, wie z.B. die unterstützten Plattformen und Versionen, angeforderte Berechtigungen und von Nutzern eingereichte Kommentare inklusive Bewertung, sorgen dafür, dass der Nutzer schnell einen Überblick über die vorhandenen Skripte und deren Funktionen erhält.
Für Skripte, die via Scriptstore heruntergeladen wurden, benachrichtigt *FASL* zudem den Nutzer über vorhandene Updates.

6 iOS

Die iOS-App entsteht durch Präprozessierung und Cross-Compilation größtenteils automatisiert aus der Android-Version. Im ersten Schritt wird durch die Präprozessierung der Java-Quelltext von den Abhängigkeiten zu Android befreit, weil z.B. die Benutzeroberfläche nicht automatisiert portiert werden kann. Der verbleibende Quelltext wird duch

[1] `http://mad.cs.fau.de/fasl/store/`

den J2ObjC-Compiler [J2OBJC] nach Objective-C übersetzt. Da die Laufzeitumgebung durch diesen automatischen Prozess portiert werden konnte, ist die Kompatibilität von *FASL*-Skripten zwischen den beiden Platformen Android und iOS sichergestellt.
Die durch Präprozessierung entfernten Passagen des Quelltextes wurden manuell an die iOS-Umgebung angepasst, z.B. die Benutzeroberfläche in Anlehnung an die Human Interface Guidelines [HIG] von Apple. Neben einem Editor mit Syntax- und Error-Highlighting, Vorschlägen und einer token-basierten Tastatur, bietet die iOS-Version ebenfalls Zugriff auf den Scriptstore, zeit- und ortsabhängige Schedules sowie einen Großteil der Bibliotheken.

7 Verwandte Arbeiten

Tasker ermöglicht die Kombination von Ereignissen zu sogenannten Tasks ohne detaillierte Programmierung, was jedoch einiges an Einarbeitungszeit erfordert. Außerdem bietet Tasker graphische Elemente, die die Inspiration für *FASL*-Views waren [TASK15].
Das Projekt SL4A (Scripting Layer for Android) portiert verschiedene Skriptsprachen auf die Androidplattform. Mit Hilfe vereinfachter APIs lassen sich sehr einfach mächtige Skripte schreiben. Es gibt Erweiterungen für viele Features, die *FASL* bereits von Haus aus bietet, z.B. Schedules, in Form von Apps (teilweise kostenpflichtig) [SL4A14].

8 Zusammenfassung

Dank der auf mobile Geräte zugeschnittenen Oberfläche und der zahlreichen, an die besonderen Eigenschaften und Anforderungen von Smartphones angepassten, Bibliotheken und Funktionen bietet *FASL* eine einfache Möglichkeit, Ideen spontan umzusetzen, ohne sich Gedanken über die Zielplattform machen zu müssen.
Hierbei wird der Entwickler durch die token-basierte Tastatur, Vorschläge, den Debugger und Schedules unterstützt.

Literatur

[GRK13] M. Grandeit, C. Romstöck, P. Kreutzer *FASL: Skriptsprache zur Programmierung mobiler Geräte*, Lecture Notes in Informatics (LNI) - Seminars, 2013

[HIG] Apple Inc. *iOS 7 Human Interface Guidelines*, 23.01.2014
`https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/`

[J2OBJC] *J2ObjC*, 23.01.2014
`http://code.google.com/p/j2objc/`

[SL4A14] Damon Kohler *SL4A*, 23.01.2014
`http://code.google.com/p/android-scripting`

[TASK14] Crafty Apps EU, *Tasker*, 23.01.2014
`https://play.google.com/store/apps/details?id=net.dinglisch.android.taskerm`

Automatische Erkennung von Model Smells in Simulink-Modellen

Jonas Paul Winkler, Quang Minh Tran

jwinkler@mailbox.tu-berlin.de, tu-berlin.tran@daimler.com

Abstract: Simulink-Modelle werden mit zunehmender Komplexität anfällig für Qualitätsdefizite. Die Ursache dafür sind strukturelle Probleme, sogenannte *Model Smells*. Die manuelle Erkennung von Model Smells ist aufwändig, daher wird ein Verfahren zur automatisierten Erkennung von Model Smells vorgestellt. Dieses ermöglicht es Modellierern, effizient die Qualität von Modellen zu verbessern.

1 Einführung

Matlab/Simulink von The MathWorks ist ein Werkzeug zur modellgetriebenen Softwareentwicklung. Simulink wird insbesondere in der Automobilbranche zur Entwicklung der Software für Microcontroller eingesetzt.

Bedingt durch die stetig steigende Komplexität der umzusetzenden Funktionen und sich während der Entwicklung ändernden Anforderungen sinkt in großen Simulink-Modellen über längere Zeit die interne Qualität [KKT11, S. 194]. Dies führt zu schlechter Wartbarkeit, Erweiterbarkeit und Testbarkeit des Modells.

Die Ursachen von Qualitätsdefiziten werden in der Regel manuell gesucht und behoben. Dieser Prozess ist fehleranfällig und sehr zeitaufwändig. Daher ist es wünschenswert, die Analyse der internen Modellqualität und die Beseitigung von Defiziten durch Werkzeuge zu automatisieren, wie es bereits bei imperativen Programmiersprachen der Fall ist. Dort gibt es umfangreiche Analysewerkzeuge[1] und in die Entwicklungsumgebungen integrierte Refactoring-Tools zum Verbessern von schlechtem Quellcode.

Ziel der Arbeit ist die Schaffung einer Grundlage zur Erkennung von strukturellen, qualitätsmindernden Problemen in Simulink-Modellen und die Entwicklung eines Prototyps zur Demonstration der Umsetzbarkeit der gezeigten Grundlagen.

2 Verwandte Arbeiten

Strukturelle und qualitätsmindernde Probleme existieren auch in imperativen Programmiersprachen. Dort sind sogenannte Code Smells ein Indiz für geringe interne Codequa-

[1] SonarQube, http://www.sonarqube.org/

lität [Fow99, S. 67ff]. Diese dienen in dieser Arbeit als Ausgangspunkt. Beispiele für bekannte Smells sind „Lange Methode“, „Große Klasse“ und „Duplizierter Code“. Code Smells können durch gezielten Einsatz von Refactoring behoben werden.

In Simulink gibt es bereits unterschiedlich Ansätze zur Qualitätsanalyse von Modellen. Jan Scheible beschreibt ein auf Metriken basierendes Qualitätsmodell für Simulink-Modelle [Sch12]. In diesem werden Kennzahlen (Anzahl der Blöcke, Durchschnittliche Anzahl an Subsystemen pro Subsystem) zu einer quantitativen Qualitätsbewertung von Simulink-Modellen zusammengesetzt. In dieser Arbeit überführen wir die Idee der Code Smells nach Simulink und führen den neuen Begriff „Model Smell“ ein. Zur Beschreibung von Model Smells nutzen wir unter anderem die Metriken aus [Sch12].

3 Model Smells

Ein Model Smell ist ein allgemeines Muster, dessen Vorhandensein in einem Simulink-Modell ein Indiz für geringe interne Modellqualität ist. Ein Simulink-Modell besteht aus Blöcken und gerichteten Linien. Blöcke repräsentieren die Operationen in einem Modell, Linien verbinden mehrere Blöcke miteinander. Über Linien werden Signale zwischen Blöcken transportiert.

Model Smells und Code Smells haben folgende Eigenschaften gemeinsam:

- Die Existenz eines Smells muss nicht zwingend niedrige Qualität bedeuten, daher muss der Entwickler für jeden in einem Artefakt (Code, Modell) gefundenen Smell selbst entscheiden, ob der Smell tatsächlich ein Problem darstellt.
- Wenn ein Problem erkannt wurde, muss dieses durch geeignete Gegenmaßnahmen („Refactoring“) eliminiert werden. Dabei verändert sich die Funktionalität des Artefakts nicht.

Das Muster eines Model Smells besteht aus auf abstrakten Simulink-Elementen formulierten *Bedingungen* und zwischen diesen Simulink-Elementen bestehenden *Zusammenhängen*. Mögliche Simulink-Elemente sind insbesondere Blöcke, Linien und Signale. Treffen alle Bedingungen und Zusammenhänge auf einen konkreten Teil eines Modells zu, so stellt dieser Teil im gegebenen Modell eine *Instanz* des Model Smells dar.

Nachfolgend werden beispielhaft zwei Model Smells vorgestellt.

Großes Subsystem. Dieser auf Basis des Code Smells „Lange Methode“ [Fow99, S. 69] entwickelte Model Smell erfasst Subsysteme, die mehr als eine festgelegte Anzahl an Kindblöcken besitzen. Sehr große Subsysteme implementieren meist zu viele Funktionen und sind daher ein Hinweis auf schlechte Modularisierung des Systems.

Signalumbenennung. Werden Signale entlang ihres Pfades umbenannt, wird der Signalname überschrieben. Für den Modellierer wird es dadurch schwieriger, Ursprung und Bedeutung von umbenannten Signalen zu bestimmen. Instanzen dieses Smells verschlechtern die Lesbarkeit und Verständlichkeit eines Modells.[2]

[2] http://www.mathworks.de/de/help/simulink/mdl_gd/signals-and-signal-labels.html

4 Formulierung in Prolog

Um Model Smells automatisiert finden zu können werden Simulink-Modelle und Model Smells in Prolog dargestellt. Modelle werden durch Fakten und Model Smells durch Regeln beschrieben. Prolog ermöglicht es daraufhin, im Modell automatisiert nach Model Smells zu suchen.

Folgende Konventionen gelten bei der Darstellung von Simulink-Modellen durch Fakten:

`block(b).`	Legt fest, dass `b` ein Simulink-Block ist.
`block_type(b,t).`	Der Typ des Blocks `b` ist `t`. `t` ist ein String.
`block_parent(b,p).`	`p` ist ein Subsystem und enthält den Block `b`.
`line(l).`	`l` ist eine Simulink-Linie.
`line_source(l,b).`	Die Linie `l` startet an Block `p`.
`line_name(l,n).`	Die Linie `l` trägt den Namen `n` (String).
`signal_source(s,b).`	Das Signal `s` wird an Block `p` erzeugt.

Fakten für ein konkretes Simulink-Modell werden mithilfe eines Skripts automatisch generiert. Auf Basis dieser Konventionen können nun die in Abschnitt 3 eingeführten Model Smells durch Regeln beschrieben werden.

```
large_subsystem(X) :-
    block(X),
    block_type(X, 'SubSystem'),
    aggregate_all(count, block_parent(_, X), C),
    C >= 15.
```

Wenn `X` ein Simulink-Block, insbesondere ein SubSystem-Block ist und die Anzahl der Blöcke, die `X` als Vater haben, größer oder gleich 15 ist, so ist das durch `X` repräsentierte Subsystem eine Instanz des Smells *Großes Subsystem*.

```
signal_rename(S, L) :-
    signal_source(S, Block),
    line_source(StartLine, Block),
    line_signal(L, S),
    line_name(L, _),
    \+ StartLine = L.
```

Die Quelle des Signals `S` ist der Block `Block`. Mit diesem Block verbunden ist die Linie `StartLine` - die erste Linie, auf der `S` geführt wird. Existiert eine Linie `L`, die ebenfalls das Signal `S` führt (ausgedrückt durch `line_signal`), einen beliebigen Namen trägt und sich von `StartLine` unterscheidet, so wird das Signal `S` durch die Linie `L` umbenannt. Die Kombination von `S` und `L` ist eine Instanz des Smells *Signalumbenennung*.

Gegeben sei ein durch Fakten repräsentiertes Simulink-Modell. Dann können mithilfe der als Regeln ausgedrückten Model Smells und einer Prolog-Engine zwei Operationen ausgeführt werden: Wird eine Model Smell-Regel mit konkreten Simulink-Objekten aufgerufen, beweist oder widerlegt die Prolog-Engine die Regel für die Eingaben. Wird eine Regel mit Variablen aufgerufen, sucht die Prolog-Engine nach allen möglichen Kombinationen, die zur Erfüllung der Regel führen.

5 Weitere Beiträge

Zwei mögliche Model Smells wurden in diesem Paper präsentiert. Neben den beiden vorgestellten wurden dreizehn weitere Muster als Model Smells identifiziert und in Prolog implementiert. Mit der Prolog-Implementierung ist nach Generierung von Fakten zu einem Simulink-Modell die Suche nach Model Smells in diesem Modell möglich. Analysen haben allerdings gezeigt, dass die Einsetzbarkeit der Prolog-Implementierung in realen Modellen mit moderater Größe (mehr als 1000 Blöcke) unpraktikabel ist und die Laufzeit der Erkennung einiger Model Smells quadratisch mit der Anzahl der Blöcke wächst.

Daher wurde auf Basis der Prolog-Regeln ein Werkzeug in M-Script entwickelt, welches aufgrund der Verwendung der Simulink-API deutlich schneller arbeitet. Dieses Werkzeug bietet darüber hinaus einige besonders für Anwender interessante Funktionen:

Das Werkzeug bietet eine graphische Benutzeroberfläche, durch die gefundene Instanzen in einer Liste dargestellt werden. Zur schnellen Lokalisierung kann der Modellierer mittels Schaltflächen die zu einer Instanz gehörenden Modellteile farbig hervorheben.

6 Ausblick

Der Einsatz des Werkzeuges in einem realen Modell mit geringer interner Qualität hat gezeigt, dass die gefundenen Smell-Instanzen ein guter Ausgangspunkt zur zielgerichteten Verbesserung der Modellqualität sind. Ausbesserungsmaßnahmen zur Entfernung von Model Smells können mithilfe von werkzeuggestützten Transformationen [TD13] automatisiert durchgeführt werden.

Zur Weiterführung des Themas müssen weitere Model Smells insbesondere in Kooperation mit Simulink-Modellierern identifiziert, in einem Katalog aufgelistet und implementiert werden.

Literatur

[Fow99] Martin Fowler. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional, 1. Auflage, 1999.

[KKT11] Sören Kemmann, Thomas Kuhn und Mario Trapp. Extensible and automated model-evaluations with INProVE. In *Proceedings of the 6th international conference on System analysis and modeling: about models*, SAM'10, Seiten 193–208, Berlin, Heidelberg, 2011. Springer-Verlag.

[Sch12] Jan Scheible. *Automatisierte Qualitätsbewertung am Beispiel von MATLAB Simulink-Modellen in der Automobil-Domäne*. Dissertation, Universität Tübingen, 2012.

[TD13] Quang Minh Tran und Christian Dziobek. Ansatz zur Erstellung und Wartung von Simulink-Modellen durch den Einsatz von Transformationen/Refactorings und Generierungsoperationen. In Holger Giese, Michaela Huhn, Jan Phillips und Bernhard Schätz, Hrsg., *MBEES*, Seiten 1–12. fortiss GmbH, München, 2013.

Evolution Sozialer Netzwerke - von Facebook zu P2P

Alexander Altmann

altmann@uni-potsdam.de

Abstract: Soziale Netzwerke wie Facebook werden von der Mehrheit der Internetteilnehmer genutzt. Die derzeit verbreiteten Systeme werden von großen kommerziellen Anbietern betrieben und bieten ihren Nutzern vielfältige Funktionen, weisen aber Mängel in den Bereichen Sicherheit, Datenschutz und Anbieterunabhängigkeit auf. In einer umfassend überwachten Welt eignen sich diese Sozialen Netzwerke daher nicht für persönliche Daten. Systeme nach dem Föderationsprinzip bringen den Nutzern Anbieterunabhängigkeit, wirksame Sicherheit und Datenschutz können aber nur Systeme auf P2P-Basis erreichen.

1 Einleitung

Diese Zusammenfassung basiert auf der im November 2013 fertiggestellten Diplomarbeit „Vergleich und Bewertung Sozialer Netzwerke im Hinblick auf Architektur, Sicherheit, Datenschutz und Anbieterunabhängigkeit“ von Alexander Altmann.

In der Arbeit werden dreizehn aktuelle Soziale Netzwerke miteinander verglichen (Facebook, Google+, LinkedIn, Moodle, Friendica, Lorea, Retroshare, Briar, Secushare, Twitter, StatusNet, Pump.io und Buddycloud). Der Vergleich basiert auf sechs selbstgewählten Fallbeispielen und einer begleitenden Nutzerumfrage. Beim Vergleich anhand grundlegender Anforderungen wird besonderer Wert auf Sicherheit, Datenschutz und Anbieterunabhängigkeit gelegt. Die Sozialen Netzwerke werden in Architekturen eingeteilt, welche Einfluss auf das Geschäftsmodell des Anbieters und die Möglichkeiten für Selbstbestimmung und Freiheit der Nutzer haben. Diese drei Architekturen und ihre Entwicklung werden im Folgenden vorgestellt.

2 Evolution der Architekturen

Der technische Aufbau eines Sozialen Netzwerks ist ein wichtiges Merkmal, denn er bestimmt, welche Möglichkeiten einem System innewohnen und welche ausgeschlossen sind. Es gibt grundsätzlich drei Architekturen: 1-Anbieter-Systeme, Föderation und P2P. Die beiden ersten sind Ausgestaltungen des Client-Server-Prinzips.

Die drei Architekturen haben sich auch zeitlich in dieser Reihenfolge entwickelt. Zuerst gab es 1-Anbieter-Systeme, welche noch von der Mehrheit aller Nutzer verwendet werden, dann folgten Föderations-Netzwerke und zuletzt entstanden P2P-Systeme, deren techno-

logische Entwicklung noch nicht abgeschlossen ist.

2.1 1-Anbieter-Systeme

Das 1-Anbieter-System ist die zentralistische Variante des Client-Server-Prinzips. Es gibt ein Serversystem von einem Anbieter für ein Soziales Netzwerk. Alle Nutzer greifen auf dieses eine System zu (siehe Abbildung 1).

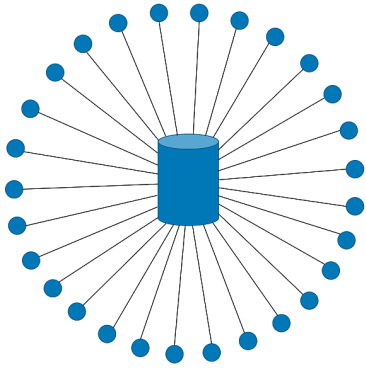


Abbildung 1: 1-Anbieter-System: Die Nutzer sind für Datenhaltung und Kommunikation auf den zentralen Server angewiesen.

Die derzeit nutzerstärksten Systeme Facebook, Twitter und Google+ liefern auf diese Weise alles aus einer Hand. Die prinzipiellen Vor- und Nachteile eines 1-Anbieter-Systems sind leicht ersichtlich. Möchte man dem Sozialen Netzwerk beitreten, ist man an diesen Anbieter gebunden und in allen Belangen auf ihn angewiesen. Der Anbieter ist oft eine kommerzielle Firma mit eigenen Interessen, welche nicht mit den Interessen des Nutzers übereinstimmen müssen.

Die 1-Anbieter-Systeme können als die erste Entwicklungsstufe von Sozialen Netzwerken betrachtet werden. Sie haben durch einfache Bedienung viele Nutzer überzeugt und ein zufriedenstellendes funktionales Niveau erreicht. Diese erste Stufe hat dem Nutzer Funktionalität gebracht, ihn aber abhängig werden lassen.

2.2 Föderation

Föderation basiert ebenfalls auf dem Client-Server-Prinzip, im Unterschied zu 1-Anbieter-Systemen verteilt sich die Nutzerlast aber auf mehrere „föderierte Server“. Die Sozialen Netze Friendica, Lorea, StatusNet, Pump.io und Buddycloud machen sich das Föderationsprinzip

zunutze, bei dem jede Anwenderin ihren „Heim“-Server hat und die Heim-Server der an der Kommunikation beteiligten Personen sich untereinander austauschen (siehe Abbildung 2).

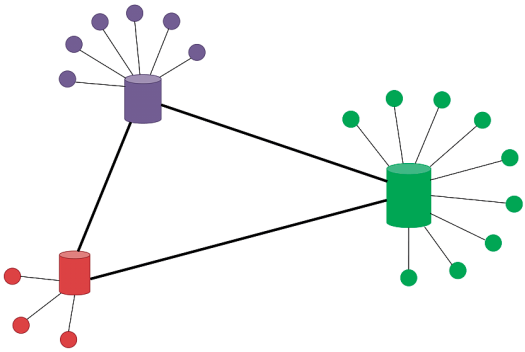


Abbildung 2: Föderation: Die Nutzerin wird unabhängiger, da sie sich ihren Server innerhalb des Systems aussuchen kann.

Föderationssysteme stellen die zweite Entwicklungsstufe Sozialer Netzwerke dar. Sie erlauben der Nutzerin die Wahl eines Betreibers innerhalb des Sozialen Netzwerks und befreien sie so aus der Abhängigkeit eines einzigen Anbieters. In der Steigerung dieser zweiten Stufe erfolgt die Vernetzung auch systemübergreifend und die Nutzerin kann Kontakte in fremden Netzwerken erreichen. Die zweite Entwicklungsstufe hat der Nutzerin Anbieterunabhängigkeit und Entscheidungsfreiheit gebracht, aber keinen Schutz ihrer Privatsphäre und persönlichen Daten.

2.3 P2P-Systeme

Das „Peer-To-Peer“-Prinzip ist ein Gegenentwurf zu „Client-Server“. Bei P2P gibt es keine Server mehr, keine Hierarchie, sondern nur noch Gleiche (Peers), die sich miteinander austauschen – diese werden Knoten (Nodes) genannt. Die Knoten übernehmen sowohl Client- als auch Serverfunktionen, das bedeutet, sie stellen Ressourcen bereit und nutzen Ressourcen anderer Knoten. Retroshare, Briar und Secushare sind Soziale Netzwerke auf P2P-Basis, von denen Stand Januar 2014 allerdings nur Retroshare für den praktischen Einsatz bereit ist. Briar befindet sich im Alpha-Stadium und Secushare ist noch nicht nutzbar.

In Abbildung 3 kann man sich die Knoten als Personen vorstellen und die Kanten als Beziehungen zwischen den Personen und erhält so das Bild eines natürlichen sozialen Netzwerks, wie es Soziologen kennen. Das P2P-Prinzip ist unter den hier beschriebenen Architekturen die natürlichste Abbildung eines sozialen Netzwerks auf eine technische

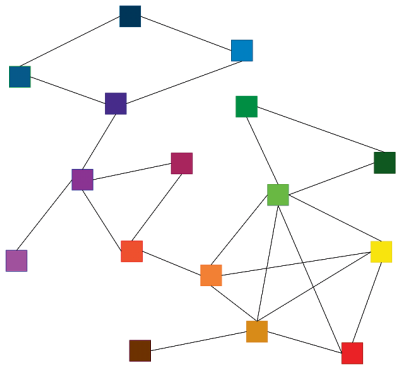


Abbildung 3: P2P: Ohne zentrale Server sorgen die Knoten (und damit die Nutzer) selbst für Datenhaltung und Konnektivität.

Struktur.

Ein P2P-System ist aber nicht nur die natürlichste Abbildung eines sozialen Netzwerks zwischen Personen, sondern auch die Struktur, welche Sicherheit und Datenschutz am besten gewährleisten kann. In einem Client-Server-System liegt die Kontrolle über die Daten der Nutzer in der Hand des Anbieters, liegen die Daten auf einem zentralen Server. Der Server stellt damit ein attraktives Ziel für technische Angriffe, rechtliche Kontrollmaßnahmen und Zugriffe von Geheimdiensten dar. Bei einem P2P-Netzwerk, in dem jeder Nutzer seinen eigenen Knoten betreibt, ist der Aufwand für derlei Angriffe weitaus höher und lohnt sich nur für wirklich wichtige Ziele.

Föderation kann man sich als Zwischenschritt vorstellen, bei dem sich die Daten zwar auf mehrere Server verteilen, aber noch keine vollständige Dezentralisierung erreicht wird.

3 Fazit

In einer Welt der umfassenden Überwachung und Kontrolle kann es keine Freiheit ohne den Schutz persönlicher Daten und der Privatsphäre geben. In dieser Welt möchte der Nutzer nur noch seinen direkten Kommunikationspartnern vertrauen müssen, keinem womöglich kommerziellen oder staatlichen Betreiber eines Dienstes. Datenschutz und Sicherheit werden durch Dezentralisierung und die Kontrolle über die eigenen Daten auf dem eigenen Endgerät erreicht, Austausch und Kommunikation erfolgen über verschlüsselte Verbindungen in einem P2P-Netzwerk.

P2P-Systeme sind die dritte Entwicklungsstufe von Sozialen Netzwerken. Diese dritte Stufe ermöglicht dem Nutzer tatsächliche Freiheit.

4 Quellen (Auszug)

- David Barkai: *Peer-to-Peer Computing – Technologies for Sharing and Collaboration on the Net*. Intel Press, 2002
- Foucault, Michel: *Überwachen und Strafen. Die Geburt des Gefängnisses*. Paris, 1975
- Mahlmann, Peter; Schindelhauer, Christian: *Peer-to-Peer-Netzwerke*. Berlin, Heidelberg, Springer-Verlag, 2007
- Toth, Gabor X.: *Design of a Social Messaging System Using Stateful Multicast*. Amsterdam, University of Amsterdam, Master's, 2013. - 76 S.
- Wachs, Matthias; Schanzenbach, Martin; Grothoff, Christian: *On the Feasibility of a Censorship Resistant Decentralized Name System*. In: 6th International Symposium on Foundations and Practice of Security (FPS 2013). La Rochelle, France, Springer Verlag, 10/2013

Die vollständige Liste der Quellen ist ebenso wie die Arbeit selbst unter `http://www.edition1.net/rs7/da/` abrufbar.

Design- und Testmethoden für interaktives Kinderspielzeug

Betina Bertleff

Hochschule Reutlingen
Fakultät für Informatik
betina.bertleff@gmail.com

Art der Arbeit: Semesterprojekt im Fach „Interaktive Systeme“
Betreuer der Arbeit: Dr. Kai Holzweißig

Abstract: Für das Testen von Kinderspielzeug mit Kindern sind spezielle Design- und Testmethoden notwendig. Die vorliegende Arbeit diskutiert grundsätzliche Aspekte dieser Methodiken und zeigt anhand eines beispielhaften Praxistests, wie diese gewinnbringend im Produktentstehungsprozess eingesetzt werden können. Durch die gezielte Analyse können Verbesserungsvorschläge erarbeitet werden, die sich positiv auf die User Experience des entsprechenden Produktes auswirken.

1 Einleitung

„The Nature of Child Computer Interaction is […] a study of the Activities, Behaviours, Concerns and Abilities […], as they interact with computer technologies.” [RB11, S.7]

Kind-Computer Interaktion ist ein Forschungsfeld in der Mensch-Maschine-Interaktion [vgl. RB11], wobei die Zielgruppe „Kind“ von Neugeborenen über Kleinkinder bis hin zum Jugendalter reicht [vgl. Du13]. Für Kinder existieren spezielle Ansätze für Design- und Testmethoden, da erkannt wurde, dass Erwachsene nicht für sie testen können. Kinder haben eine andere Wahrnehmung, fühlen und denken anders und stellen mittlerweile eine immer häufiger fokussierte Benutzergruppe dar [vgl. CG05; JS05]. Heutzutage nutzen immer mehr Kinder und in einem immer früheren Alter computerbasierte Spielzeuge oder Softwareprodukte [vgl. BGM]. Bei der Entstehung von interaktiven Produkten sind eine Reihe von Herausforderungen zu meistern, zu denen auch grundsätzliche rechtliche und ethische Erwägungen zählen [vgl. SP05, S.37; Ho08, S.306ff]. Wie Kinder in die Design- und Testmethodik miteinbezogen werden können, wird im Folgenden theoretisch beschrieben und praktisch in einem Test mit einer Spielzeug-Fernbedienung untersucht.

2 Gedanken zur Design- und Testmethodik

"If you make something for children, the first question you must ask yourself is, 'What does the world look like to children?'" [HRA97, S.794]

Um gute Produkte zu schaffen, sollten Informationen über die Meinung, Einstellung und das Verhalten von Kindern direkt von ihnen gesammelt werden [vgl. RF05]. Die Interaktion mit einem elektronischen Produkt kann nach Chiasson et al. eine Bereicherung für das Kind darstellen, da neue Technik selbst kleinsten Kindern wissenschaftliche Sachverhalte näher bringt [vgl. CG05]. Für Shneiderman und Plaisant sollten geeignete Designprinzipien für Kinderprodukte den Wunsch nach Interaktion, Kontrolle und entsprechendes Feedback aufnehmen und ihr soziales Engagement unterstützen [vgl. SP05, S.36]. Sears und Jacko beschreiben es simpel: „Mach es so einfach wie möglich, und vielleicht sogar noch einfacher" [vgl. SJ08, S. 796]. Kinder können als Anwender, Tester, Informanten oder als Designpartner in den Produktentstehungsprozess miteinbezogen werden [vgl. GD05; IB07]. Ein Schlüsselfaktor dabei ist das Thema „Vertrauen", welches zunächst zu den beteiligten Personen, dem Testobjekt und der Testumgebung aufgebaut werden muss. Daten können durch verschiedene Methoden erhoben werden: Vom lauten Denken, über die Eins-zu-Eins-Arbeit [vgl. BGM], einem Interview [one11], Gruppenbeobachtungen [vgl. RSP11, S.247], Fragebögen und Feldstudien ist alles möglich. Das nonverbale Verhalten der Kinder sollte berücksichtigt werden und die Kommunikation auf Augenhöhe stattfinden.

3 Beispielhafte Analyse: Die FisherPrice-Kinder-Fernbedienung

„If children cannot or do not care to use technologies we have designed, it is our failure as designers." [SJ08, S.794]

Im Rahmen eines Praxistests wurde als Erprobungsobjekt eine Kinder-Fernbedienung des Herstellers Fisher-Price gewählt[1]. Das „Kinderhaus Hochdorf" und die „Remsracker" erklärten sich dazu bereit, bei der praktischen Analyse zu helfen. Die User Experience eines Produktes kann in seiner natürlichen Umgebung am effektivsten beurteilt werden [vgl. RSP11, S.436], darum wurde ein Feldtest im Spielraum der Gruppe durchgeführt. In dieser „wilden Umgebung" ist es allerdings nicht einfach, Informationen zu sammeln, da Kinder und die Umgebung nicht beeinflusst werden können [vgl. Ho08, S.86]. Iversen et al. beschreiben Aktivitäten von Kindern als offen und erforschend. Sie wissen was sie mögen, sind neugierig und haben ihre eigenen Normen und Komplexität [vgl. IB07]. Dies sollte im Test validiert werden und zusätzlich sollte die Fragestellung geklärt werden, ob die Kinder mit dem Spielzeug umgehen können und wollen und wie sie es nutzen. Es sollte geklärt werden, ob die Fernbedienung für Kinder, wie von FisherPrice angegeben, ab sechs Monaten geeignet sei. Daher wurde der Test mit Kindern zwischen sechs und 36 Monaten durchgeführt.

[1] Vgl. FisherPrice-Kinder-Fernbedienung http://goo.gl/vzpL7s (04.Januar 2014)

Analog zu den oben beschrieben grundsätzlichen Erwägungen wurde die folgende Planungen für die Erprobung getroffen: Nach dem Ankommen und Einfügen in die Gruppe werden die Kinder kennengelernt, um Vertrauen aufzubauen. Anschließend wird die Fernbedienung zur Verfügung gestellt und die Kinder werden beobachtet, wie sie mit dem Produkt umgehen und sich verhalten. Zum Schluss findet ein Austausch mit den Erzieherinnen statt, um weitere Informationen zu erhalten und sich auszutauschen.

Die ersten drei bis fünf Kinder eines Tests reichen nach Barendrengt und Bekker aus, um 80% der Usability-Probleme zu finden. Allerdings erhält man mit mehr als fünf Teilnehmern einen klareren Eindruck der Schwere der Fehler [vgl. BB]. Im Bezug auf die Ergebnisse des durchgeführten Praxistests ist zu sagen, dass sich im Test viele Annahmen, die zuvor aus Literatur oder Gesprächen mit den Erzieherinnen angedeutet wurden, beispielsweise dass Kinder sehr ungeduldig seien und eine kurze Aufmerksamkeitsspanne besäßen, bestätigten. Bunte Flächen und blinkende Lichter auf der Fernbedienung wurden vermehrt berührt und versucht zu drücken, was bei der momentanen Spielzeugfernbedienung zu keinem Ergebnis und einem enttäuschten Kind führt. Die Nummerntasten der Fernbedienung werden gerade von kleinen Kindern schlecht getroffen, gleichzeitig ist die Fernbedienung aber auch zu sperrig, um sie in den Händen halten zu können. Für das angegebene Alter ab sechs Monaten ist das Spielzeug somit nicht geeignet. Musik und Töne kommen bei den Kindern sehr viel besser an als normal gesprochener Text. Feedback wird als sehr wichtig erachtet. Dieses sollte mit einer sehr kurzen Verzögerung ausgegeben werden, da sonst andere Knöpfe betätigt werden oder das Spielzeug komplett aus der Hand gelegt wird. Beim getesteten Produkt ist die Feedbackzeit zu lang gewählt.

4 Schlussfolgerungen und Ausblick

"Perhaps success in a survey in Child Computer Interaction is [...] measured by the answers to [the question ...] 'Did I learn [.. or] do anything useful?'" [RF05, S.3]

Der vorliegende Kurzartikel hat einige grundsätzliche Problemstellungen hinsichtlich Design- und Testmethodiken für Computertechnologien für Kinder diskutiert. Diese wurden anhand eines Praxisbeispiels verdeutlicht.

Das Testen mit Kindern braucht eine große Vorlaufzeit, da viel vorbereitet und abgesprochen werden muss. Gleichzeitig kann man nie wissen, was einen am Testtag genau erwartet. Die Fernbedienung kam bei den Kindern gut an, da sie neu und aufregend ist und Geräusche macht. Die Kinder waren sehr interessiert und neugierig und spielten konzentriert mit der Fernbedienung. Für das angegebene Alter ist sie eher ungeeignet und Feedbackzeit und Farbe sollten angepasst werden. Da Lieder imitiert und nachgesungen werden, könnte zusätzlich ein Lernaspekt mitaufgenommen werden, beispielsweise um Kindern spielend Zahlen oder das Alphabet beizubringen.

Wichtig bei einer Einbeziehung von Kindern sind darüber hinaus auch grundsätzliche rechtliche und ethische Erwägungen. Angefangen bei der Zustimmung der Eltern sowie bei Fragen des Datenschutzes, muss darauf geachtet werden, dass die verwendeten Design- und Testmethoden ethisch vertretbar sind. Kinder dürfen in solchen Tests nicht ausgenutzt werden, sondern müssen als Experten und vollwertige Mitglieder in den Prozess miteinbezogen werden. Durch die Einbringung ihrer eigenen Meinung und ihrer Ideen können unter der Beteiligung von Kindern interaktive Produkte effektiv entwickelt und angepasst werden.

Literaturverzeichnis

[BB] W. Barendrengt, M.M. Bekker: Guidelines for user testing with children (Department of Industrial Design, Eindhoven University of Technology) w3.id.tue.nl/fileadmin/id/objects/doc/Guidelines_for_user_testing_with_children.pdf

[BGM] M. Burmester, C. Görner, J. Maly: Usability für Kids-So testet und gestaltet man interaktive Medien für Kinder (Studie der UID GmbH in Kooperation mit der HdM www.ibusiness.de/wrapper.cgi/www.ibusiness.de/files/jb_967842977_1235144147.pdf

[CG05] S. Chiasson, C. Gutwin: Design Principles for Children's Technology (Department of Computer Science, University of Saskatchewan, HCI-TR-2005-02)

[Du13] http://www.duden.de/rechtschreibung/Kind Zugriff am 19.Juni 2013

[GD05] M.L. Guha, A. Druin, G. Chipman, J.A. Fails, S. Simms, A. Farber: Working with Young children as Technology Design Partners (CACM, Jan'05 Vol.48, No.1)

[Ho08] J.Hourcade: Interaction Design&Children (Foundations and Trends in HCI Vol.1, No.4)

[HRA97] Hanna, Risden & Alexander: Guidelines for Usability Testing with Children (interactions - methods & tools, 1997)

[IB07] O. Iversen, C. Brodersen: Building a BRIDGE between children and users: a socio-cultural approach to child-computer interaction (Cogn Tech Work, Springer-Verlag London Limited 2007)

[JS05] J. Jensen, M. Skov: A Review of Research Methods in Children's Technology Design (IDC ´05 Proceedings of the 2005 conference on Interaction design and children)

[one11] Top Tips for Usability Testing with Kids (Januar 2011: onetooneglobal.com/blog/2011/01/12/top-tips-for-usability-testing-with-kids/, 11.06.13)

[RB11] J. Read, M. Bekker: The Nature of Child Computer Interaction (BCS-HCI ´11 Proceedings of the 25th BCS Conference on HCI)

[RF05] J. Read, K. Fine: Using Survey Methods for Design and Evaluation in Child Computer Interaction (workshop on CCI: Methodological Research at Interact 2005, Rome, Italy)

[RSP11] Rogers, Shart, Preece: Interaction Design – beyond human-computer interaction (John Wiley & Sond Ltd 2011, Third Edition)

[SJ08] A. Sears, J. Jacko: The Human-Computer Interaction Handbook (Taylor & Francis Group, LLC)

[SP05] B. Shneiderman, C. Plaisant: Designing the User Interface – Strategies for effective Human-Computer Interaction (Pearson Education 2005, Fourth Edition)

Usability-Evaluation-Framework für mobile Anwendungen

Tobias Braumann, Tobias.Braumann@student.hs-rm.de
Andreas Otto, Andreas.Otto@student.hs-rm.de

Hochschule Rhein-Main
FB DCSM - Informatik
Betreuer: Bodo A. Igler

Abstract:

Usability stellt im mobilen Bereich den wichtigsten Faktor für die Zufriedenheit der Nutzer dar. Somit sind verlässliche Usability-Tests essenziell für den Erfolg mobiler Anwendungen. Zudem hat sie direkten Einfluss auf die Wirtschaftlichkeit einer Anwendung. Gute Usability führt zu geringeren Entwicklungskosten, verringerten Wartungskosten und zu einer höheren Kundenbindung.

Für Desktop-Anwendungen und Webseiten gibt es eine Vielzahl an Ansätzen und Usability-Evaluations-Frameworks. Jedoch gibt es nur wenige Forschungsarbeiten und geeignete Werkzeuge zur Evaluierung von Usability mobiler Anwendungen. Zudem zeigen aktuelle Forschungsergebnisse, dass sich die Usability für mobile Anwendungen nur eingeschränkt mit gängigen Verfahren aus dem Desktopbereich bewerten lässt. Das liegt zum größten Teil an der Art des Kontextes. Desktop-Anwendungen werden meist in einer isolierten Umgebung mit konstanten Umgebungseigenschaften benutzt. Der Nutzungskontext mobiler Anwendungen dagegen ist durch natürliche Störungen wie Lärm, Multitasking, Bewegung und andere Einflüsse geprägt.

Dieses Paper basiert auf den Ergebnissen aus einem Masterprojekt (T. Braumann) und einer Master-Thesis (A. Otto). Es präsentiert ein Konzept und dessen Umsetzung zur Evaluierung der Usability mobiler Anwendungen.

1 Einleitung

Als Grundlage für das hier betrachtete Usability-Evaluation-Framework diente das abgeschlossene Forschungsprojekt SMAT (Success Factors of Mobile Application Design for Public Transportation) [Bö13, BIB13]. Ziel von SMAT war es, die Haupterfolgsfaktoren von mobilen Applikationen für den öffentlichen Nahverkehr zu untersuchen. Daraus resultierte die Entwicklung eines Prototyping-Frameworks für mobile Applikationen, um diese Einflussfaktoren und Ideen schnell testen zu können. Auch wurde ein Werkzeug benötigt, um die Güte der Prototypen im Hinblick auf Usability bewerten zu können.

Kundenzufriedenheit ist entscheidend für den Erfolg oder Misserfolg einer mobilen Anwendung auf dem hart umkämpften App-Markt [Res]. Hinzu kommt, dass Usability nicht nur zu einer höheren Kundenbindung führt, sondern auch Entwicklungs- und Wartungskosten senken kann [BM05].

Mobile Anwendungen erobern immer mehr Bereiche unseres Lebens und sind von ih-

rer Natur stark kontext-sensitiv. Somit sind Feldtests besser geeignet, wie bereits Tamminen [TOTK04] in seiner Untersuchung bestätigt, da Labortest nur schlecht Einflüsse, wie Lärm, Multitasking, Bewegung oder Unterbrechungen simulieren können. Hinzu kommen gesteigerte Anforderungen an die Usability, aufgrund geringerer Größen der Benutzeroberfläche und eine Fokussierung der Anwender schnell auf die wesentlichen Informationen zugreifen zu können. Jedoch fanden laut Coursaris und Harrison die meisten Usability-Tests von mobilen Anwendungen noch immer im Labor statt und der Kontext wurde nur in den wenigsten Fällen berücksichtigt [HFD13, CK11]. Laut dieser Untersuchungen gibt es bis heute noch keine gängigen Untersuchungsmethoden speziell für die Bewertung von Usability von mobilen Anwendungen. Die Entwicklung eines solchen Frameworks ist ein wichtiges Thema zukünftiger Forschungsarbeiten. Ein weiterer Grund für den geringen Einsatz sind zudem die höheren Kosten von Feldtests, verursacht durch spezielle Ausrüstung oder zusätzliche Mitarbeiter [dSCDR08] [KS04].

Das Hauptziel der hier vorgestellten Arbeiten ist die Entwicklung eines Frameworks zur Evaluierung von Usability mobiler Anwendungen. Die Anforderungen an ein solches Framework sind "...eine minimalinvasive Messung von Usability für mobile Applikationen, die vom Nutzer kaum bzw. gar nicht wahrgenommen wird, geringere Kosten als Feld- und Labortests verursacht, den Arbeitsaufwand für Usabilitytests verringert, sich früh in den Entwicklungsprozess integrieren lässt, ein möglich großes Spektrum an Usability-Problemen (automatisiert) aufdeckt und in der Umgebung stattfindet, in der mobile Geräte auch genutzt werden."[Bra13].

2 Usability-Evaluation-Framework

Die Master-Thesis [Ott01] beschäftigt sich mit der Architektur und der Entwicklung eines Usability-Evaluation-Frameworks (AMUSES - Architektur für mobile Usability-Evaluierungs-Systeme).

AMUSES beschränkt sich dabei nicht nur auf das Sammeln von Messdaten, sondern skizziert den kompletten Prozess eines Usability-Tests. Das FMC-Block-Diagramms [1] in Abbildung 1 zeigt das komplette Konzept. AMUESES steht dabei Mittelpunkt und verbindet alle beteiligten Personen und die zu testende mobile Anwendung. Es unterstützt sie bei ihren Aufgaben. Während der Durchführung eines Usability-Test werden jedoch nur die Probanden benötigt.

Zudem lässt sich anhand der Abbildung der komplette Ablauf eines Usability-Tests skizzieren. Um eine beliebige **Mobile Anwendung** mit Probanden zu testen, integriert der **Anwendungsentwickler** AMUSES in sein Programm. Das unterstützt ihn beim Tracken von Daten und stellt den **Probanden** zusätzlich ein **Feedback-Tool** zur Verfügung, die den **Analysten** zusätzlich unterstützt, um genauere Rückschlüsse auf Usability-Probleme zu ziehen. Der Test selber kann mit einer **Konfigurationsdatei** eingestellt werden, welche vorher vom **Moderator** festgelegt wird. Die Feedback- und Trackerdaten werden auf

[1] Fundamental Modeling Concepts (FMC) ist eine semi-formale Methodik zur vereinfachten Darstellung und Kommunikation über komplexe Softwaresysteme, http://www.fmc-modeling.org/

dem Server gespeichert, von wo aus sie später oder live durch den **Analysten** mit Hilfe eines **Analyse-Tools** ausgewertet werden können. [Ott01] bietet eine weiterführende ausführliche Darstellung von AMUSES und der Entwicklung.

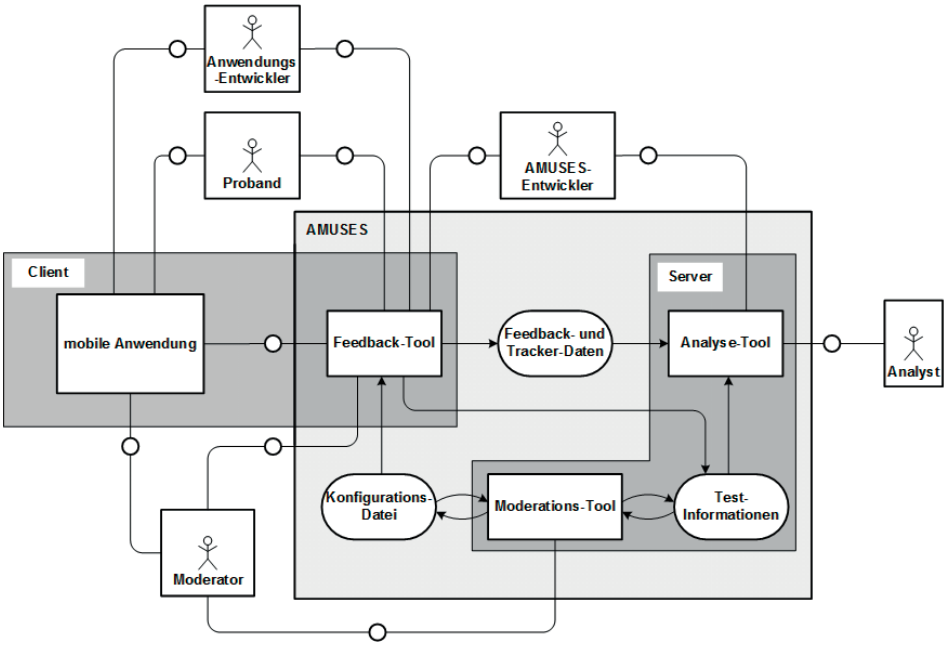


Abbildung 1: Aktueller Architekturentwurf von AMUSES anhand eines FMC-Block-Diagramms [Ott01]

3 Aktueller Stand

Zum jetzigen Zeitpunkt ist AMUSES bereits in vielen Teilen umgesetzt. Es existiert eine Library für Android, die sich in beliebige mobile Anwendungen mit nur einer Zeile Code integrieren lässt. Des weiteren können Daten wie GPS, Screenshots, Benutzungspfad, Kontextwechsel und Touch-Events getrackt werden. Zudem ist ein Feedback-Tool für die Probanden implementiert, welches im SMAT-Forschungsprojekt prototypisch entwickelt wurde und sich bei ersten Tests mit Probanden bewährt hat [Bö13]. Ein Web-Analyse-Tool, mit dem sich die gesammelten Daten auswerten lassen, erarbeiten gerade Bachelorstudenten im Rahmen eines studentischen Projekts.

4 Ausblick

Die nächsten Schritte bestehen darin, die bislang nur prototypische Teilimplementierung von AMUSES weiterzuentwickeln und erste Usability-Evaluierungen durchzuführen, um einen direkten Vergleich zu anderen Feld- und Labortests zu ziehen. Es wird sich zeigen, ob die Erwartungen an das Framework erfüllt werden können und AMUSES eine Alternative zu bestehenden Ansätzen von Usability-Tests für mobile Anwendungen darstellt.

Literatur

[Bö13] S. Böhm. A Tool-based Approach for Structuring Feedback for User Interface Evaluations of Mobile Applications. *ICBM*, 5(1):1–4, 2013.

[BIB13] T. Braumann B. Igler und S. Böhm. EVALUATING THE USABILITY OF MOBILE APPLICATIONS. *ICBM*, 5(1), 2013.

[BM05] R. G. Bias und D. J. Mayhew. *Cost-justifying usability: An update for the Internet age*. Morgan Kaufmann, 2005.

[Bra13] T. Braumann. Ein Ansatz für ein umfassendes, kostengünstiges und minimalinvasives Remote Usability Testing Framework für mobile Applikationen, Masterprojekt, HS-RM, 2013.

[CK11] C. K. Coursaris und D. J. Kim. A Meta-Analytical Review of Empirical Mobile Usability Studies. *J. Usability Studies*, 6(3):11:117–11:171, Mai 2011.

[dSCDR08] M. de Sá, L. Carriço, L. Duarte und T. Reis. A Mixed-fidelity Prototyping Tool for Mobile Devices. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, Seiten 225–232, New York, NY, USA, 2008. ACM.

[HFD13] R. Harrison, D. Flood und D. Duce. Usability of mobile applications: literature review and rationale for a new usability model. *Journal of Interaction Sc.*, 1(1):1–16, 2013.

[KS04] Kjeldskov und Stage. New techniques for usability evaluation of mobile systems. *International Journal of Human-Computer Studies*, 60(5–6):599 – 620, 2004. HCI IMC.

[Ott01] A. R. Otto. Eine Architektur für mobile Usability-Evaluierungs-Systeme, Master-Thesis, HS-RM, Abgabe: 2014-01.

[Res] Goldmedia Custom Research. Wichtigste Ansprüche der Nutzer an ihre Smartphones: Gute Bedienbarkeit und viele Apps 2011, http://goo.gl/K8sg1, [Accessed 10.01.2013].

[TOTK04] S. Tamminen, A. Oulasvirta, K. Toiskallio und A. Kankainen. Understanding mobile contexts. *Personal and ubiquitous computing*, 8(2):135–143, 2004.

Trisda the Robot – ein Beitrag zur Visualisierung der Objektorientierung

Karsten Klaus

Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik
karsten.klaus@uni-jena.de

Art der Arbeit: Abschlussarbeit zur 1. Staatsprüfung
Betreuer der Arbeit: Michael Fothe, Wolfram Amme

Abstract: Objektorientierung ist bereits fester Bestandteil des Informatikunterrichts an allgemeinbildenden Schulen. Derzeit fehlt es noch an passenden Werkzeugen, um die Konzepte für Schüler aufzubereiten und zu visualisieren. Das Programm „Trisda the Robot“ ist ein solches Werkzeug, mit dem die Kernkonzepte Klassen, Objekte, Vererbung und Polymorphie mit Hilfe eines Roboters visuell dargestellt werden können. Durch seine interaktive und intuitive Steuerung bietet es eine geeignete Lernumgebung für Schüler.

1 Einleitung

Objektorientierte Programmierung ist in der Softwaretechnik bereits etabliert und hält auch in den Informatikunterricht Einzug. Beispielsweise sollten die Grundprinzipien der objektorientierten klassenbasierten Programmierung bereits nach dem Thüringer Informatiklehrplan an Gymnasien von 1999 im Unterricht behandelt werden [TKM99]. Es wurden in diesem Zusammenhang die Begriffe Objekt, Klasse, Kapselung, Vererbung und Polymorphie genannt. Diese sollten theoretisch besprochen werden, der Einsatz einer Programmiersprache ist erst seit dem neuen Lehrplan von 2012 verpflichtend. Es sind nun sowohl objektorientierte Modellierung als auch deren Implementierung vorgesehen [TMB12]. Um Schülern die Konzepte anschaulich zu vermitteln, entstand das Programm „Trisda the Robot“. Es greift die Idee „Lasst uns kleine Welten schaffen“ [Fo11] auf und erschafft für den Schüler eine überschaubare Welt der Objektorientierung, sodass sie die Konzepte erlernen können, ohne dass sie eine Programmiersprache beherrschen müssen.
Der Schwerpunkt liegt auf der Unterstützung der Schüler beim Erlernen von informatikspezifischen Kompetenzen im Bereich der Objektorientierung [Fo10, S.74], wie sie aus den Anforderungen der EPA Informatik heraus konkretisiert wurden. Bei „Trisda the Robot“ handelt es sich also um ein Lernprogramm für den schulischen Einsatz. Es geht dabei nicht darum, Quelltexte zu analysieren, sondern die Konzepte auf spielerische Weise kennenzulernen.

2 Die Konzepte der objektorientierten Programmierung und deren Umsetzung

2.1 Klasse und Objekt

Klasse und Objekt sind die Grundbausteine der Objektorientierung. Eine Klasse beschreibt allgemein, welche Eigenschaften und welches Verhalten Objekte besitzen, die aus dieser Klasse erzeugt werden. Anders herum betrachtet ist eine Klasse eine Zusammenfassung von gleichartigen Objekten unter einem Oberbegriff. Man spricht in diesem Fall von einer Klassifikation. Beide Sichtweisen beschreiben ein und denselben Sachverhalt. Es ist hierbei besonders wichtig, den Schülern den Unterschied zwischen diesen beiden klar zu machen, z.B. indem man veranschaulicht, dass Klassen eine Art Schablone für Objekte darstellen, nach denen diese erzeugt werden können.
Das Programm „Trisda the Robot“ greift diesen Vergleich auf, denn hier werden Roboter anhand von Bauplänen erstellt. Ein Bauplan besteht in diesem Fall aus Bauteilen und Funktionen, die ein Roboter besitzt, wenn er nach diesem Plan erzeugt wird. Baupläne stellen demnach Klassen dar und Roboter entsprechen Objekten. Folglich können Bauteile als Variablen betrachtet werden und Funktionen als Methoden. Um die Trennung zwischen Klassen und Objekten noch deutlicher darzustellen, ist das Programmfenster in zwei Bereiche aufgeteilt (siehe Abb. 1). Der erste Bereich ist der Entwurfsbereich, in welchem die Baupläne erstellt werden. Erzeugt man einen Roboter aus einem Bauplan, so wird dieser auf dem zweiten Bereich, dem Testareal, abgesetzt. Jeder Roboter, der sich auf dem Testareal befindet, kann nun gesteuert werden, indem man die vom Bauplan vorgegebenen Funktionen nutzt. Dies erfolgt ausschließlich durch Mausklicks.
Besitzt ein Roboter beispielsweise das Bauteil „Greifarm“, so stehen dem Roboter die Funktionen „greifeZu()“ und „lasseLos()“ zur Verfügung. Hier greift auch das Prinzip der Kapselung, denn der Greifarm kann nur über diese Schnittstellen gesteuert werden. Wie das Bauteil im Einzelnen funktioniert, bleibt verborgen.

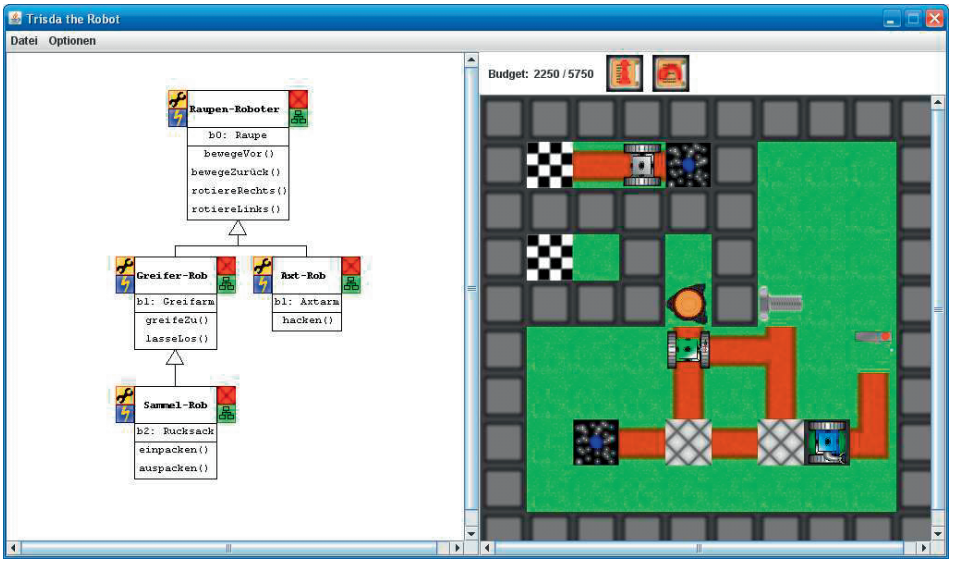


Abbildung 1: Aufbau des Programms. Links: Entwurfsbereich; Rechts: Testareal

2.2 Vererbung

Klassen stellen nach den obigen Ausführungen Klassifikationen dar. Bei der Modellierung von Problemen aus der realen Welt kommt es vor, dass es zu einer Klasse mehrere Unterklassen gibt, sodass eine Hierarchie entsteht. Eine Unterklasse erbt von der Oberklasse alle Variablen und Methoden und kann zusätzliche Variablen und Methoden beinhalten.
Vererbung wurde im Programm durch die sogenannte Weiterentwicklung der Baupläne umgesetzt. Wird ein Bauplan weiterentwickelt, so entsteht ein neuer Bauplan, der alle Bauteile und Funktionen der alten Version besitzt und in dem auch zusätzliche Bauteile und Funktionen hinzugefügt werden können.
Ein besonderer Effekt, der beim Vererben von Klassen auftreten kann, ist das Überschreiben. Wird eine Methode der Oberklasse in der Unterklasse unter dem gleichen Namen und mit den gleichen Parametern neu definiert, so wird die geerbte Methode überschrieben. Dies bedeutet, dass in einer Unterklasse die Methoden der Oberklasse angepasst werden können und immer nur die aktuellsten Methoden zur Verfügung stehen.
Auch dieses Konzept wurde im Programm realisiert. Wird eine gleichnamige Funktion in zwei Bauplänen verwendet, wobei ein Plan eine Weiterentwicklung des anderen darstellt, so kann nur die Funktion des weiterentwickelten Plans verwendet werden.

2.3 Beziehungen zwischen Objekten

Beziehungen zwischen Objekten ist ein wichtiger Aspekt, der insbesondere bei der Modellierung eine wesentliche Rolle spielt. Es wird zwischen drei Beziehungs-Typen unterschieden: Ist-, Hat- und Benutzt-Beziehung.

- Die ***Ist-Beziehung*** repräsentiert die Vererbung, das heißt ein Objekt B, welches von einem Objekt A erbt, steht in einer Ist-Beziehung zu A. Diese Beziehung ist, wie bereits beschrieben, mit der Weiterentwicklung eines Bauplans umgesetzt worden.

- Die ***Benutzt-Beziehung*** gibt an, dass ein Objekt der Klasse A die Methoden eines Objektes der Klasse B aufruft oder auf dessen Variablen zugreift.
  Im Programm wird dies deutlich, wenn der Roboter auf dem Testareal mit anderen Objekten interagiert. So kann er beispielsweise Gegenstände aufnehmen und Schalter aktivieren. Der Roboter nutzt also die Funktionen anderer Objekte, um eine eigene Funktion auszuführen.

- Die ***Hat-Beziehung*** besteht immer dann, wenn ein Objekt aus mehreren anderen Objekten zusammengesetzt ist. Es besteht also aus diesen Objekten.
  Der Unterschied zur Benutzt-Beziehung ist häufig nur schwer zu erklären, jedoch bietet das Programm ein äußerst anschauliches Beispiel hierfür. Der Roboter besteht aus Bauteilen, welche ebenfalls Objekte sind und ist somit aus diesen Objekten zusammengesetzt. Sein Verhalten wird durch sie definiert. Er benutzt diese Objekte also nicht nur, sondern er besteht auch aus ihnen.

2.4 Ausblick: Polymorphie

Um Polymorphie zu erklären, benötigt man zwei weitere Begriffe: dynamischer und statischer Typ eines Objektes. Der statische Typ eines Objektes ist dabei die Klasse, unter der das Objekt deklariert wurde. Der dynamische Typ ist die Klasse, der das Objekt zur Laufzeit des Programmes zugeordnet wird. Die Eigenschaft, dass Variablen auf Objekte von verschiedenem Typ zeigen können, nennt man in objektorientierten Sprachen Polymorphie [Mö11]. Diese Eigenschaft ist immer dann nützlich, wenn ein Programm mit verschiedenen Unterklassen arbeiten soll, wie es bei Listen oder Bäumen häufig der Fall ist. Es fördert auch die Erweiterbarkeit von Programmen, da Methoden des „alten“ Programms mit Objekten des „neuen“ Programms arbeiten können, sofern sie aus den bestehenden Klassen abgeleitet wurden.

Im Programm sind bislang die wesentlichen Konzepte umgesetzt, es bietet aber auch noch viel Potenzial für weitere Ideen und Weiterentwicklungen. Dazu gehört auch die Polymorphie. Im Programm wird diese anhand der unterschiedlichen Reaktionen des Roboters auf die verschiedenen Felder des Testareals visualisiert. Beispielsweise ereignet sich beim Betreten eines einfachen Feldes etwas anderes, als beim Betreten eines Schalterfeldes. Um die Polymorphie noch greifbarer zu machen, soll folgende Idee zur Weiterentwicklung genannt werden, die den Roboter als Objekt mehr in den Vordergrund rückt: Jeder Roboter verfügt über eine Funktion „wartungDurchführen()“.

Ruft man diese auf, wird jedes Bauteil einmal bewegt, um zu zeigen, dass es noch funktioniert. Diese Funktion kann von einem „Wartungsfeld“ auf dem Testareal aufgerufen werden, wenn der Roboter dieses Feld betritt. Da jeder Roboter aus verschiedenen Bauteilen zusammengesetzt wurde, diese jedoch alle die Funktion unterstützen, wird deutlich, dass das Feld mit unterschiedlichen Robotern arbeiten kann. Dies ist nach der Definition von Mössenböck [Mö11] Polymorphie.

3 Anmerkungen zum Programm

„Trisda the Robot“ wurde mit der objektorientierten Programmiersprache Java erstellt. Die verschiedenen Felder, Bauteile und Roboter werden auch intern in objektorientierter Art und Weise realisiert. Beispielsweise gibt es eine abstrakte Oberklasse „Feld“, von der jede spezielle Feldart abgeleitet werden muss.[1] Dies ermöglicht es, das Programm durch zusätzliche Feldarten zu erweitern. Analoges gilt für Bauteile und Gegenstände. [2]

Man kann also sagen, dass das Programm neben der Visualisierung der Objektorientierung diese auch selbst umsetzt. Somit wurde nicht nur die Erweiterbarkeit des Programms realisiert, sondern auch ein gewisses Maß an Authentizität sichergestellt.

Literaturverzeichnis

[Fo10] Fothe, M.: Kunterbunte Schulinformatik. Ideen für einen kompetenzorientierten Unterricht in den Sekundarstufen I und II. Berlin: LOG IN Verlag, 2010.

.

[Fo11] Fothe, M.: Lasst uns kleine Welten schaffen! KARA, PUCK und die optische Telegrafie. In: LOG IN, 31/32 Jg. (2011/2012), H. 172/173, S. 40 - 44.

[Mö11] Mössenböck, H.: Sprechen Sie Java? Eine Einführung in das systematische Programmieren. Heidelberg: dpunkt-Verlag, ^{4}2011.

[TKM99] TKM - Thüringer Kultusministerium (Hrsg): Lehrplan für das Gymnasium. Informatik. Erfurt: TKM, 1999.

[TMB12] TMBWK - Thüringer Ministerium für Bildung, Wissenschaft und Kultur (Hrsg): Lehrplan für den Erwerb der allgemeinen Hochschulreife. Informatik. Erfurt: TMBWK, 2012.

[1] Das Testareal besteht aus mehreren quadratischen Feldern. Die Klasse *Feld* und alle daraus abgeleiteten Klassen legen fest, welche Eigenschaften diese Felder besitzen und wie sie beim Kontakt mit Robotern reagieren.

[2] Gegenstände befinden sich ebenfalls auf dem Testareal und können beispielsweise von einem Roboter aufgehoben oder entfernt werden. Realisierte Gegenstände sind beispielsweise *Kiste*, *Schraube* und *Kristall*. Weiterhin wurden bereits die Bauteile *Greifarm*, *Axtarm*, *Raupe* und *Rucksack* implementiert.

Persuasive Design für Second-Screen-Anwendungen bei TV-Übertragungen

Matthias Merk
Matthias.Merk@Student.Reutlingen-University.de
Semesterarbeit „Interaktive Systeme“
Betreuer: Dr. Kai Holzweißig

Abstract: Für die Werbebranche entwickeln sich Second-Screen-Anwendungen zu einer vielversprechenden Werbeplattform. Durch gezieltes persuasive Design einer Second-Screen-Anwendung, insbesondere die Werbeintegrationen und die Art und Weise, wie der Benutzer zum Ansehen der Werbung animiert wird, kann das Verhalten der Nutzer hinsichtlich einer Kaufentscheidung beeinflusst werden. Dieses kann, ohne dass die Werbung als störend empfunden wird, die Werbewirksamkeit nachhaltig steigern, so die Hypothese. Anhand einer prototypischen Implementierung zeigt die Ausarbeitung, wie Werbeeinblendungen auf Second Screens mit Hilfe von Methoden des Persuasive Designs optimiert werden können.

1 Einleitung

Der Begriff Second Screen beschreibt Endgeräte, die zeitgleich zur Nutzung des Fernsehgerätes benutzt werden. Ein Second Screen stellt zusätzliche Inhalte und Interaktionsmöglichkeiten zum TV-Bild bereit. Die häufigsten Vertreter der Second Screens sind laut einer Studie der NPD Group Notebooks und Desktop PCs (60 %), Smartphones (55 %) und Tablets (49 %) [Gro13]. Elkington sieht die Zukunft der Second Screen Nutzung in der aktiven Beeinflussung der TV-Inhalte durch den Zuschauer [Elk14]. Durch diese neu geschaffene Interaktionsmöglichkeit stellen Second-Screen-Anwendungen eine zunehmend attraktive Werbevariante für Unternehmen dar. So tätigen bereits heute 45 % der Tablet-Nutzer während des Fernsehens online Einkäufe und 26 % suchen nach weiteren Informationen über Produkte, die in der TV-Werbung zu sehen sind [New13]. Martinolich sieht in Second-Screen-Anwendungen die Grundlage für eine starke Kundenbindung für TV-Sender [Mar12]. In vielen Fällen wird die Werbeintegration in Second-Screen-Anwendungen als störend empfunden. Laut Mancusco und Stuth sind 48 % der Second Screen Nutzer mit Werbeeinblendungen bei Second-Screen-Anwendungen sozialer Netzwerke unzufrieden. Sie wünschen sich stärker personalisierte Werbeeinblendungen [MS13]. Ziel dieser Arbeit ist es, die Möglichkeiten für eine nicht störende Werbeintegration aufzuzeigen. Dazu wird die Werbeintegration einer Second-Screen-Anwendung analysiert und unter Zuhilfenahme persuasiver Methoden konzeptuell überarbeitet. Durch den Einsatz von Persuasive Design soll so eine Werbewirksamkeit erreicht werden, wie man sie von klassischen Product Placements bei TV-Übertragungen kennt. Unter einem Product Placement (dt. Produktplatzierung oder Werbeintegration)

versteht man das Einbinden von Markenprodukten in ein Medium, ohne dass der Konsument es als störende Werbung wahrnimmt.

2 Persuasive Design

Sharp et al. definieren Persuasive Design als Schnittmenge der Techniken, die die Aufmerksamkeit des Betrachters auf spezielle Informationen lenken, um Handeln oder Denken zu beeinflussen. [RSP11, S. 195]. Ein mit persuasiven Methoden gestaltetes Interaktives System dient dem Zweck, das Verhalten oder die innere Einstellung einer Person zu ändern [Fog03, S. 37]. Diese Beeinflussung geschieht durch Überzeugung, niemals durch Täuschung, Zwang oder Manipulation des Nutzers. Für den Einsatz persuasiver Technologien im IT Umfeld begründete Fogg den Begriff „Captology“, welcher als Schnittmenge von Persuasive Design und Computertechnologie definiert ist [Fog03, S. 15ff]. Soll die Verhaltensweise des Nutzers durch den Einsatz von Persuasive Design beeinflusst werden, ist ein grundlegendes Verständnis der Verhaltensweisen des Nutzers notwendig. Fogg entwickelte hierzu in [Fog09] ein einfaches Verhaltensmodell: Das Fogg Behaviour Model. Laut Fogg setzt sich das Verhalten einer Person aus Motivation, Befähigung und einem Auslöser zusammen. Die Motivation beschreibt den Grund, warum etwas getan wird. Die Befähigung besagt, dass die Person in der Lage sein muss, etwas zu tun. Unter einem Auslöser versteht man einen Impuls, der die Person dazu bewegt etwas zu tun. Die Motivation und die Befähigung stehen in einem direkten Zusammenhang zueinander. Durch eine hohe Motivation besteht die Chance, dass das Zielverhalten auch dann erreicht wird, wenn die Befähigung dazu nur gering ist.

3 Methoden des Persuasive Designs

Fogg beschreibt folgende Methoden für die Umsetzung des Persuasive Designs bei bildschirmgestützten Systemen: Pop-Ups, Warnhinweise, Benachrichtigungen, personalisierte Nachrichten, Empfehlungen, Interaktive Medien und Verlinkungen. All diese Methoden sind laut Fogg am effektivsten, wenn sie eine Interaktion mit dem Benutzer zulassen [Fog03, S. 7ff]. Die technischen Eigenschaften moderner Second Screens erlauben die Umsetzung persuasiver Methoden auf vielfältige Weise. So können zum Beispiel Warnhinweise durch haptisches Feedback (Vibrieren des Second Screen), durch visuelles Feedback (helles Aufblitzen des Displays) oder durch einen Benachrichtigungston umgesetzt werden. Auch können personalisierte Nachrichten durch die Verwendung von Standortdaten des Benutzers stärker personalisiert werden. Das Vorhandensein von unterschiedlichen Geräten zur Mediennutzung zur selben Zeit ist laut J. Romaniuk ein großes Problem für die Werbebranche, da sich die Aufmerksamkeit des Betrachters auf immer mehr Geräte aufteilt [Rom12]. Im hier betrachteten Fall teilt sich die Aufmerksamkeit des Nutzers zwischen TV-Gerät und Second Screen [Gro13]. Es ist also notwendig, die Aufmerksamkeit des Zuschauers bei Bedarf auf den Second

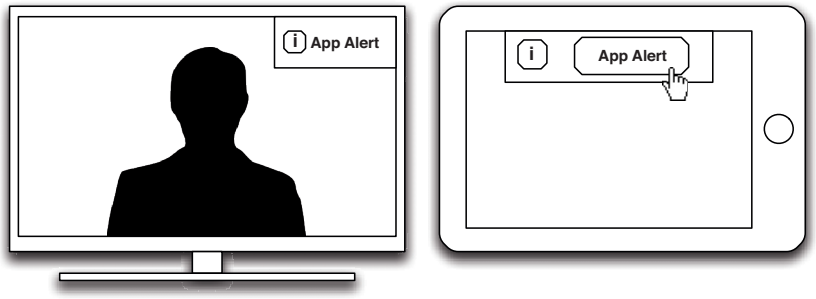


Abbildung 1: Interaktion und Aufmerksamkeitsübergang zwischen TV und Second Screen)

Screen zu lenken, ohne dass sich der Zuschauer abgelenkt fühlt. Nur so kann eine Werbeeinblendung auf dem Second Screen auch wahrgenommen werden. Der Auslöser dieses Aufmerksamkeitsüberganges kann vom TV-Gerät und / oder vom Second Screen ausgehen.

4 Prototypische Optimierung einer Second-Screen-Anwendung

Grundlage des Prototyps ist die Android-Version der Second-Screen-Anwendung „RTL Inside“ (Stand 26.06.2013). Es besteht keine Kooperation mit RTL, die Umsetzung erfolgt rein konzeptuell im Rahmen einer Seminararbeit. Werbeeinblendungen bei „RTL Inside“ befinden sich am oberen und unteren Rand in Form von Werbebannern. Die Werbung hat keinerlei Zusammenhang mit den im TV gezeigten Inhalten. Die Optimierung wird exemplarisch an einer Umfrage vorgenommen. Grundlage der Maßnahmen ist das Verhaltensmodell von Fogg. Das beabsichtigte Wunschverhalten des Benutzer ist es, nach einer abgeschlossenen Umfrage auf eine Werbefläche zu klicken. Als Auslöser der Interaktion dient ein am oberen Rand des TVs eingeblendeter Hinweis. Dieser Hinweis wird durch einen Hinweiston ergänzt. Hat der Benutzer die Second-Screen-Anwendung bereits geöffnet, gibt auch das Tablet Vibration und visuelles Feedback durch Aufblitzen des Displays zurück. Um die Verknüpfung zwischen Second Screen und TV-Bild zu schaffen, wird der selbe Hinweiston mit einem kurzen Zeitversatz auch auf dem Second Screen abgespielt. Um die Befähigung zur Durchführung der Interaktion zu verstärken, wird auch in der Second-Screen-Anwendung eine zur Benachrichtigung im TV-Bild korrespondierende Benachrichtigung am oberen Bildschirmrand angezeigt. Berührt der Nutzer diese Benachrichtigung, wird er direkt zur Umfrage weitergeleitet (siehe Abbildung 1). Diese Maßnahme beschränkt die notwendigen Schritte um zur Umfrage zu gelangen auf ein Minimum und steigert damit die Befähigung, die Umfrage durchzuführen. Von Praktiken, wie dem kompletten Ausblenden der Bildinhalte, sei an dieser Stelle abgeraten. Dies würde den Nutzer zum Handeln zwingen, was gegen die

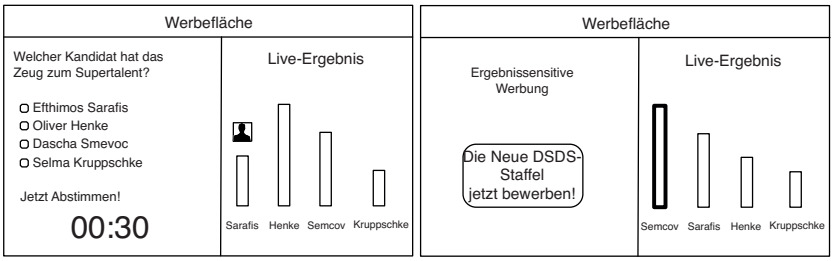


Abbildung 2: Konzept einer Umfrage auf dem Second Screen (links Umfrageansicht, rechts Ergebnisansicht)

Grundsätze des Persuasive Designs spricht (siehe Abschnitt 3). Die Optimierung der Befähigung sowie der Auslöser bleiben für alle Interaktionsmöglichkeiten der Second-Screen-Anwendung dieselben. Um aber die Motivation des Nutzers zu erhöhen, ist es notwendig, die Maßnahme dem Interaktionstyp anzupassen. Die Motivation des Nutzer zum Ausfüllen eines Spendenformulares unterscheidet sich von der Motivation zur Teilnahme an einem Gewinnspiel. So könnte die Motivation zum Spenden beispielsweise durch drastisches Bildmaterial (hungernde Kinder, kränkliche Tiere etc.) erhöht werden, während die Abbildungen möglicher Gewinne die Motivation zur Teilname am Gewinspiel erhöht. Es gibt aber auch vom Interaktionstyp unabhängige Optimierungsmaßnahmen. Blendet man einen Countdown auf der Benachrichtigung des Second Screens ein, symbolisiert dieser eine künstliche Verknappung und kann den Nutzer zum Handeln motivieren. Abbildung 2 zeigt die konzeptuelle Umsetzung. Am oberen Rand der Applikation befindet sich ein Werbebanner, das wechselnde Informationen zum Programm des Senders darstellt. Auf der linken Bildschirmhälfte befindet sich die Umfrage mit ihren Antwortmöglichkeiten. Der Aufruf „Jetzt abstimmen!“ und ein Countdown animieren den Nutzer zum Beantworten der Frage. Wenn der Nutzer abgestimmt hat, wird in der rechten Bildhälfte das Endergebnis dargestellt. Stimmt ein Kontakt des Nutzers aus sozialen Netzwerken des Nutzers ab, wird dies durch ein kleines Icon mit dem Profilbild des Freundes symbolisiert. Die linke Seite wird nun zu einer zur Gewinnerantwort thematisch passenden Werbefläche. Im hier benutzten Beispiel fordert nun die ergebnissensitive Werbefläche auf der linken Bildschirmseite zur Anmeldung für die kommende Staffel eines Talentwettbewerbs auf.

5 Fazit

Persuasive Design ist ein effektives Werkzeug. Auch für Konsumenten ergeben sich durch den Einsatz des Persuasiv Designs Vorteile. Thematisch passende Werbung wird vom Betrachter beispielsweise als weniger störend empfunden als thematisch unpassende. Ob man durch persuasive Methoden optimierte Werbeeinblendung als Product Placement betrachten kann müssen abschließende Nutzertests noch zeigen. Durch die Anwendung

des Verhaltensmodells von Fogg, wurde die theoretische Grundlage dafür gelegt. Ein gezielt abgesetzter Aufmerksamkeitsübergang hilft dabei, die Aufmerksamkeit des Nutzers zu lenken. Dies ist notwendig, da sich die Aufmerksamkeit des Nutzers immer zwischen TV und Second Screen aufteilt [Gro13]. Die durch eine Benachrichtigung auf dem Second Screen optimierte Befähigung des Nutzers hilft dabei sein gewünschte Verhalten zu erreichen. Werbeintegrationen in digitalen Medien sollten sich in Zukunft weg von Ablenkung und Zwang bewegen. Um dies zu erreichen, können die Methoden des Persuasive Design gewinnbringend für Anbieter und Nutzer aufgegriffen werden. Dass dazu die Möglichkeit besteht, wurde durch die Analyse und prototypische Optimierung von „RTL Inside“ im Rahmen dieser Ausarbeitung gezeigt. Die Relevanz des Persuasive Design im hier dargestellten Anwendungsfall wurde einige Monate nach Fertigstellung der Arbeit deutlich. RTL rüstete in einem Update von „RTL Inside“ die hier beschriebenen Benachrichtigungen auf dem TV nach.

Literatur

[Elk14] Tim Elkington. The second screen. *Market Leader*, Ausgabe Q1, 2014.

[Fog03] B.J. Fogg. *Persuasive Technology - Using Computers to Change What We Think and Do*. Morgan Kaufmann, San Francisco, Calif, 1. Auflage, 2003.

[Fog09] B.J. Fogg. A behavior model for persuasive design. In *Proceedings of the 4th international Conference on Persuasive Technology*. ACM, 2009.

[Gro13] NPD Group. Divided viewing: Second screens vying for TV viewers' attention, April 2013.

[Mar12] Jim Martinolich. The SECOND screen. *Broadcast Engineering (World Edition)*, Ausgabe 54, 2012.

[MS13] Joan Mancusco und Karen Stuth. Social Media and the Second Screen. *Marketing Insights*, Ausgabe 25, 2013.

[New13] Multichannel News. The second-screen landscape. *Multichannel News*, Seite 22, Januar 2013.

[Rom12] Jenni Romaniuk. Lifting the productivity of TV advertising : nothing matters more than the brand, nothing. *Journal of advertising research*, Ausgabe 52, 2012.

[RSP11] Yvonne Rogers, Helen Sharp und Jenny Preece. *Interaction Design - Beyond Human Computer Interaction*. John Wiley & Sons, New York, 3. Auflage, 2011.

Transformation of generic user interfaces into a web-based representation for network document scanners

Carsten Pape

University of Applied Sciences Osnabrück
Faculty of Engineering and Computer Science
carsten.pape@p4p3.net

Type of work: Bachelor thesis
Supervisors: Prof. Dr. Frank M. Thiesing, University of Applied Sciences Osnabrück (GI liaison lecturer)
Dipl.-Inf. (FH) Michael Rosemann, NT-ware Systemprogrammierung GmbH, Bad Iburg

Abstract: This paper covers the interpretation of generic user interface definitions within a web standards context. Asynchronous screen and content updates, also based on a generic data interchange format, contribute to the dynamic nature of the defined system. Reflecting one of the main challenges of the bachelor thesis, the goal was to provide a well-performing on-the-fly generation of these user interfaces. Suitable conversion approaches have been designed for an embedded, web-based scanning client with support of a Microsoft .NET-driven server application.

1 Introduction

Many company workflows rely on paper-based information as input data. For efficient electronic processing, these documents need to be captured via either dedicated network document scanners or multifunction peripherals. The different scan workflows can be supported by software solutions that offer sophisticated image processing and integration with complex business processes.

Scan workflows not only have an impact on the server-side processing, but also on the user interaction through device-integrated touchscreen displays. The need for highly adaptive and responsive UIs (user interfaces) led to the invention of an "Apparatus for providing a user interface" [HR08], used in the print, scan, and device management software solution *uniFLOW* of the company NT-ware Systemprogrammierung GmbH[1]. This patented UI technology decouples the platform code on the scanning devices from the UI layout and workflow definition. XML is used for the underlying description of all UI-related elements, actions and the responses to asynchronous server requests, which trigger state and representation changes.

[1] http://nt-ware.com/

Within the scope of the bachelor thesis, this existing technology was made available for a new generation of web-based network document scanners. As simple web clients with an embedded browser, such scanners require the XML data to be represented in XHTML markup, augmented by the common web standards CSS and JavaScript. The developed transformation concept covers a central part of the bachelor thesis and is outlined throughout this document.

2 XML Processing Approaches

XML was designed to be easily processable by programs [W3C08b]. The common term *parsing* in this context refers to the interpretation of characters in an XML string by breaking down and separating parts of the XML data [Ben03]. Parsing can happen in several ways, looking at the XML either as plain text, as a stream of events or as a tree structure [HM04].

For text-based XML processing, regular expressions can be used to match and likewise replace tags, attributes and text. However, the use is limited to a very simple document structure [HM04]. The main pitfall is that in XML semantically identical information can be represented in a variety of syntactically different ways [See W3C08a].

Tree-based models build up a hierarchical tree structure of node elements. Once successfully parsed, this object model represents the entire XML document, with the context for any requested part of the document always available. On the contrary, memory consumption is remarkably high for large documents [HM04].

Event-driven and pull-based XML parsers both process the document sequentially and make the content available progressively. Event-based parsers report all occurrences of key information to the client application, thus representing a *push* model. The *pull* model, however, requires the developer to explicitly request the next document portions from the parser, leading to more readable code [HM04]. Both stream-based approaches are united in the need to process data as soon as it is received [Von11].

Another type of programming interface to XML is the declarative specification of transformations. The W3C XSL Transformations (XSLT) specification describes a language for transforming an XML source document to another XML or also HTML or even plain text document. To generate the output, an XSLT processor reads in an XSLT stylesheet along with the source data. The vocabulary of a stylesheet is made up of XSLT directives and functions together with XPath expressions to address subsets of the input document [Von11; Ben03].

3 Conversion Concept

To reasonably describe the chosen concept for the generation of web-based UIs, the underlying software architecture and technologies are briefly introduced: A server-side component interprets the XML UI definitions and generates dynamic web pages,

including style definitions and scripting code. This data is pushed to a single page web application on the scanner using a real-time communication framework. Server requests are generated from application state information and also represented in XML, similar to the appropriate server responses. Execution environment for the server part is ASP.NET MVC 4, bound to the .NET Framework 4.0-given XML feature set, particularly meaning XSLT 1.0 [See @XSL].

XML data occurs in different scenarios throughout scan workflow processing and can be logically separated into the fragments shown in table 1. The appropriate target respectively source format for the conversion is specified in each case.

XML data		Target/source format
Layout, components and content definition	→	XHTML 1.0
Styles and themes	→	CSS 2.1
Events, actions and conditions	→	JS function calls with JSON parameters
Asynchronous server requests	←	JS application state
Asynchronous server responses	→	JS function calls with JSON and XML parameters
Programmatic content updates	→	JS data transfer objects (DTOs) for internal use

Table 1: Occurrence of XML data with its appropriate conversion target or source format

XSLT fits perfectly for the main XHTML generation, thus transforming from one XML vocabulary to another. For many markup elements, information needs to be combined from distinct XML parts, which can be achieved through XPath navigation inside the always available document context. Stylesheets can be modularly structured based on templates. Using a *push* design with multiple specialized templates increases the reusability and provides an efficient handling of recursively nested elements [Kay08].

Being able to write fixed parts of the output document directly into the stylesheet, predestines the XSLT approach also for JavaScript function call generation. Parameters are formed in JSON by iterating over the elements. CSS definitions represent element styles and overall theme information and need to be prefixed with appropriate IDs and classes, which are again fetched from the context via XPath expressions. The generation is also accomplished via XSLT, resulting in three independent stylesheets that operate on the same input document. Targeting an optimal performance, these transformations are therefore invoked as parallel threads.

The missing feature set of especially needed string processing functions in the XSLT 1.0 standard is made available through recursive templates and dynamic XPath expressions. To reuse the templates across all XSLT stylesheets, they are exported to a utility stylesheet and imported on demand.

Not all conversions are straight forward, but rather involve multiple parsing steps. The designed stylesheets – being XML files on their own – contain entity definitions for the integration of external information in a structured way. Their value definitions are

replaced by regular expressions when the associated data changes, reported through an Observer pattern. The common source document for the transformations is held in a tree-based Document Object Model (DOM). Before passing this DOM to the XSLT processor, certain attribute values are manually validated and adjusted if required. This leads to the design of different document processing pipelines, expressing an effective, hybrid XML conversion approach.

In a web browser, the DOM is the prevalent approach to handle XML data. Browser APIs differ, but have a parsing and serializing functionality in common. The browser-side also features a wide-spread XML and XSL support, is however also limited to the first version of the XSLT standard [See @XSLT]. XML creation is solved through programmatic appending of new elements to a temporary DOM with a subsequent serialization to an XML string. Server responses represent only a fraction of the UI definition complexity but can be treated with the same declarative XSLT approach inside the browser. Generated JavaScript function calls delegate to a business logic façade that manages the internal processing. The content updates are also handled on client-side with the conversion target being JavaScript DTOs. Creation of the objects is achieved by programmatically accessing the XML data made available through a DOM.

4 Conclusion and Outlook

Backed by a smart combination of XML technologies, a module for the transformation of generic UI and data definitions into common web standards has been designed and implemented. It serves as a foundation for the intended overall software product and future web-based application scenarios. Slight improvements are planned for the test methodology of the declarative XML to XHTML conversion. At the moment, testing is achieved through reparsing and manually validating the generated output. Interesting approaches to evaluate would be XSLT-integrated unit testing [Man06] or a validation based on custom XML Schemas.

5 References

[@XSL] Microsoft Corporation; *System.Xml.Xsl Namespace*, http://msdn.microsoft.com/en-us/library/System.Xml.Xsl(v=vs.100).aspx, accessed 17 Jan 2014

[@XSLT] W3Schools; *XSLT Browsers*, http://www.w3schools.com/xsl/xsl_browsers.asp, accessed 17 Jan 2014

[Ben03] Benz, B.; *XML Programming Bible*. New York. Wiley, 2003

[HM04] Harold, E. R.; Means, W. S.; *XML in a Nutshell*. Sebastopol, CA. O'Reilly, 2004

[HR08] Huster, K.; Rosemann, M.; *Apparatus for providing a user interface,* Patent EP 1 983 427 A1, 22.10.2008

[Kay08] Kay, M.; *XSLT 2.0 and XPath 2.0: Programmer's Reference*. Indianapolis, IN. Wiley, 2008

[Man06] Mangano, S.; *XSLT Cookbook*. Beijing, Sebastopol, CA. O'Reilly, 2006

[Von11] Vonhoegen, H.; *Einstieg in XML: Grundlagen, Praxis, Referenz*. Bonn. Galileo Press, 2011

[W3C08a] World Wide Web Consortium (W3C); *Canonical XML Version 1.1.* Recommendation, 02.05.2008, http://www.w3.org/TR/2008/REC-xml-c14n11-20080502/

[W3C08b] World Wide Web Consortium (W3C); *Extensible Markup Language (XML) 1.0,* Fifth Edition. Recommendation, 26.11.2008, http://www.w3.org/TR/REC-xml/

Behavior Based Web User Identification

Christian Schäff, Gaston Pugliese, Timo Götzelmann

Department of Computer Science
Technische Hochschule Nürnberg Georg Simon Ohm
Germany

{christian.schaeff, pugliesega45509, timo.goetzelmann}@th-nuernberg.de

Abstract: This paper examines different approaches for the identification of users by their personal behavior and discusses techniques which could be used in the context of websites. Such web tracking approaches have the potential to identify users even if they use multiple or shared devices. For web pages, mouse and touch input are widely used. Therefore, we propose a survey to evaluate the feasibility to identify users by their interaction behavior.

1 Introduction

Web tracking [LC] and web usage mining [CG13] are common techniques to identify users and aggregate user specific information into profiles. These profiles are useful to improve and optimize services but can also threat user's privacy. Especially, when the user can be re-identified within different domain contexts, the collected profiles can be merged and gradually cover large parts of each individual's life and habits.

Most cross domain tracking services use application specific identifiers, stored within the user's browser [CC09]. From a tracking perspective such storages are not very reliable, and with rising privacy awareness and stricter browser default settings even less. This led to new techniques based on device and browser specific criteria, known as fingerprinting, to distinguish users without stored identifiers [Sol11]. The more data is aggregated into a single profile, the higher the probability that the collected information itself is sufficient to identify the user unambiguously [Pau08, TWC12].

User behavior has been shown effective as an additional security factor for authentication, but could also be used for web tracking. We examine the feasibility of such an approach and how users could be detected. In particular, this research focuses on the question, in which cases exclusively behavioral criteria are sufficient to distinguish users.

2 Related Work

Identifiers based on biometric properties, in particular physiological and behavioral criteria, are used as additional security factors for authentication. Such identifiers are usually difficult to obtain and reproduce for attackers, but also can not be lost, forgotten, or transfered by users. There are application contexts which require to identify users without ambiguity. Additional biometric factors are often used — in regards to privacy those contexts are considered unproblematic. The following list highlights behavioral criteria used in other works to identify users:

- Keyboard interaction behavior is well researched and can distinguish users by comparing keystroke dynamics and other typing specific criteria [SP09]
- Mouse interaction creates a two dimensional path with user specific properties, but is highly defendant on the application context and hardware [ZPW11].
- Touch input behavior is also influenced by device and application circumstances, but can also calculate the pressure applied by the user [KDB+10]

In other application contexts the collection of user information and behavior is considered as a privacy violation, because it can be used to identify persons unnoticeable and without their agreement. The main challenge is to cope with the ambiguity of an identification only based on user behavior without knowledge of the total number of users. In reality, this uncertainty can be reduced by additionally considering communication and application specific data.

Eckersley [Eck10] showed with the use of browser fingerprints that web users can be identified successfully by evaluating network, device and software specific data. Devices shared among multiple users, e.g., public computers or within families, exhibit the same configuration. In these cases, browser fingerprinting identifies rather a device than individual users. In such situations interaction behavior could be used to identify and distinguish all device users. As the number of users is expected to be rather small, this approach looks quite promising.

Even more interesting is the question if users can be identified in two distinct contexts based only on behavioral data and how much data is needed to archive a certain confidentiality. Since pointing/cursor devices are the primary interaction method with web content, it is feasible to concentrate on those.

3 Identification Based on Mouse and Touch Interaction Behavior

In this section, we focus on investigating user-browser interaction behavior and if the user's identity can be deducted. To answer this question, the possibilities for cursor interaction have to be classified in order to distinguish device and user specific behavior. For each class, we define a set of detection criteria and approaches that can be tested in future surveys. Even if user identification solely based on interaction behavior turns out to be unfeasible, extending browser fingerprinting with behavior data could be quite powerful.

For now, we focus our research on mouse and touch screen devices and discuss how user specific information could be extracted.

3.1 Mouse Interaction

According to Chen et al. [CAS01] there is a relationship between eye and mouse movements. A popular application of this phenomenon is eye-tracking used for the evaluation of web pages. However, these results could also reveal some user specific information. Assuming that different user groups (e.g., female and male) get attracted from different visual stimuli this could affect the positioning of mouse, too. Beside that, mouse interaction offers a couple of further possibilities to gain information about biometric information of users and their technical equipment. First, parameters such as the mouse cursor acceleration and the mouse scroll wheel could be used to determine a constant value of the mouse settings. Secondly, the characteristic of the acceleration as well as the general movement could reveal user specific information. The well-known problem of over- and undershooting of a click target can be analyzed as well as characteristic. Finally, users may differ regarding their preferred mouse positions and their clicking behavior when reading long texts. We assume, that when interactive or hyperlinked elements are integrated with a multicolumn design (e.g., Google News) it is likely that the mouse position is preferably in a region where mouse clicks do not accidentally release some action, except to obtain the focus of the window.

3.2 Touch Interaction

In case of direct interaction by touch input the duration of taps could be specific for users. Furthermore, there could be slight marginal user specific vertical and horizontal differences of tap positions when users try to hit a target (e.g., hit a button). Beside that especially for mounted touch displays the usual position of the user relatively to the screen may affect the timings for tapping different positions on the screen. According to Fitts' law, for mounted as well as for portable touch screens, the actual size of the touch screen may influence timings when hitting targets. Additionally, for smartphone and tablet touch input further information could be gathered from the usual way of handling these portable devices. Henze et al. [HRB11] investigated touch behavior of smartphone users with a large number of participants. They found out that inputs are systematically skewed by several variables and propose to use their results for improving algorithms for touch interaction. However, their results encourage us to propose methods of smartphones' touch input to differentiate users. For one-handed smartphone touch interaction there are more and less convenient positions on the touchscreen. Measuring the timings to hit targets it could be supposed if the user preferably interacts one- or two-handed. Furthermore, it is likely that the timings differ between different users. If one-handed interaction is detected, the timings for left-aligned and right-aligned hit targets can be considered in order to determine if the user is left- or right-handed. Information about error rates in hitting targets (e.g., when using the virtual keyboard) could be exploited to speculate about movement disorders or the size of the user's fingers, especially for small screens.

4 Ongoing and Future Work

Our blind user study provides multiple tests where users get several presentations of visual stimuli whilst recording their mouse/touch interaction in a supervised context. In order to control the targeting and click behavior, inconspicuous dialogs will pop up whilst the user is solving varying tasks (e.g., playing a game or reading a text). These modal dialogs reappear at irregular intervals and always at different positions. The users must respond via click/touch in order to close them and continue with the given tasks. Mouse or touch events will be stored in the form of quadruples, i.e. $timestamp$, $abscissa$, $ordinate$ and $event$ (see [ZPW11]). We assume that this scenario, which is conceivable on any web page, determines sufficient behavioral characteristics to identify recurring users with relatively little data.

References

[CAS01] MC Chen, JR Anderson, and MH Sohn. What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. *CHI'01 extended abstracts on ...*, pages 281–282, 2001.

[CC09] Francesca Carmagnola and Federica Cena. User identification for cross-system personalisation. *Information Sciences*, 179(1-2):16–32, January 2009.

[CG13] Kamika Chaudhary and SK Gupta. Web Usage Mining Tools & Techniques: A Survey. *International Journal of Scientific & Engineering ...*, 4(6):1762–1768, 2013.

[Eck10] Peter Eckersley. How unique is your web browser? In *Privacy Enhancing Technologies*, pages 8–16. Springer, 2010.

[HRB11] Niels Henze, Enrico Rukzio, and Susanne Boll. 100,000,000 Taps: Analysis and Improvement of Touch Performance in the Large. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 133–142, New York, NY, USA, 2011. ACM.

[KDB+10] David Kim, Paul Dunphy, Pam Briggs, Jonathan Hook, John Nicholson, James Nicholson, and Patrick Olivier. Multi-touch authentication on tabletops. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, page 1093, New York, New York, USA, 2010. ACM Press.

[LC] Li Li and Wu Chou. Position Paper for W3C Workshop on Web Tracking and User Privacy. *w3c.org*.

[Pau08] Azigo Paul Trevithick. Privacy vs. Personalization Paradox in Online Advertising. *w3c.org*, page 3, 2008.

[Sol11] Ashkan Soltani. Identifiers and Online Tracking. *w3c.org*, 2011.

[SP09] D Shanmugapriya and G Padmavathi. A survey of biometric keystroke dynamics: Approaches, security and challenges. *arXiv preprint arXiv:0910.0817*, 5(1):115–119, 2009.

[TWC12] Eran Toch, Yang Wang, and Lorrie Faith Cranor. Personalization and privacy: a survey of privacy risks and remedies in personalization-based systems. *User Modeling and User-Adapted Interaction*, 22(1-2):203–220, March 2012.

[ZPW11] Nan Zheng, Aaron Paloski, and Haining Wang. An efficient user verification system via mouse movements. *Proceedings of the 18th ACM conference on Computer and communications security - CCS '11*, page 139, 2011.

Geo-Referenced dAtaVIsualizaTIon Framework: Presenting Weather Forecasts

Marcus Seiler

Software Engineering Research Group
Department of Electrical Engineering and Computer Science
University of Kassel
Wilhelmshöher Allee 73
34121 Kassel, Germany
marcus.seiler@student.uni-kassel.de

Abstract: In general, the visualization of geo-referenced data touches the everyday life. The processing of meteorological data such as the current weather forecast or the representation of population density are just two examples for this domain. Nowadays, this type of data must be provided both on classical computers and on mobile devices. On the one hand, a lot of tools and services exist, aiming to fetch and visualize geo-referenced data. On the other hand, they either restrict the visualization to predefined types or it can be difficult or impossible to extend them in order to enrich the functionality. This raises the need for combining the mining and visualization of geo-referenced data sets with a generic interface provided to developers that allows the extension of the given functionality. Therefore, this paper introduces Graviti: An extendable web framework for geo-referenced data set visualization. Currently, Graviti only focuses on the processing of weather forecasts but the approach can also be applied to other geo-referenced data.

1 Introduction

Applications for the visualization of geo-referenced data sets such as weather forecasts were limited exclusively for the use with traditional desktop computers. Over time, favored by the increasing spread of the internet, mobile devices and less expensive even more powerful hardware, more and more services and tools that provide the visualization of geo-referenced data and weather forecasts in special originated. One of these tools is a web-based visualization platform for meteorological data analysis [SSL+12]. However, this platform does not provide a dynamic user interaction with the displayed weather data, since the data is rendered on the server side and passed as static images to the client web interface. Another approach describes a framework that permits a remote access to weather forecasts with which the user can access, visualize and interact with the data through a web browser [LAAQ13]. A weakness of this approach is the use of a data pool that is designed for the management of individual files. Therefore, the Graviti web framework was developed to obtain, store and visualize geo-referenced data. Graviti uses the power

of a modern scalable database and different types of visualization on the client side. The framework aims to be as extensible as possible to provide developers the opportunity to easily implement new features.

The next section first delivers insight into the acquisition of geo-referenced data in special related to weather forecasts. Section 3 outlines the requirements, describes the implementation and covers the current state of the Graviti framework. Finally, section 4 finishes with a conclusion and presents future work.

2 Weather Forecast Data Collection

This section provides a brief overview of the weather forecast data collection. First, it is explained where weather data can be obtained. Secondly, it is discussed how the data can be managed and stored in an appropriate manner. In the following sections the term feature is used. A feature denotes a representative of one weather data type, for example, temperature or humidity.

The National Climatic Data Center[1] from the US Oceanic and Atmospheric Administration Center is only one source for retrieving weather forecast data. They offer freely available data in the Global Forecast System model, which is produced by the National Centers for Environmental Prediction[2] and can be obtained as gridded binary, so-called *GRIB* file - a data format for storing historical and forecast weather data. These files are updated several times a day over a period of six hours and include data of the current day as well as the forecasts of the following 192 hours. Since each file has a size of nearly 50 MB and the forecasts of 192 hours are covered in 64 files which are updated six times a day this leads to a data size of 19 GB per day. The records of a feature consist of a tuple of coordinates and the corresponding value. Therefore, one file contains 64800 data points for each feature. This yields to over 24 million records for one feature a day if the daily update and forecast rate is applied. Due to this large amount of data, the use of a database that specializes in high performance, scalability and the management of huge data masses such as *Apache Cassandra*[3] should be considered.

3 The Graviti Framework

The first part of this section covers the requirements that have been identified for the Graviti framework. The second part gives an overview of the Graviti architecture. Finally, the third part documents the current state of the framework.

The major requirement of the web framework is the visualization of various geo-referenced data sets like weather forecasts and the provision of different map types to visualize differ-

[1] http://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs

[2] http://www.ncep.noaa.gov/

[3] http://cassandra.apache.org/

ent types of data. The framework should be capable to periodically retrieve and manage a large amount of data. Developers may want to implement other features than weather forecasts or want to enrich the given functionality. Therefore, Graviti needs to be extensible. On the one hand, the visualization of a current snapshot of the processed data is needed. On the other hand, it has to be ensured that the framework is able to handle the visualization of data changes over a certain time. The performance and response of Graviti should be considered to provide the best compatibility across different devices including mobile phones and tablet computers. Finally, a dynamic user interaction like moving the map to an area of interest and the selection or highlighting of data is needed.

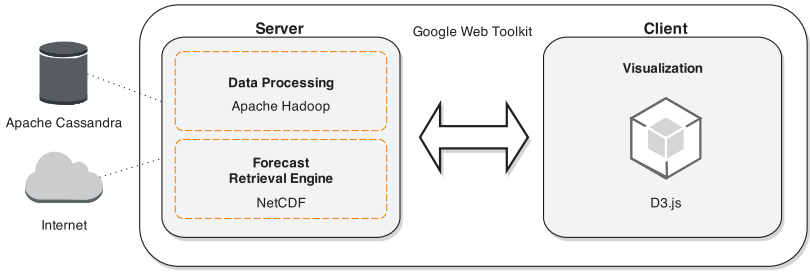


Figure 1: Graviti Framework Architecture

The implementation of the framework consists of three components. Figure 1 shows the architectural overview of the Graviti web framework. The first component is responsible for the retrieval of weather forecast data. The data is periodically fetched from a FTP-Server and passed to an *Apache Cassandra* database. The component uses the Java *NetCDF*[4] library which extracts the corresponding features from the downloaded *GRIB* files. The data is preprocessed within the second component to reduce the access times for reading the forecast data out of the database using *Apache Hadoop*[5]. The third component manages the connection between server and client and is also responsible for the visualization. For the creation of the web interface and the communication between server and client *Google Web Toolkit*[6] a framework for creating Java based web applications is used. The visualization of weather forecast data is build on top of *D3.js*[7]. The *D3.js* library provides the creation of dynamic interactive data visualization including but not limited to basic charts, heat maps and choropleths.

For visualizing weather and forecast data a world map is provided by Graviti. Two different map types are currently derived from the world map. Firstly, a map to display wind conditions such as direction and speed is implemented. Secondly, a generic heat map is implemented which provides the visualization of color-coded regions. The color-codes are determined by mapping the particular feature values into a corresponding color range using a linear scale function. In order to provide an example implementation, a surface

[4] http://www.unidata.ucar.edu/software/netcdf/
[5] http://hadoop.apache.org/
[6] http://www.gwtproject.org/
[7] http://d3js.org/

temperature map was derived from the generic heat map. Furthermore, each map type has an interface that can be used either to visualize weather conditions based on a single day or to display a forecast animation including several days. Since the performance of mobile devices is limited, Graviti offers an automatic reduction of the displayed feature rate by decreasing the resolution. This is achieved by reducing the transmitted feature set size if the framework determines a mobile platform at runtime. A web application that is build on top of the Graviti framework, can enable weather forecast visualization by simply adding one of the predefined map widgets.

4 Conclusion and Future Work

Almost every human is interested in both information about the current weather conditions as well as a forecast for the upcoming days. The most common information that is retrieved about the weather forecasts include temperature, humidity and wind conditions. Due to the ubiquity of mobile devices the visualization of weather data has to be enabled for both traditional computers and mobile devices. Therefore, this paper presented Graviti: A web framework for processing geo-referenced data sets. The framework combines the power of a modern scalable database and a dynamic web application interface to target desktop computers as well as mobile devices equally. Currently, Graviti only focuses on the processing and visualization of weather forecast data.

Due to the prototype status of the presented approach, there still is much to do: One of the major challenges is the adoption of other geo-referenced data. With such an integration choropleth maps that visualizes the results of a regional election could be generated. The integration of user interaction such as scroll and zoom to an area of interest on the map is planned. Currently, only one world map is provided for visualization. The integration of regional maps and different projection types is missing. By implementing the corresponding database functions more weather forecast features can be provided for visualization. With the implementation of a responsive web design, the size of the map and the placement of web elements can be dynamically adjusted and thus increases the usability on mobile devices. In addition to this, an evaluation of the Graviti framework is still missing. A first evaluation would be conceivable in form of a student project or seminar paper at the local department, for instance.

References

[LAAQ13] Maider Laka, Ion Alberdi, Kevin Alonso, and Marco Quartulli. Cloud based N-dimensional weather forecast visualization tool with image analysis capabilities. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, 2013.

[SSL+12] Xiaojuan Sun, Suhung Shen, Gregory G. Leptoukh, Panxing Wang, Liping Di, and Mingyue Lu. Development of a Web-based Visualization Platform for Climate Research Using Google Earth. *Comput. Geosci.*, 47:160–168, October 2012.

A High-Performance Hardware Accelerator for HEVC Motion Compensation

Matthias Göbel
Embedded Systems Architecture Group
Dept. of Computer Engineering and Microelectronics
Technische Universität Berlin

m.goebel@tu-berlin.de

Abstract: The presented master's thesis has focused on the design and implementation of a motion compensation hardware accelerator for use in HEVC hybrid decoders, i.e. decoders that contain hardware as well as software parts. As the motion compensation is the most time consuming step in the decoding process it is crucial to implement it in a fast and efficient way. This paper elaborates the theoretical background and motivation and highlights the main design choices. In the following evaluation a comparison between the hybrid decoder and a pure software decoder is performed. The results show that the design is capable of increasing the decoding frame rate in the range of 60% for 1080p video streams when running at 100 MHz.

1 Introduction

High Efficiency Video Coding (HEVC) [1] is the latest video coding standard by the *Joint Collaborative Team on Video Coding* (JCT-VC) and has been ratified as H.265 in April 2013. It is the direct successor to the famous H.264/*Advanced Video Coding* (AVC) standard and reduces the bit rate by 50% for the same video quality when compared to H.264. For cost and power reasons it is common practice in video decoding to use dedicated hardware accelerators. Dedicated hardware blocks can perform the most expensive parts of the decoding process in a fast and efficient way thereby offering more performance while consuming less power than a pure CPU-based solution. This paper in particular focuses on designing and implementing a hardware accelerator for motion compensation, i.e. an interpolation filter that should substitute the according part in an existing software decoder as it is the most time-consuming part of software decoders.
Similar to its predecessors HEVC allows to exploit temporal and spatial redundancy in video streams by referring to similar regions in previous frames instead of storing all the data explicitly. This technique that is known as *inter-frame prediction* is implemented in HEVC by using so called *motion vectors* that point to such regions in previously decoded frames. These motion vectors can also have a horizontal or vertical shift relative to the target region with an accuracy of 1/4th of a sample for the luma plane and 1/8th of a sample for the chroma planes. In order to successfully decode an inter-frame predicted HEVC video stream these *fractional samples* must be derived from the adjacent full samples by

using an interpolation filter. This process is called *motion compensation* and has been the main task of the discussed master's thesis.
This paper is organized as follows. Section 2 lists related work that focuses on hardware solutions for motion compensation in general as well as for HEVC in particular. In Section 3 the design process is highlighted followed by a discussion of the evaluation in Section 4. Finally, in Section 5 a conclusion is given regarding the results of the thesis.

2 Related Work

As HEVC has only been standardized in April 2013 the amount of related work in general and regarding motion compensation in particular has been very limited. Guo et al. [2] deal with the motion compensation interpolation and propose a resource-efficient ASIC implementation for the FIR interpolation filter as well as an efficient filter engine that is based on splitting a frame into blocks of 4x4 luma samples. An HEVC video-decoder chip for 4K applications has been presented by Tikekar et al. [3]. This chip is capable of processing 249 MPixel/s which is sufficient for real-time decoding of 4K video streams with 30 FPS. However, a huge amount of related work for AVC motion compensation has been available. An efficient memory access solution is discussed by Tsai et al. [4]. By reusing previous pixels via a cache they can decode a 2048x1024 video stream running at 30 FPS in real-time with less than 200 MB/s of memory bandwidth.
While these approaches focused mostly on pure hardware implementations, this work follows a hardware/software codesign approach. By partitioning the task accordingly the advantages of software and hardware can be combined thus getting a maximum of performance.

3 Design

For the design decision several approaches have been analyzed. While the parallel processing of multiple samples has theoretical advantages regarding the throughput, such solutions tend to occupy many logic resources. Furthermore, the memory will probably be a bottleneck for them. Therefore a solution that is capable of filtering one sample per cycle has been chosen with parallel processing of luma and chroma planes.
The final design consists of two similar independent datapaths: one for luma as well as one for chroma. An overview that is valid for both datapaths can be seen in Figure 1. As the interpolation process involves a two-dimensional FIR filter a two-step procedure has been selected that performs first a one-dimensional horizontal interpolation filtering and afterwards another one-dimensional vertical one. Between these steps a buffer is implemented that stores the results of the first filter before they can be processed by the second filter. This is required as almost the complete horizontal filter process must have finished before the vertical filter process can start. As a result the theoretical throughput is reduced to 0.5 samples per cycle. For each luma and chroma two reference blocks can be processed

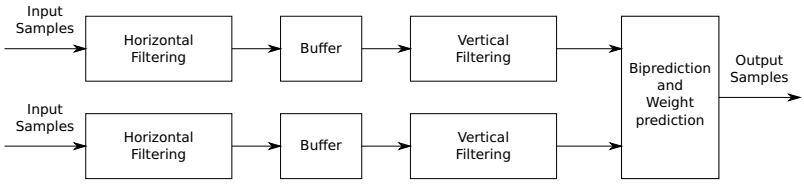


Figure 1: One of the two independent datapaths. Each of them again consists of two sub-datapaths that are required for biprediction. Note that one chroma datapath is sufficient for a subsampling ratio of 4:2:0.

in parallel to support *biprediction*, i.e. interpolating a region in a frame by using two different regions as a reference or input. If biprediction has been selected the results of the two vertical filters will be averaged; otherwise only the result of the first vertical filter will be used. Finally, the result can also be weighted, i.e. be multiplied with a certain factor. This feature is called *weighted prediction* and is implemented by an additional multiplier at the end of each sub-datapath.
For the level of granularity, i.e. the partitioning of the overall work into software and hardware parts of the decoder, the *prediction unit* (PU) has been selected. This is a rectangular block of between 8x4/4x8 and 64x64 luma samples and the according numbers of chroma samples that has a fixed set of parameters. This choice allows to perform most of the complex tasks like parameter evaluation in software while offering the advantage of massive parallelism that hardware solutions provide for the actual interpolation process.

4 Evaluation

The discussed design has been implemented for the Zynq-7020 SoC from Xilinx. A theoretical analysis of the accelerator itself (i.e. only of the motion compensation) yielded an upper bound for the throughput of 50.5 FPS for 1080p video streams when running at 100 MHz. For the software part of the hybrid decoder a scalar software decoder developed at TU Berlin has been modified to use the hardware accelerator for the interpolation process. The interface between hardware and software parts is implemented using a register-based solution in which the CPU handles all the memory access. As the memory overhead is expected to be high, an additional DMA-based interface has been implemented as well to be able to derive the speed-up of such a solution. To be able to compare all three implementations (pure software, register-based implementation, DMA-based implementation) the Kimono video stream of the JCT-VC test sequences [5] has been used in different 1080p encodings. The results when using a frequency of 100 MHz can be seen in Figure 2.
While the frame rate for the register-based interface is reduced significantly by the huge memory overhead, the DMA-based interface is capable of delivering a significant speed-up of about 60% compared to the pure software decoder. However, the memory access still poses the main bottleneck. Figure 2 also shows the luma throughput of the accelerator for PUs of different sizes. For large PUs it converges to the theoretical maximum of 0.5 samples per cycle as the interpolation overhead is decreasing.

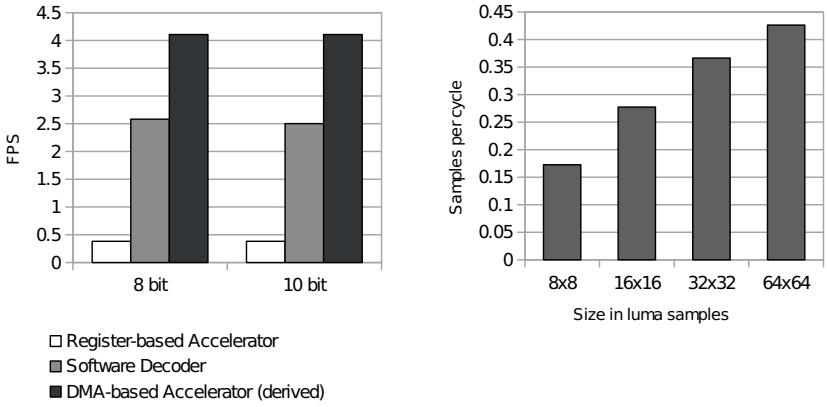


Figure 2: An evaluation of the accelerator. The left diagram shows the achieved frame rates using the evaluation setup. On the right side the luma throughput for PUs of different sizes can be seen.

5 Conclusion

This paper described the design of a hardware-accelerator for HEVC motion compensation. Based on the idea of a hybrid decoder such an accelerator has been implemented. The evaluation proved the feasibility and reasonability of the design as it offers a speed-up of about 60% compared to a pure software solution. Based on the results of this thesis additional work is currently in progress. In particular, further optimizations regarding the memory access will be performed as this turned out to be the major limiting factor in the implementation.

References

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand. Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Transactions on Circuits and System for Video Technology, Volume 22, No. 12:1649-1668, 2012.

[2] Z. Guo, D. Zhou, and S. Goto. An Optimized MC Interpolation Architecture for HEVC. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012.

[3] M. Tikekar, C.-T. Huang, C. Juvekar, V. Sze, and A.P. Chandrakasan. A 249-Mpixel/s HEVC Video-Decoder Chip for 4K Ultra-HD Applications. IEEE Journal of Solid-State Circuits, Volume 49, Issue: 1, 2014.

[4] C.-Y. Tsai, T.-C. Chen, T.-W. Chen, and L.-G. Chen. Bandwidth Optimized Motion Compensation Hardware Design for H.264/AVC HDTV Decoder. 48th Midwest Symposium on Circuits and Systems, 2005.

[5] F. Bossen. Common test conditions and software reference configurations. ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-K1100, 2012.

Design and Implementation of a High-Throughput CABAC Hardware Accelerator for the HEVC Decoder

Philipp Habermann
Embedded Systems Architecture
Technische Universität Berlin

p.habermann@tu-berlin.de

Abstract: HEVC is the new video coding standard of the Joint Collaborative Team on Video Coding. As in its predecessor H.264/AVC, Context-based Adaptive Binary Arithmetic Coding (CABAC) is a throughput bottleneck. This paper presents a hardware acceleration approach for transform coefficient decoding, the most time consuming part of CABAC in HEVC. In addition to a baseline design, a pipelined architecture and a parallel algorithm are implemented in an FPGA to evaluate the gain of these optimizations. The resulting baseline hardware design decodes 62 Mbins/s and achieves a 10× speed-up compared to an optimized software decoder for a typical workload at only a tenth of the processors clock frequency. The pipelined design gives an additional 13.5%, while the parallel design provides a 10% throughput improvement compared to the baseline. According to these results, HEVC CABAC decoding offers good hardware acceleration opportunities that should be further exploited in future work.

1 Introduction

High Efficiency Video Coding (HEVC [1]) is a new video coding standard that targets to halve the bitrate requirements while remaining comparable in perceptive image quality to its predecessor H.264/AVC. HEVC is also designed to better exploit the capabilities of todays multicore architectures. High-level coding tools for sub-picture parallelization were introduced, while low-level throughput limitations were removed to allow more efficient implementations.

A component of HEVC that can only hardly be parallelized is Context-based Adaptive Binary Arithmetic Coding (CABAC). The reason is that there are strong data dependencies which make CABAC a throughput bottleneck. This paper gives a brief overview of a master thesis that aims to evaluate the hardware acceleration opportunities for HEVC CABAC decoding. Therefore, a customized hardware decoder for transform coefficient coding [2] is built, as it is the most time-consuming part of CABAC in HEVC.

The paper is structured as follows. Section 2 presents related work while Section 3 explains the basic functionality of CABAC and describes the design of the hardware accelerator. Section 4 provides a performance evaluation of the resulting implementation. Finally, a conclusion is given in Section 5.

2 Related Work

Sze and Budagavi describe the general throughput improvements in HEVC CABAC compared to H.264/AVC in [3]. The improvements include a reduced total number of bins, a reduced amount of context-coded bins, grouped bins with the same context and grouped bypass-coded bins. Furthermore, there are less context selection dependencies and lower memory requirements.

A comparison of different H.264/AVC CABAC accelerator architectures is provided by Jan and Jozwiak in [4]. According to their analysis, a parallel pipeline approach is most promising and gives the highest throughput. Sze et al. present Parallel CABAC in [5]. This technique allows the decoding of mutiple bins in parallel by using n-ary instead of binary arithmetic coding. This increases the computational complexity and is only suitable for a low number of parallel bins.

The scope of this paper is to evaluate the hardware acceleration opportunities for HEVC CABAC decoding. Therefore, a hardware accelerator is built that covers the transform coefficient coding part while exploiting the throughput improvements introduced in HEVC. The above-mentioned optimizations are also implemented to evaluate the speed-up that can be achieved when using them in HEVC CABAC.

3 Design

Binary arithmetic coding is an efficient entropy coding method, where the encoded bitstream is represented by an offset inside an interval (range). To decode a binary symbol (bin), the range is divided into two subranges. The size of each subrange relative to the initial range is proportional to the probability with that each bin state can appear. The decoded bin state corresponds to the subrange where the offset is located in. This subrange becomes the new range for the decoding of the next bin. Context models are used to estimate the probabilities for bins that represent specific syntax elements. They are updated based on the results of previously decoded bins to provide a good estimation for different inputs. Beside these context-coded bins, there are also bypass-coded bins that have fixed equal probabilities for both bin states. They are coded in a less complex procedure that does not involve context models.

The CABAC decoding process is highly sequential due to strict bin-to-bin dependencies (see Figure 1). The decoding of a bin that is part of a syntax element starts with the contex model selection. The context model is used to perform the actual bin decoding. Depending on the result, the range and offset are updated, the involved context model is adapted and the next syntax element is chosen. As the context model selection for the next bin depends on the selection of the corresponding syntax element, it is not possible to overlap the decoding steps for consecutive bins without modifications.

Based on the analysis of the bin decoding process, three designs are implemented that differ in the amount of resources they use to increase the throughput. First, a baseline design implements the decoding process as seen in Figure 1. The second design implements a

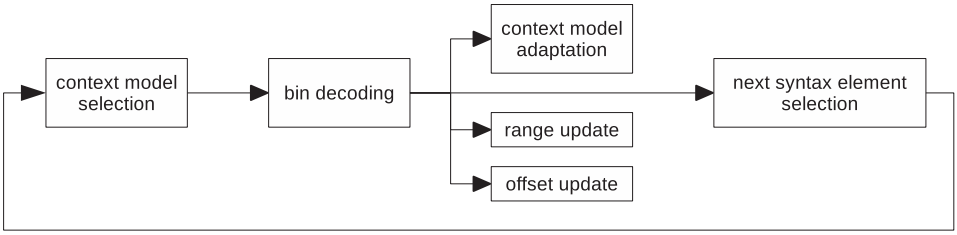


Figure 1: Steps in the decoding process of a context-coded bin. Context model selection and adaptation and the range update are not needed for bypass-coded bins.

two-stage pipeline. Therefore, the selection of the next syntax element is moved to the beginning of the decoding for the next bin. The first pipeline stage performs the syntax element and context model selection. The data path is duplicated to be able to speculate on both possible results of the previously decoded bin. This bin is then forwarded to select the correct context model. The second pipeline stage consists of the bin decoding, the range and offset update and the context model adaptation. Finally, the third design works exactly like the baseline design, except that it uses quaternary arithmetic coding to decode two bypass-coded bins in parallel.

4 Evaluation

All designs are implemented in the programmable logic of the Zynq-7020 System-on-Chip. It contains an ARM Cortex-A9 processor and a Xilinx Artix-7 FPGA, thereby allowing efficient hardware-software co-design. Three different videos are used to represent a wide spectrum of inputs: a small video (low bitrate), an average video (medium bitrate) and a big video (high bitrate). An optimized software decoder from the Embedded Systems Architecture Group at TU Berlin is used as a reference for a throughput comparison (see Figure 2). The speed-up of the baseline design compared to the software decoder is $8\times$ for the big video, $10\times$ for the average one and $14\times$ for the small video. The baseline hardware accelerator operates at a clock frequency of 66 MHz, which is only a tenth of the processors clock frequency. Due to a higher maximum clock frequency of 75 MHz caused by the pipelined architecture, the pipelined design can process 13.5 % more bins per second than the baseline design. The parallel design achieves a speed-up of 8 to 10 % over the baseline due to the simultaneous decoding of two bypass-coded bins.

The resulting hardware accelerator has also been integrated into the software decoder. A low-throughput software register based interface has been used with the main purpose of verifying the functionality of the hardware decoder. Due to the high data transfer overhead, the overall speed-up for this hybrid decoder is only 1.5 %.

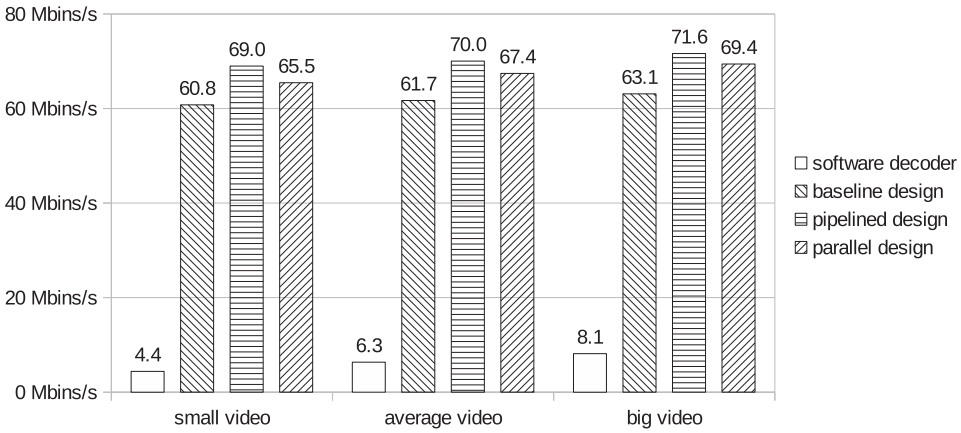


Figure 2: Throughput comparison of the hardware accelerator designs for different input videos.

5 Conclusions

Three different implementations of a hardware accelerator for HEVC transform coefficient decoding were presented. For an average video, a speed-up of 10× over software decoding is achieved at a tenth of the clock frequency, which is a promising result for a highly sequential process like CABAC. When the hardware accelerator is integrated into a software decoder, the theoretical speed-up is reduced due to a very high data transfer overhead caused by the fine-grained hardware acceleration. Currently, a complete CABAC hardware decoder is developed that significantly reduces the data transfer overhead.

References

[1] G. J. Sullivan, J. Ohm, Woo-Jin Han, T. Wiegand, ”Overview of the High Efficiency Video Coding (HEVC) Standard”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, issue 12, pp. 1649 - 1668, December 2012

[2] J. Sole, R. Joshi, Nguyen Nguyen, Tianying Ji, M. Karczewicz, G. Clare, F. Henry, A. Duenas, ”Transform Coefficient Coding in HEVC”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, issue 12, pp. 1765 - 1777, December 2012

[3] V. Sze, M. Budagavi, ”High Throughput CABAC Entropy Coding in HEVC”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, issue 12, pp. 1778 - 1791, December 2012

[4] Y. Jan, L. Jozwiak, ”CABAC Accelerator Architectures for Video Compression in Future Multimedia: A Survey”, Proceedings of 9th International Workshop on Embedded Computer Systems: Architectures, Modeling and Simulation, pp. 24 - 35, July 2009

[5] V. Sze, A. P. Chandrakasan, M. Budagavi, Minhua Zhou, ”Parallel CABAC for low power Video Coding”, Proceedings of 15th IEEE International Conference on Image Processing, pp. 2096 - 2099, October 2008

Low-Cost-Interaktionsgeräte in chirurgischen Anwendungsszenarien: Möglichkeiten und Grenzen

Stanislas Mauser

Hochschule Reutlingen
Fakultät für Informatik
stanislas.mauser@gmail.com

Art der Arbeit: Ausarbeitung im Fach „Interaktive Systeme“
Betreuer der Arbeit: Dr. Kai Holzweißig

Abstract: Die Einführung der Multi-Touch-Technologie hat in vielen Bereichen einen bemerkbar großen und überaus erfolgreichen Durchbruch erzielt. Allerdings gibt es Anwendungsszenarien, in denen Multi-Touch-basierte Technologien an ihre Grenzen stoßen. So beispielsweise in der Chirurgie, wo aus Sterilitätsgründen unsterile Multi-Touch-Geräte nicht eingesetzt werden können. Im Rahmen des vorliegenden Artikels wird vorgeschlagen, neuartige gestenbasierte Interaktionsgeräte wie die „Leap Motion“ in Verbindung mit bestimmten Gestenkonzepten einzusetzen, um solche Szenarien effektiv zu unterstützen.

1 Einleitung

Die Motivation zu dieser Arbeit basiert auf einer Umfrage von Dressler et al. (2011). In dieser Umfrage wurden Chirurgen befragt, ob sie während einer Operation lieber selbst die Informationsbeschaffung aus der Patientenakte vornehmen oder dieses per Anweisung durch eine Schwester erledigen lassen würden. Über 80% der Befragten antworteten, dass sie es vorziehen die Beschaffung selbst vorzunehmen (vgl. [CDT11, S.906]). Es gibt mehrere Gründe, warum der Einsatz von Computertechnologien in chirurgischen Anwendungsszenarien sinnvoll ist. Dies schließt den Zugriff auf die Eingriffsplanungen und -notizen genauso ein wie den unkomplizierten Zugriff auf OP-Geräte, um beispielsweise die Einstellungen einer Lichtquelle des Videoendoskopes vornehmen zu können. Fordert die Aufgabe der Informationsbeschaffung eine zu große mentale Anstrengung bzw. muss hierfür sogar der Standort gewechselt und der sterile Bereich verlassen werden, führt dies zu einem Zeitverlust und implizit zu einem höheren Gesundheitsrisiko für den Patienten (vgl. [CDT11, S.900], [CL09, S.13f]). Im Rahmen der vorliegenden Arbeit wird mit der Leap Motion-Technologie eine vielversprechende Möglichkeit zur gestenbasierten Steuerung im chirurgischen Umfeld untersucht. Dabei wird die berührungslose Technologie prototypisch für die Regulierung einer Lichtquelle eines Videoendoskops umgesetzt. Mittels entsprechender Gestenkonzepte soll dabei der Chirurg optimal unterstützt werden.

2 Bestehende Systeme und Leap Motion

In der menschlichen nonverbalen Kommunikation wird zum Austausch von Informationen auf Gesten zurückgegriffen, welche insbesondere durch Bewegungen von Händen, Armen, des Kopfes oder des ganzen Körpers ausgeführt werden. Im Rahmen der Informatik bildet eine Geste eine nonverbale Information zur Interaktion mit einem digitalen Interaktionsgerät ab. Dabei existiert eine Vielzahl von unterschiedlichen Eingabegeräten. Im Vorfeld wurden eine Reihe von Interaktionsgeräten betrachtet, die für chirurgische Anwendungsszenarien potenziell in Frage kommen könnten. Beginnend bei Maus und Tastatur (vgl. [Boh13]) bis hin zu neuartigen kamerabasierten Systemen wie dem Mi-Report, entwickelt vom Fraunhofer HHI (vgl. [SL09], [CL09]), den Polaris® Systemen von Northern Digital (vgl. [Nor13]) als auch diversen Low-Cost Ansätzen wie der Microsoft Kinect (vgl. [RRA12]) oder dem WiiMote Controller (vgl. [FRC09]). Für die gestenbasierte berührungsfreie Interaktion im praktischen Einsatz sticht vor allem der Leap Motion Controller heraus. Laut Hersteller ist dieser Controller bis zu 200-mal präziser als vergleichbare Technologien. Seine größte Einschränkung ist allerdings der begrenzte Arbeitsbereich von nur ca. 25 bis 600mm rund um das Gerät. Auch wenn dieser Controller nicht perfekt ist und eine Gestensteuerung von Fräsmaschinen nicht anzustreben ist, wäre der Einsatz der Leap Motion für einige Aufgaben und Bereiche durchaus denkbar. Beispiele hierfür könnten die berührungsfreie Steuerung der Maus zur Interaktion mit dem Computer, die Regulierung von Lichtquellen oder die Höhenverstellung von Geräten im OP sowie das Erweitern gewöhnlicher Displays zu berührungslosen Touch-Screens sein.

Der sehr geringe Anschaffungspreis von 90€, die hohe Genauigkeit von bis zu einem 1/100mm für bis zu zwei Hände mit zehn Fingern machen dieses Low-Cost-Gerät für viele Anwendungen interessant. Die Leistungsfähigkeit dieses Controllers wird durch den sehr hohen Sichtwinkel von 150° besonders unterstützt (vgl. [Lea13]).

3 Ansätze von Gestenkonzepten

Es gibt zwar keine offiziellen „Standardgesten“, jedoch lassen sich eine Menge von Gesten in der Literatur wiederfinden. Unter anderem beschreiben Henkens (2011) oder Saffer (2009) sieben Gesten [Hen11, Abb.3.4], die allgemeinen bekannt sind und von den meisten Anwendern intuitiv genutzt werden. (vgl. [Hen11, S.26f], [Saf09, S.179ff]). Mit der Einführung des Apple iPhones in 2007 wurden die heute bekannten Gesten wesentlich geprägt (vgl. [Dor11, S.10]). Bei Touch-Interfaces können Objekte durch die Tap-Geste markiert werden. Bei herkömmlichen Computersystemen kann durch Bewegung der Maus, bei einem berührungsfreien Bedienkonzept auf eine Zeigegeste zurückgegriffen werden. Dabei wird einfach auf die gewünschte Stelle mit dem Finger gezeigt (Point-Geste), um dort hin die Maus zu bewegen oder um dort mit einem Objekt zu interagieren. In Anlehnung an die Standardgesten von Touch-Schnittstellen gibt es fünf Gesten (Point, Grab&Release, Pincer grasp, Wave, Slap) für berührungslose Benutzerschnittstellen (vgl. [Hen11, Abb.3.5], [Hen11, S.26f]). Nicht alle Touch-geprägten Gesten oder die der Zeichensprache lassen sich auf berührungsfreie Schnittstellen übertragen. Abhängig von

der Technik zur berührungsfreien Interaktion kann jedoch eine Vielzahl der von Anwendern bereits bekannte Gesten übertragen werden. Zunächst bietet sich für einen geringen Lernaufwand eine Weiterverwendung im übertragenen Sinne an (vgl. [Dor11, S.12]). Unabhängig von der Interaktionstechnologie ist die Visualisierung von Feedback für den Anwender äußerst bedeutend, besonders die Visualisierung interaktiver Objekte und möglicher Gesten, was aus der Studie von Yee (2009) hervor geht. Weitere hilfreiche Beschreibungen zu Gesten und Studien finden sich zum Mi-Report, dem berührungslos bedienbaren Patientenvisualisierungssystem von Karl Storz (vgl. [CL09]). Der Einfachheit halber orientieren sich die Gesten an den Interaktionskonzepten des Computers. Ein Beispiel hierfür ist die Grundinteraktion der Mausbewegung mittels einer einfachen „Fingerzeig-Geste“. Im Rahmen des Mi-Reports wurden verschiedene Ausführungen für den Mausklick realisiert und praktisch erprobt (vgl. [CL09, S.15ff]).

4 Umsetzung von Gesten mittels Leap Motion-Technologie

Ziel des Prototypen ist es eine berührungsfreie Regulierung einer Lichtquelle von Videoendoskopen oder ähnlichen Geräten mit unterschiedlichen Gesten zu ermöglichen. Aufgrund der äußerst hohen Präzision des Leap Motion-Controllers stehen deutlich mehr Möglichkeiten für Gesten zu Verfügung als beispielsweise mit einer Microsoft Kinect. Eine passende und intuitive Geste für den entsprechenden Anwendungsfall zu finden, ist jedoch nach wie vor ein Problem. Im Prototypen wurden drei Gesten zur Lichtregulierung und eine Geste zur Entsperrung der Bedienung realisiert.

Die sowohl logisch als auch technisch einfachste Geste ist die Anzeige der Helligkeitsstufe durch eine definierte Anzahl von Fingern, analog zur Gebärdensprache. Demnach entspricht eine Faust der Helligkeitsstufe 0, ein Finger entspricht der ersten Helligkeitsstufe (ca. 20%), bis hin zu fünf Finger für eine maximale Helligkeit. Eine seitliche Neigung der Handfläche, analog der Tragfläche eines Flugzeugs, kann bei diesem Gestenansatz das Licht regulieren. Im visuellen Prototypen ist diese Geste auf den horizontalen Schieberegler gemappt. Hält der Chirurg seine Handfläche vollkommen orthogonal über den Controller wird die Lichtquelle mit 50% Helligkeit angesteuert. Durch Neigen der Handfläche nach rechts unten wird das Licht zum Beispiel heller und durch das Neigen nach links unten dunkler. Der dritte Ansatz geht von einem einfachen Hinauf-und-Hinab-Bewegen der Handfläche aus. Bei dieser Geste wird der Abstand zwischen dem Standort der Leap Motion und dem Mittelpunkt der Handfläche gemessen. Es wurden bei dieser Geste 250 Abtaststufen, beginnend ab ca. 15 cm oberhalb des Controllers bis zu ca. 40 cm definiert. Ein vertikaler Schieberegler stellt hierbei die Helligkeit visuell dar. Analog zum Mi-Report wurde für dieses Gestenkonzept auch eine Sperr- und Entsperrgeste implementiert. Damit während der gesamten OP nicht darauf geachtet werden muss, eine der definierten Gesten über dem Controller versehentlich auszuführen, wurde eine sogenannte Bediensperre eingeführt. Zum Entsperren muss der Chirurg seine Handfläche mit ausgestreckten Fingern orthogonal und möglichst gerade für eine definierte Zeit über den Controller halten. Ein angemessenes Feedback für den Chirurgen, das signalisiert, wann die Entsperrgeste erfolgreich war, konnte im Prototyp noch nicht umgesetzt werden. Bei der Neigungsgeste und der horizontalen Geste besteht oft das Problem, dass durch Entfernen der Hand vom Controller ungewollt Werte

eingestellt werden. Die Ursache dafür ist, dass der Controller im Grenzbereich nicht präzis genug arbeitet. Deshalb wurde die Fingeranzahl der interagierenden Hand als zusätzliche Bedingung der Geste hinzugefügt. Wird die Beschränkung beispielsweise auf zwei Finger gestellt, so wird die Lichtquelle nur verändert, wenn an der Handfläche mind. zwei Finger erkannt werden. Wenn die passende Einstellung an der Lichtquelle gefunden ist, kann die Hand als Faust entfernt werden, ohne dass sich die Lichtquelle ungewollt verändert.

5 Fazit und weiterführende Forschungsfragen

Die Entwicklung des Prototypen zur Regulierung einer Endoskopie Lichtquelle ist ein Beispiel für die Präzision und die praktische Eignung der Leap Motion für den medizinischen Einsatz. Der Prototyp zeigt, wie umfangreich die Verbindung der Leap Motion mit an die Anwendungsszenarien angepassten Gesten ist. Der Prototyp verfügt über drei unterschiedliche Gesten, welche alle zur Regulierung der Lichtquelle verwendet werden. Zusätzlich wurde eine Geste zur Sperrung der Gestenerkennung implementiert, um ungewollte Interaktionen zu vermeiden und die Sicherheit für den Patienten zu garantieren. Aufbauend auf den Ergebnissen dieser Arbeit liegt der nächste Schritt in der Evaluation der konzipierten Gesten unter realen Bedingungen. Durch die Evaluation soll herausgefunden werden, welche der Gesten sich speziell für diesen Anwendungsfall anbieten und wie viel Bewegungstoleranz in der Sperrgeste enthalten sein muss, um eine optimale Bedienung durch verschiedene Personen zu erreichen.

6 Literaturverzeichnis

[Boh13] Bohn, S.: Design einer generischen modularen IT-Integrationsarchitektur für die computerassistierte Chirurgie. Dissertation, Universität Leipzig, 2013.

[CDT11] Dressler, C. et al.: Intraoperative Bedienung einer elektronischen Patientenakte durch den Operateur. In: HNO. Vol. 59. Issue 9; Universität Leipzig, 2011, S. 900-907.

[CL09] Chojecki, P.; Leiner, U.: Berührungslose Gestik-Interaktion im Operationssaal. In i-com. Vol. 8. Issue 1; Mensch-Computer-Interaktion im Operationssaal; 2009, S. 13-18.

[Dor11] Dorau, R.: Emotionales Interaktionsdesign, Springer Verlag, Berlin, 2011.

[FRC09] Ritter, F. et al.: Benutzungsschnittstellen für den direkten Zugriff auf 3D-Planungsdaten im OP. In i-com. Vol. 8. Issue 1. MCI im Operationssaal; 2009, S. 24-31.

[Hen11] Henkens, D.: Kombination gestenbasierter Interaktion in multimodalen Anwendungen. Belegarbeit, Technische Universität Dresden, 2010, http://goo.gl/9UTUV8.

[Lea13] Leap Motion Developer Portal; http://developer.leapmotion.com/.

[Nor13] Northern Digital Inc: Polaris Family of Optical Tracking Systems. http://goo.gl/kHvVtC.

[RRA12] Ruppert, G. et al.: Touchless gesture user interface for interactive image visualization in urological surgery; In World Journal of Urology. Vol. 5; 2012, S. 687–691.

[Saf09] Saffer, D.: Designing gestural interfacese. O'Reilly, Sebastopol CA, 2009.

[SL09] Schrader, S.; Leiner, U.: Neuartiges System zur Visualisierung von Patientendaten mit berührungsloser Handgestik-Steuerung. In (Kain, S., et. al. H., Hrsg.): Workshop-Proceedings der Tagung Mensch & Computer; Logos Vlg, Berlin, 2009, S. 302-304.

[Yee09] Yee, W.: Potential Limitations of Multi-touch Gesture Vocabulary: Differentiation, Adoption, Fatigue. In (Hutchison, D. et al. Hrsg.): Human-Computer Interaction. Novel Interaction Methods and Techniques. Springer, Berlin, Heidelberg, 2009; S. 291–300.

Rendering großer Volumendatensätze mit CUDA

Oliver Jato

Hochschule Bonn-Rhein-Sieg
Institut für Visual Computing
Oliver.Jato@h-brs.de

Art der Arbeit: Masterarbeit (Informatik)
Betreuer der Arbeit: Prof. Dr. André Hinkenjann

Abstract: Die Volumenvisualisierung befasst sich mit der Repräsentation, Extraktion, Manipulation und Darstellung von Informationen, denen der Wertebereich dreidimensionaler Funktionen zugrunde liegt. Das Rendering großer Volumendatensätze bezeichnet einen Bereich, in dem die Größe der zu visualisierenden Datensätze ein zusätzliches Problem darstellt. Ziel der hier vorgestellten Arbeit war die Entwicklung eines Volumenrenderers zur Visualisierung von Volumendatensätzen, die aufgrund ihrer Größe nicht vollständig im Speicher der Grafikhardware gehalten werden können. Dazu wurde das Parallel Computing-API CUDA der Firma NVIDIA eingesetzt.

1 Einleitung

Beim Volumenrendering basiert die Darstellung von Informationen, Phänomenen oder Modellen nicht auf der Beschreibung von Oberflächen. Hier sind die Punkte im Raum die Informationsträger. In dieser Arbeit wird direktes Volumenrendering verwendet. Das Volumen wird, ohne die vorherige Approximation von Isooberflächen, in einem sehstrahlbasierten und parallelen Bildraumverfahren direkt abgetastet. Eine Approximation des Volumenrenderingintegrals nach [Max95] wird mit einer vereinfachten Transparenzberechnung nach [BB06] und einer Opazitäskorrektur nach [EHK$^+$06] kombiniert (1). An der Stelle D des Strahls trifft die Strahlungsdichte I ein. I_0 stellt die bereits bei Eintritt in das Volumen vorhandene Strahlungsdichte dar. Der Sehstrahl wird in n Segmente der Länge $\Delta x = \frac{D}{n}$ unterteilt. Die Absorption durch ein Segment ist mit τ gegeben. g liefert das Produkt von Emmission und Absorption für einen Punkt auf einem Sehstrahlsegment i.

$$I(D) \approx I_0 \prod_{i=1}^{n} (1 - \tau(i\Delta x)\Delta x) + \sum_{i=1}^{n} g(i\Delta x)\Delta x \prod_{j=i+1}^{n} (1 - \tau(j\Delta x)\Delta x) \tag{1}$$

Solch ein Ansatz wurde in einem interaktiven Volumenrenderer für kleine Volumendatensätze in [Jat11] realisiert. Abb. 1 wurde auf diese Weise erzeugt.

Nun genügt der Speicher der GPU nicht mehr, um den Datensatz aufzunehmen. Daher ist ein Big Data Problem zu lösen [CE97] und es wird ein Out-of-Core Verfahren realisiert.

Abbildung 1: Volumenrendering der Computertomographie eines Rucksacks. (Datenquelle: [Bar05], zur Verfügung gestellt durch Kevin Kreeger, Viatronix Inc., USA)

2 Verwaltung und Darstellung großer Volumendatensätze

Im Rahmen der vorgestellten Masterarbeit wurden Erweiterungen zur Verwaltung und zum interaktiven Rendering großer Volumendatensätze hinzugefügt. Sie sind konzeptionell in die Bereiche räumliche Unterteilung, Vereinfachung, Paging und Caching einzuordnen.

Bricking: Volumen werden räumlich unterteilt, um diese als Bricks bezeichneten Subvolumen in einem Octree zu organisieren [LHJ99]. Bricks werden im Zusammenhang mit dieser Arbeit auch als Speicherseiten betrachtet.

Branch-on-Need Octree: Um die Anzahl der Blätter zu minimieren, wird ein Branch-on-Need Octree [WVG92] verwendet. Zum Multiresolution-Volumenrendering wird der Octree zu einer hierarchischen Level-of-Detail Struktur erweitert. Daher verweisen alle Knoten auf Bricks. Der Octree dient auch als hierarchische Seitentabelle.

Brick-Cache: Im Speicher der GPU wird eine 3D-Textur als Brick-Cache reserviert und in Blöcke unterteilt. Benötigte Bricks werden in die freien Blöcke dieser Textur übertragen.

Empty-Space-Skipping: Der Transferfunktionsraum wird partitioniert, um vollständig transparente Bricks zu erkennen und diese zur Beschleunigung der Sehstrahlintegration zu überspringen. Die Sichtbarkeit eines Intervalls der Transferfunktion kann durch die Verwendung einer Summed-Area-Table effizient festgestellt werden.

Octreeschnitt: Für jedes Einzelbild muss eine Menge von Bricks gefunden werden, mit der das Volumen vollständig zu visualisieren ist. Dabei sind sowohl die Größe des Brick-Caches als auch die Darstellungsqualität und die Anzahl der je Einzelbild zu übertragenden Bricks zu berücksichtigen. Die Octreeknoten werden anhand der Differenz zwischen ihrem Fehler und dem durchschnittlichen Fehler ihrer sichtbaren Kinder priorisiert. Dies wird im Split-and-Collapse Algorithmus [CF11] genutzt, um einen Schnitt durch den Octree zu adaptieren und den Gesamtfehler zu reduzieren.

Anwendungsgesteuerter Seitenaustausch: Mit dem Octreeschnitt wird ein Pre-Paging Ansatz verfolgt. Das Nachladen von Bricks On-Demand, wie in [CNLE09], würde bei hohen Transparenzen häufige Unterbrechungen des CUDA-Kernels notwendig machen. Die Verwaltung des Brick-Caches wird durch ein unabhängiges Paging-Modul vorgenommen, dessen dedizierter Kernel für die effiziente parallele Aktualisierung der Blöcke sorgt.

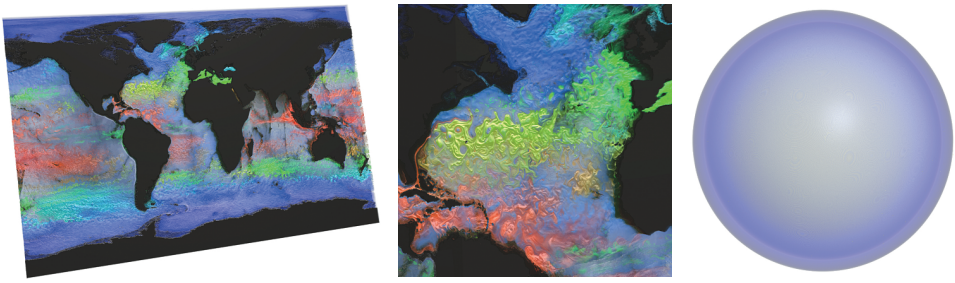
Indexstruktur: Auf der GPU wird eine Indexstruktur benötigt, in welcher die relevanten Informationen der hierarchischen Seitentabelle kodiert sind. Der Rendering-Kernel muss mit dieser Struktur zu einem Abtastpunkt effizient die Adresse des entsprechenden Bricks ermitteln können. Hierfür wird ein Bucket-PR-Octree verwendet, der aus einem Bucket-PR-Quadtree [Sam06] abgeleitet ist. Da die inneren Knoten immer acht Kinder besitzen genügt es von einem Elternknoten jeweils auf den Index des ersten Kindes zu verweisen.

Traversierung: Zur Traversierung der Indexstruktur wird ein Algorithmus ähnlich zu kd-Restart [FS05] verwendet. Dieses erfordert beim Übergang zu einem benachbarten Brick zwar ein erneutes Absteigen von der Wurzel an, da es sich aber um einen sehr kompakten hierarchischen Index handelt können die GPU-Caches effizient genutzt werden.

3 Ergebnisse

Abb. 2 (a) und (b) zeigen die Visualisierung eines Datensatzes des Deutschen Klimarechenzentrums in Hamburg mit 1024^2 Sehstrahlen. Dieser Zeitschritt einer Simulation der globalen Ozeantemperatur besitzt eine Auflösung von $3602 \cdot 2394 \cdot 80$ Voxeln. Der Octree belegt $2,8$ GiB Systemspeicher, der Brick-Cache auf der GPU ist 512 MiB groß. Die Schrittweite der Sehstrahlintegration beträgt $0,49$ Voxellängen und die Berechnung eines Sehstrahls wird durch Early-Ray-Termination bei einer Sättigung über 95% abgebrochen.

Die weiteren Messungen für Abb. 3 wurden mit dem synthetischen Datensatz in Abb. 2 (c) bis zu einer Größe von 2560^3 Voxeln (32 GiB, Octree 45 GiB) auf einer NVIDIA GTX 680 GPU durchgeführt. Abhängig von der Opazität und der Größe des Brick-Caches wurden Einzelbildraten zwischen 3 Hz und 9 Hz erreicht.



(a) (b) (c)

Abbildung 2: (a) und (b) zeigen die semitransparente Visualisierung einer Simulation der globalen Ozeantemperatur, (c) zeigt einen synthetischen Datensatz für Messungen.

Literatur

[Bar05] Dirk Bartz. Volvis. Online: http://www.volvis.org, Tübingen, Deutschland, 2005.

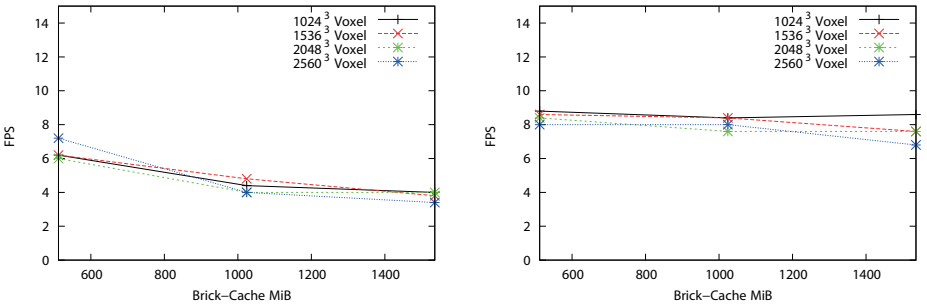


Abbildung 3: Einzelbildraten für unterschiedliche Größen von Volumen und Brick-Cache mit Bricks der Größe 64^3. (a) Semitransparente Visualisierung, (b) vollständige Opazität.

[BB06] Michael Bender und Manfred Brill. *Computergrafik - ein anwendungsorientiertes Lehrbuch*. Hanser Verlag, München, 2. Auflage, 2006.

[CE97] Michael Cox und David Ellsworth. Application-controlled demand paging for out-of-core visualization. In *Proceedings of the 8th conference on Visualization '97*, VIS '97, Seiten 235–ff., Los Alamitos, CA, USA, 1997. IEEE Computer Society Press.

[CF11] Rhadamés Carmona und Bernd Froehlich. Error-controlled real-time cut updates for multi-resolution volume rendering. *Computers & Graphics*, 35(4):931 – 944, 2011.

[CNLE09] Cyril Crassin, Fabrice Neyret, Sylvain Lefebvre und Elmar Eisemann. GigaVoxels: ray-guided streaming for efficient and detailed voxel rendering. In *I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games*, Seiten 15–22, New York, NY, USA, 2009. ACM.

[EHK+06] Klaus Engel, Markus Hadwiger, Joe M. Kniss, Christof Rezk-Salama und Daniel Weiskopf. *Real-time Volume Graphics*. A. K. Peters, Ltd., Natick, MA, USA, 2006.

[FS05] Tim Foley und Jeremy Sugerman. KD-tree acceleration structures for a GPU raytracer. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, HWWS '05, Seiten 15–22, New York, NY, USA, 2005. ACM.

[Jat11] Oliver Jato. Volt: Interaktives Volumenrendering mit CUDA. In *8. Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*, Seiten 73–84, 2011.

[LHJ99] Eric LaMar, Bernd Hamann und Kenneth I. Joy. Multiresolution techniques for interactive texture-based volume visualization. In *Proceedings of the conference on Visualization '99: celebrating ten years*, VIS '99, Seiten 355–361, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[Max95] Nelson Max. Optical Models for Direct Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[Sam06] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, San Francisco, CA, USA, 2006.

[WVG92] Jane Wilhelms und Allen Van Gelder. Octrees for faster isosurface generation. *ACM Trans. Graph.*, 11(3):201–227, Juli 1992.

Entwicklung eines Multiplayer Augmented Reality Spiels mithilfe von Unity und Vuforia

Anja Keicher, Katharina Rakebrand

Hochschule Osnabrück
Fakultät Ingenieurwissenschaften und Informatik
Barbarastr. 16, D-49076 Osnabrück
anja.keicher@hs-osnabrueck.de
katharina.rakebrand@hs-osnabrueck.de

Art der Arbeit: Semesterarbeit (M.Sc. Verteilte u. Mobile Anwendungen)

Betreuer/in der Arbeit: Prof. Dr. Heinz-Josef Eikerling, Hochschule Osnabrück
Prof. Dr. Frank M. Thiesing, Hochschule Osnabrück
(GI-Vertrauensdozent)

Abstract: Die vorliegende Arbeit entstand im Rahmen des Faches Mobile Anwendungen an der Hochschule Osnabrück in Zusammenarbeit mit der Werbeagentur Die Etagen. Sie untersucht die Möglichkeiten zur Entwicklung eines Multiplayer-Augmented Reality (AR)-Spiels unter Verwendung der Game Engine Unity und der Augmented Reality SDK Vuforia. Hierbei werden die grundlegenden Prinzipien der Multiplayerentwicklung vorgestellt und verschiedene Techniken und Protokolle analysiert, um im Anschluss mithilfe einer Beispielimplementierung die am besten geeigneten Varianten für das Aufbauen einer Verbindung zwischen den Spielern zu demonstrieren und eine Empfehlung für zukünftige Entwicklungsprojekte aussprechen zu können.

1 Motivation

Mit der steigenden Verbreitung mobiler Endgeräte wird auch das Thema Augmented Reality zunehmend für Firmen und Nutzer interessanter. Neben der Ergänzung des Kamerabildes durch Informationstexte oder Bilder können die dargestellten Inhalte auch 3D-Objekte umfassen und interaktiv gestaltet sein.

Da es sich bei einem Großteil der beliebtesten Smart-Phone-Applikationen um Spiele handelt [STA1], liegt es nahe diese Nachfrage zu nutzen und Spielkonzepte in ein Augmented Reality-Umfeld zu übertragen. Die Verknüpfung virtueller Komponenten mit einer realen Umgebung kann hierbei genutzt werden, um neuartige Ansätze für Spiele auf mobilen Endgeräten zu entwickeln.

Die vorliegende Arbeit ist in Zusammenarbeit mit der Werbeagentur Die Etagen [ET1] entstanden, die ihren Sitz in Osnabrück hat. Für verschiedene Kunden hat die Agentur bereits Augmented Reality-Inhalte und -Spiele entwickelt. Als Softwarelösungen kamen hierbei die Game Engine Unity [UN1] und für die Augmented Reality-Anbindung das Unity Plugin [VU2] von Vuforia [VU1] zum Einsatz. Die entstandenen Anwendungen sind darauf ausgerichtet, jeweils nur auf einem Endgerät angezeigt und bedient zu werden. Es besteht jedoch der Wunsch, Interaktionen zwischen mehreren Endgeräten zu ermöglichen. Da die Agentur bisher noch keine Erfahrungen mit einer Augmented Reality-Multiplayer-Anwendung gesammelt hat, soll diese Arbeit dazu dienen, verschiedene Realisierungsmöglichkeiten vorzustellen und ihre Vor- und Nachteile zu beleuchten.

2 Implementierung verschiedener Verbindungsvarianten

Um verschiedene Verbindungsvarianten zu testen, wurde eine Spielszene mit Unity erstellt, welche die Spielerobjekte und Augmented Reality-Elemente enthält. Über ein Menü lässt sich auswählen, mit welcher Verbindungsvariante das Spiel gestartet werden soll. Das folgende Kapitel stellt die untersuchten Techniken vor, die anhand beispielhafter Implementierungen getestet und bewertet wurden.

2.1. Direkte Verbindung

Die direkte Verbindung nutzt die in Unity enthaltene Network-Klasse [UN2]. Dadurch ist die Implementierung mit relativ wenig Aufwand möglich. Für den Verbindungsaufbau wird jedoch die IP-Adresse des Gerätes benötigt, welches als Server fungiert und das Spiel bereitstellt. Diese Variante ist daher ausschließlich für die Verbindung von Spielern innerhalb eines lokalen WLAN-Netzes geeignet und bietet nur wenig Bedienkomfort.

2.2. MasterServer

Die Verbindung über einen MasterServer nutzt ebenfalls die von Unity bereitgestellte Network-Klasse. Der MasterServer dient dazu, Clients automatisch über Server zu informieren, mit denen sie sich verbinden können [UN3], wodurch der Verbindungsaufbau komfortabler ist als die direkte Verbdingung. Sofern der Server über eine öffentliche IP-Adresse verfügt, können mit dieser Variante auch Endgeräte, die sich in unterschiedlichen Netzen befinden, verbunden werden. Jedoch ist für die Nutzung des MasterServers immer ein Internetzugang erforderlich.

2.3. AllJoyn

Bei AllJoyn handelt es sich um ein von Qualcomm entwickeltes Framework zur Verbindung verschiedener Geräte in dynamischen Netzwerken. Die AllJoyn SDK ermöglicht einheitliche Kommunikation zwischen verschiedenen Gerätetypen über eine Reihe von Verbindungsmöglichkeiten hinweg [AJ1].

Da AllJoyn durch seine mangelnde Unterstützung für Unity unter iOS und starke Einschränkungen bei der Nutzung von Bluetooth unter Android den Unity-eigenen Lösungen gegenüber für das gegebene Szenario keine Vorteile bietet, wurde darauf verzichtet, es auch in einer Augmented Reality-Anwendung genauer zu untersuchen.

2.4. Bluetooth

Unity selber bietet keine Bluetooth-Unterstützung, jedoch ermöglicht es die Einbindung von Java Code als Plugin. Dadurch können zusätzliche APIs genutzt werden. Java-Methoden, welche die gewünschte Funktionalität umsetzen, können von Unity aus über native C/C++-Interface-Klassen oder direkt aus C#-Skripten aufgerufen werden [UN4]. Im Rahmen dieser Arbeit wurde ein einfaches Java Plugin mit Hilfe der Android Bluetooth API geschrieben, das eine Verbindung zwischen zwei Geräten aufbaut und Signale bei Interaktion mit dem Augmented Reality-Objekt verschickt und empfängt.

3 Bewertung

Die einzelnen Verbindungsvarianten wurden im Hinblick auf zuvor definierte Kriterien bewertet, um so eine Empfehlung für Die Etagen aussprechen zu können. Die folgende Tabelle dient dementsprechend als Orientierung, um bei möglichen Folgeprojekten die Verbindungsvariante zu wählen, die für das jeweilige Einsatzszenario am besten geeignet ist.

	Direkte Verbindung	**MasterServer**	**AllJoyn**	**Bluetooth**
Verbinden der Clients über verschiedene öffentliche Netze	eingeschränkt möglich	möglich	eingeschränkt möglich	nicht möglich
Verbinden der Clients ohne Internet- oder Mobilfunkzugang	möglich	nicht möglich	möglich	möglich
Einfache Bedienung der Anwendung	ausreichend	gut	sehr gut	sehr gut
Implementierungsaufwand	gut	befriedigend	befriedigend	befriedigend

Dokumentation	ausreichend	ausreichend	mangelhaft	befriedigend

Tabelle 1: Bewertung der Verbindungsvarianten

4 Fazit

Obwohl die Multiplayerfähigkeit von Unity häufig erwähnt wird [UN5], findet sich in der entsprechenden Fachliteratur keine Bearbeitung des Themas, insbesondere nicht in Bezug auf die Entwicklung für mobile Endgeräte. Für die Implementierung stehen allein Online-Ressourcen, wie die Unity-Seiten oder private Tutorials als Referenzen zur Verfügung, die jedoch nicht qualitätsgesichert sind.

Unter Berücksichtigung der aufgestellten Kriterien, vor allem was die Bedienbarkeit der Anwendung, sowie die Spontanität und Unabhängigkeit von Zugang zu bestehenden Netzwerken angeht, hat sich Bluetooth als die für die geplante Anwendung als geeignetste Technik herausgestellt. Allerdings muss hier beachtet werden, dass in diesem Fall ein eigenes Plugin entwickelt werden muss, welches die Verbindung mit allen gewünschten Sicherheitsaspekten von Grund auf aufbaut und verwaltet. Soll eine Portierung der Anwendung nach iOS stattfinden, muss eine separate Lösung entwickelt werden.

Bei den vorgestellten Ergebnissen und Implementationen handelt es sich um Ansätze zur Untersuchung der grundsätzlich vorhandenen Möglichkeiten. Sie sollten als erste Orientierung und Empfehlung für Die Etagen dienen und zur Anwendung in realen Projekten genauer analysiert und verfeinert werden. Insgesamt lässt sich jedoch feststellen, dass das Thema Augmented Reality Multiplayer viel Potential zeigt und sich sicherlich im Rahmen einer umfangreicheren Arbeit weiter vertiefen lässt.

5 Literaturverzeichnis

[AJ1] About AllJoyn, https://www.alljoyn.org/about/

[ET1] Die Etagen , http://www.die-etagen.de/

[STA1] Top 15 Kategorien im App Store, http://de.statista.com/statistik/daten/studie/166976/umfrage/beliebteste-kategorien-im-app-store/

[UN1] Unity 3D, http://unity3d.com/

[UN2] Network, http://docs.unity3d.com/Documentation/ScriptReference/Network.html

[UN3] MasterServer, http://docs.unity3d.com/Documentation/ScriptReference/MasterServer.html

[UN4] Building Plugins for Android, http://docs.unity3d.com/Documentation/Manual/PluginsForAndroid.html

[UN5] Unity Networking, http://unity3d.com/unity/workflow/networking

[VU1] Qualcomm Vuforia, http://www.qualcomm.com/solutions/augmented-reality

[VU2] Vuforia Developer Unity Extension, https://developer.vuforia.com/resources/sdk/unity

Untersuchung und Entwicklung von Algorithmen für das Erkennen und Identifizieren von Münzen

Dennis Ziegenhagen

Hochschule Osnabrück
Fakultät Ingenieurwissenschaften und Informatik
Barbarastr. 16, D-49076 Osnabrück
dennis.ziegenhagen@hs-osnabrueck.de

Art der Arbeit: Semesterarbeit (M.Sc. Verteilte und Mobile Anwendungen)
Betreuer der Arbeit: Prof. Dr. Karsten Morisse, Hochschule Osnabrück
Prof. Dr. Frank M. Thiesing, Hochschule Osnabrück
(GI-Vertrauensdozent)

Abstract: In dieser Arbeit werden zunächst verfügbare Ansätze zur Objekterkennung untersucht. Insbesondere wird dabei deren Eignung zur Erkennung und Identifizierung von Münzen getestet. Hierfür werden Fotografien verwendet, auf denen unterschiedliche Euromünzen dargestellt sind. Basierend auf den Untersuchungsergebnissen wird eine Münzerkennung entwickelt, welche eine Fotografie auswertet und Informationen wie z. B. die Anzahl und den Gesamtwert der Münzen zurückgibt. Die Münzerkennung wird als Java-Applikation und als Android-App umgesetzt. Für die Bildverarbeitung wird die Open-Source Bibliothek *OpenCV* eingesetzt.

1 Einleitung

Die große Geschwindigkeit, mit der sich die digitale Bildverarbeitung weiter entwickelt, ermöglicht es, die Verarbeitung von Bildmaterial in einer Vielzahl von Anwendungsfällen einzusetzen [Jä12]. Beispiele dafür sind das Zählen und Vermessen von Objekten, die Analyse von dynamischen Prozessen (Bewegungen, Wachstumsstudien etc.) und das Identifizieren von Personen und Objekten. Es sind Algorithmen und Standard-Vorgehensweisen entstanden, welche auf spezifische Probleme angewendet werden können. Mit der Open-Source Bibliothek *OpenCV*[1] existiert eine Sammlung solcher Funktionen.

Ein Ziel dieser Arbeit ist das Entwickeln einer Anwendung, welche mit Elementen der Bildverarbeitung das Erkennen und Identifizieren von Münzen auf Fotografien ermöglicht. Dabei wird untersucht, ob verfügbare Ansätze der Objekterkennung ausreichend sind oder eine spezielle Lösung entwickelt werden muss. Die entstehende Anwendung soll fähig sein, qualitativ durchschnittliche Fotos von Digitalkameras oder Smartphone-Kameras

[1] http://opencv.org/

entgegenzunehmen, auszuwerten und als Ergebnis die Anzahl und den Gesamtwert der im Foto enthaltenen Münzen auszugeben. In Form einer Android-App kann die Münzerkennung z. B. verwendet werden, um mit Hilfe der eingebauten Smartphone-Kamera Kleingeld zu fotografieren und dessen Gesamtwert zu ermitteln.

Für das Erkennen von Münzen auf Fotografien sind einige besondere Eigenschaften zu berücksichtigen:

- Die Farbe von fotografierten Münzoberflächen kann, abhängig von der Art und dem Einfallswinkel des Lichts, stark von der tatsächlichen Farbe abweichen
- Die Oberflächenfarbe variiert je nach Abnutzungsgrad und Unreinheiten
- Charakteristika, wie z. B. Ränder mit Einkerbungen, sind je nach Kamerawinkel nicht immer zu erkennen
- Die verwendete Kamera und Fotoeigenschaften (Auflösung, nicht durchgeführter oder falscher Weißabgleich etc.) beeinflussen das Erkennungspotential
- Münzen unterschiedlicher Werte können aus demselben Material bestehen, ähnliche Durchmesser besitzen und/oder das gleiche Rückseitenmotiv aufweisen

2 Untersuchung der Ansätze

Um möglichst brauchbare Untersuchungsergebnisse zu erhalten, werden Testfotografien mit einer gewöhnlichen Smartphone-Kamera erstellt und ohne weitere Bildverbesserungen verwendet. Art und Einfallswinkel des Umgebungslichts werden dabei variiert. Es werden Münzen unterschiedlicher Abnutzung und Verschmutzung verwendet. Jede Fotografie enthält eine zufällige Anzahl verschiedener Münzen, welche entweder die Vorder- oder Rückseite zeigen. Zusätzlich enthalten einige Fotografien weitere Objekte, welche von den Objekterkennungen aussortiert werden sollen.

2.1 Klassifizierung und Trainingsdaten

Eine häufig genutzte Möglichkeit, um bekannte Objekte in Bildern zu erkennen, ist die Bildung von Merkmalklassen (Klassifizierung) durch die Auswertung von Stichproben (Training) [NFHS11, S. 434]. Die OpenCV-Bibliothek enthält verschiedene Algorithmen für die Klassifizierung. Aus Trainingsdaten und Beispielbildern können damit *Classifier* gewonnen werden, in denen die Merkmalinformationen gespeichert werden [OC1]. Mit Hilfe einer entsprechenden Funktion können diese Classifier auf ein Bild angewendet werden, um eine Liste von Bildregionen zu ermitteln, in denen übereinstimmende Merkmale gefunden wurden. Um die Eignung der OpenCV-Klassifizierung für die Münzen-Identifikation zu untersuchen, werden Classifier für unterschiedliche Münzwerte erzeugt. Bei der Anwendung der Classifier zeigt sich jedoch, dass sie zur *Unterscheidung* der Münzwerte in den Testszenarien nicht ausreichen.

2.2 OpenCV-Algorithmen

Die OpenCV-Bibliothek enthält eine Sammlung von Bildverarbeitungs-Algorithmen, von denen sich einige als Hilfsmittel für die Münzen-Erkennung eignen. Besonders relevant sind *HoughCircle* (Erkennen von Kreisen), *findContours* (Konturenerkennung) und *FeatureDetector* (Finden von Bereichen mit bestimmten Eigenschaften).

Die Untersuchung zeigt, dass hierbei bereits ein großes Potential für das Erkennen und Unterscheiden von Münzen vorhanden ist, sofern die entsprechenden Parameter geeignet gewählt werden. Da jedoch die Anforderungen an die auszuwertenden Fotografien so gering wie möglich sein sollen, lassen sich keine allgemeinen Einstellungen und Parameterwerte finden, mit denen alle Testfälle erfolgreich abgedeckt werden.

2.3 Extraktion spezifischer Münzen-Merkmale

In einem weiteren Ansatz werden spezifische Merkmale von Euromünzen verwendet, um eine Identifizierung zu ermöglichen. Insbesondere werden dabei der Durchmesser und Materialeigenschaften berücksichtigt, wie z. B. die Farbe und die Verwendung mehrerer Materialien pro Münze. Anstelle einer automatisierten Klassifizierung durch Trainingsdaten findet daher eine gezielte Auswahl der relevanten Merkmale statt.

In exemplarischen Testszenarios werden Fotografien mit Hilfe einer prototypischen Anwendung auf diese spezifischen Merkmale hin untersucht. Als Ergebnis dieses Ansatzes ergibt sich, dass anhand der ausgewählten Merkmale mindestens eine automatisierte Zuordnung eines Objektes zu einer Münz-Gruppe[2] erzielt werden kann.

3 Entwicklung der Münzerkennung

Die vorangegangenen Untersuchungen ließen erkennen, dass eine spezielle Vorgehensweise für die Münzerkennung notwendig ist. Der entwickelte Prozess zur Extraktion der gewünschten Informationen besteht daher aus einer Hierarchie von verketteten Operationen. Diese Vorgehensweise ist ein Standardverfahren der Bildverarbeitung [Jä12, S. 12].

Das zu analysierende Bildmaterial wird zunächst mit Hilfe von Weißabgleich und Angleichung der RGB-Kanäle aufbereitet. Durch Anwendung von Schwellwert-Operationen werden Objekte vom Hintergrund getrennt und interessante Bereiche ermittelt. Um ungleichmäßiger Beleuchtung entgegenzuwirken, wird dabei ein mehrstufiges Schwellwertverfahen verwendet. Iterativ werden die gefundenen Bereiche optimiert, indem jeweils durch Hough-Transformation [NFHS11, S. 195] kreisförmige Objekte ermittelt werden und eine erneute, angepasste Schwellwert-Operation den Bereich weiter eingrenzt. Als Merkmale werden die Position und der Radius der gefundenen Kreise gespeichert.

[2] Die Münzen werden anhand ihres Materials (Kupfer, Nordisches Gold, mehrere Materialien) gruppiert

Durch das Anwenden jeweils passend skalierter Masken werden die durchschnittlichen RGB-Werte des äußeren Randes und des inneren Bereichs der gefundenen Kreise als weitere Merkmale gespeichert. Die Masken entsprechen der 1-Euro-Münze und der 2-Euro-Münze. Die Abweichung der beiden Durchschnittswerte voneinander wird ebenfalls gespeichert, um in der späteren Bewertungsphase zwischen Münzen aus einem und Münzen aus zwei Materialien unterscheiden zu können.

Aus der Menge der gewonnenen Merkmale werden zunächst Statistiken berechnet, wie z. B. größter/kleinster Radius und die Farbwerte mit der jeweils größten Ähnlichkeit zu den Erwartungswerten der Materialien. Die Merkmale jedes gefundenen Objekts werden mit diesen Statistiken verglichen und damit Wahrscheinlichkeiten für jeden Münzwert ermittelt. Objekte, die keine genügend große Wahrscheinlichkeit für einen Münzwert besitzen, werden aussortiert. Am Ende des Bewertungsprozesses existiert eine Zuordnung des wahrscheinlichsten Münzwertes zu jedem gefundenen Objekt. Die Anzahl der Zuordnungen und ihr (wahrscheinlicher) Gesamtwert werden ausgegeben. Damit der Anwender den Erfolg der Münzerkennung kontrollieren kann, werden die Zuordnungen in die Bereiche des ursprünglichen Bildes entsprechend eingetragen.

4 Fazit

Die in der Arbeit vorgenommenen Untersuchungen der Bildverarbeitungsansätze zeigten, dass für die Identifizierung von Münzen auf Fotografien ein spezieller Erkennungsprozess notwendig ist, welcher spezifische Merkmale der Münzen berücksichtigt. Es wurde daher eine Anwendung entwickelt, welche auf Basis von hierarchisch verketteten Methoden der Bildverarbeitung in der Lage ist, Objekte innerhalb eines Fotos zu erkennen und zugehörige Merkmalinformationen zu ermitteln. Durch statistische Analysen und gewichtete Bewertung der Merkmale wird jedem Objekt ein wahrscheinlicher Münzwert zugeordnet. Der Erfolg der Münzerkennung ist dabei von der Fotoqualität abhängig. Da die Berechnung der Wahrscheinlichkeit über einen Vergleich mit den übrigen Objekten geschieht, ist die Erkennungsrate am höchsten, wenn unterschiedliche Münzwerte auf der Fotografie abgebildet sind.

Literatur

[Jä12] Bernd Jähne. *Digitale Bildverarbeitung und Bildgewinnung*. Springer-Verlag, 7. Auflage, 2012.

[NFHS11] Alfred Nischwitz, Max Fischer, Peter Haberäcker und Gudrun Socher. *Computergrafik und Bildverarbeitung*. Vieweg+Teubner Verlag, 3. Auflage, 2011.

[OC1] Cascade Classifier Training — OpenCV 2.4.8.0 documentation. http://docs.opencv.org/doc/user_guide/ug_traincascade.html, abgerufen am 15.01.2014.

IT Performance Management In Multi-Supplier Environments

Fabian Gampfer

Reutlingen University
Hewlett-Packard
fabian.gampfer@gmail.com

Type of work: Master-Thesis
Supervisors: Armin Roth (Reutlingen University)
Markus Müller (Hewlett-Packard)

Abstract: There is a current trend in the industry toward consuming and integrating IT services from a combination of different internal and external suppliers. Following such an operating model for IT requires new approaches to IT performance management. Therefore, this article describes an outline to a solution architecture which can be used to design and implement future proof performance management for multi-supplier IT environments. Two architectural views are considered: the business architecture and the information systems architecture.

1 Introduction

Today, many organizations take advantage of the opportunity to consume IT services from a variety of different internal and external suppliers. A study done by Forrester finds that 41% of IT decision-makers are focusing on a multi-sourcing approach for IT because of advantages in regards to cost, flexibility and accessing specialized skills ([Mat11]). With the current trend towards cloud-services, which enable companies to consume application or infrastructure services on-demand via internet and to be billed on a pay-per-use basis, this trend is likely going to increase.

Besides all advantages of a multi-supplier business model, it also brings challenges. The most significant one is to ensure that all suppliers work together in order to guarantee end-to-end availability, functionality and responsiveness of IT services. The industry trend *Service Integration and Management (SIAM)* describes approaches overcome this challenge. From an organizational perspective, state of the art SIAM models, such as the HP SIAM enterprise blueprint ([Hew14]), promote the role of a service integrator who assures that different internal and external suppliers are working together properly. As a result, a typical multi-sourced IT environment has the following three organizational layers: (1) *Lines of Business* which consume IT services; (2) A *Service Integrator* which integrates IT services from multiple suppliers; (3) *Service Providers* that deliver IT services.

The emerging discipline *IT performance management* is crucial in such an environment because it needs to be ensured that all providers are working collaboratively and in the interest of the business. In order to achieve this, the methods, metrics, processes and systems used to manage performance need to reflect the characteristics of a multi-sourced environment. However, current practices for IT performance management mostly focus on in-sourced or single-sourced operating models. Therefore, this article investigates the subject and describes a reference architecture outline for IT performance management in multi-supplier environments.

2 Reference Enterprise Architecture

The Open Group Architecture Framework (TOGAF), a well-known and proven methodology to develop enterprise architectures ([Hor11]), is used to structure the presented reference architecture. Two architectural views of TOGAF are considered: *(1) The Business Architecture* describes how processes and metrics can reflect the characteristics of a multi-sourced IT environment. *(2) The Information Systems Architecture* describes how IT systems can organize and provide the information required for performance management.

The Business Architecture is developed based on interviews conducted with multiple industry experts. In a second step, the Information Systems Architecture is derived from the Business Architecture. This article focuses on one key area of the reference architecture, which are the metrics that are used to measure and manage performance.

2.1 Business Architecture

In total, this article proposes that four kinds of metrics are required to measure and manage IT performance in a typical multi-supplier environment (see Figure 1). These metrics and the associated targets need to be agreed on between the involved organizations, i.e. between the lines of business and the integrator or between the integrator and the providers.

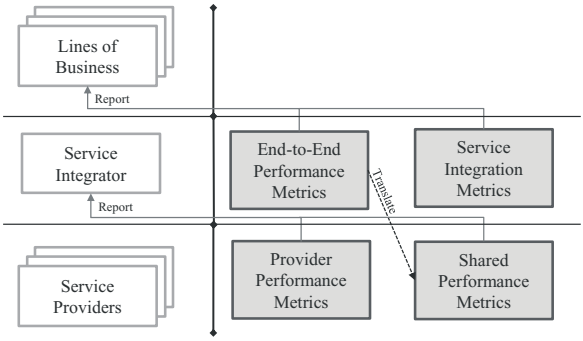


Figure 1: Metrics for Multi-Sourced IT Environments Proposed by this Article

Provider Performance Metrics assess the individual performance of an internal or external service provider. Provider performance metrics need to be agreed on between the service integrator and each provider individually. Example: Percentage of incidents meeting required resolution time for incidents that are solely handled by a certain provider.

End-to-End Performance Metrics assess the end-to-end service delivery including contributions of multiple internal or external providers. End-to-End Performance Metrics need to be agreed on between the integrator and the lines of business. Example: Overall percentage of incidents meeting required resolution time.

Shared Performance Metrics assess the end-to-end performance of a service involving multiple providers. In contrast to provider performance metrics, multiple providers have a shared responsibility – hence the name shared performance metrics ([HHKW11]). Shared Performance Metrics can be used to translate the goal of end-to-end performance to the providers. Example: Percentage of incidents meeting required resolution time for a service delivered by multiple providers.

Service Integration Metrics specifically assess the service integration capability. For a multi-sourced IT environment it is important to know how well providers are integrated because this has a significant impact on the end-to-end performance and also describes how well the service integrator is performing. Example: Percentage of incidents meeting required resolution time for incidents handled by multiple providers.

Provider Performance and End-to-End Performance Metrics are commonly used in organizations today. However, there is a natural gap between these two. The service integrator needs to ensure that both fit together, i.e. that the individual contributions of each provider result in an acceptable end-to-end experience for the lines of business. Shared Performance Metrics and Service Integration Metrics can support organizations and especially service integrators in achieving this.

2.2 Information Systems Architecture

From an information systems perspective this article proposes to implement a typical Business Intelligence (BI) architecture as presented in [KMB10]. The required systems should be operated by the service integrator since he is the link between all involved organizations and therefore has the opportunity to create a holistic, integrated view of all information for performance management. Users from the lines of business, the service integrator and the providers should be able to access these systems in order to retrieve performance information that is relevant for them.

Figure 2 depicts an overview of the Information Systems Architecture proposed by this article. The foundation for the architecture is a central data warehouse (DWH) at the data support layer which extracts data from relevant internal as well as external sources. On top of this DWH a metrics engine calculates results for required metrics and a report engine supports the generation of reports, which show information at a higher level of detail. Access is facilitated through a portal which presents a personalized view for each user.

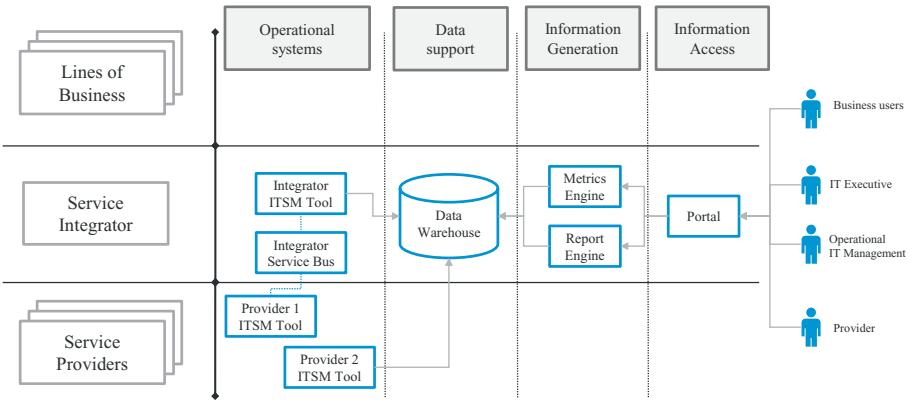


Figure 2: Information Systems Architecture Proposed by this Article

3 Conclusion

The reference architecture proposed by this article can be used to support the implementation of IT performance management in a multi-supplier environment. A prototype of the presented information systems architecture is implemented using the software "IT Executive Scorecard" of Hewlett-Packard. This prototype covers the types of metrics presented in the business architecture previously and therefore shows how IT solutions can support performance management in this context.

In the opinion of the author there will be a growing need for solutions like the one presented in this article. Performance management is a lot more important for multi-supplier environments than for traditional single-sourced environments because additional risks need to be addressed — e.g. the risk related to non-performing suppliers. Organizations which understand this will be able to realize the potential of their multi-sourced operating model more easily.

References

[Hew14] Hewlett-Packard. HP Service Integration and Management, 2014.

[HHKW11] Mark Hakkenberg, Heiner Himmelreich, Hanno Ketterer und Frans Woelders. Shared KPIs in Multivendor IT Outsourcing: Turning "I" to "We", 2011.

[Hor11] Dave Hornford. *TOGAF Version 9.1*. Van Haren Publishing, 1. Auflage, 2011.

[KMB10] Hans-Georg Kemper, Walid Mehanna und Henning Baars. *Business intelligence - Grundlagen und praktische Anwendungen*. Vieweg + Teubner, 3. Auflage, 2010.

[Mat11] Bill Matorelli. Building The Services Integration Layer In Multisourcing, 2011.

Multiagent coordination to improve just in sequence capabilities for multi-tiered supply chains

Marvin Hubl

marvin.hubl@uni-hohenheim.de

Abstract: Just in sequence capabilities feature a growing importance for coordination in multi-tiered supply chains of many industries. To improve coordination in multi-tiered supply chains, auction protocols in multiagent systems with agents acting on behalf of economic entities are applied. This research in progress paper provides a basic model of auctions with software agents operating as bidders and auctioneers. Following the design science approach, this paper describes a research plan in order to develop an auction protocol for multi-tiered supply chains to improve the just in sequence capability. The road pavement industry provides a scenario for the evaluation through simulation studies.

1 Introduction

Just in sequence (JiS) capabilities are highly relevant for practitioners. In the automotive sector JiS is already a widely used strategy and it becomes also important for other industries [WSC11]. JiS is a demand-driven strategy where deliveries are synchronised with manufacturing processes. Hence, it comes up with high demands on coordination in particular for multi-tiered decentralised supply chains. Researcher involved agent based auctions to induce coordination in supply chains [CPF+99, HYH97]. Concreting the coordination problem as resource allocation problem, recent solutions are based on multi-unit combinatorial auctions [KTV13] and multi-attribute combinatorial auctions [WPK13]. This work focuses on the understudied problem of JiS deliveries in multi-tiered supply chains where the deliveries are sensitive in time. Therefore, the question is considered how an auction protocol shall be designed to maintain the JiS deliveries. By grounding the artifact on constructs of the mechanism design theory [Mas08], this paper draws a model of an auction that can be used for implementing and evaluating a protocol for automated auctions.

The proposed auction is a public, non-mediated, double-sided auction with one item characterised by multiple attributes [Bue06]. Agents act on behalf of independent economic entities, maximising the wealth of their principals. Therefore, agents assign a certain utility to each attribute of the auctioned item. In these attributes the JiS capabilities are reflected. The utility of the auctioneer is maximum if the arrival time is exactly as desired under the side condition that other critical attributes do not exceed or fall below a certain level (e.g. a pre-defined price or quality attributes). The auction protocol (mechanism) design needs to be aimed at maximising the social welfare. The protocol constitutes the relation

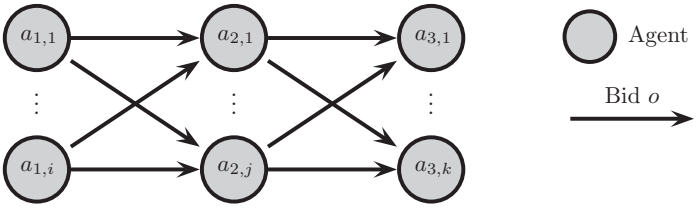


Figure 1: Bid relation in auctions for multi-tiered supply chains

of the agents in terms of sender and receiver of bids (cf. figure 1). Also, the protocol determines temporal conditions for bidding and the mutual relationship between each a bid. Pareto-optimality and incentive compatibility must be satisfied.

2 Auction model

For the usage in multi-tiered supply chains an auction is specified as follows (cf. [Woo09] and [WWW01]):

Let I be the set of auction items.

Let $B \subseteq \wp(I)$ the set of bunches of items (power set of I). Note that for single-item auctions with $|I| = 1 \Rightarrow B = I$ (the empty set is excluded).

Let C be the set of attributes (characteristics).

Let Y_c be the value range (co-domain) of the attribute $c \in C$ and $Y = \bigcup_{\forall c \in C} Y_c$.

Let $A = \{a_1, ..., a_n\}$ be the set of agents operating on different tiers of the supply chain.

Let $\tau : A \to \mathbb{N}$ be the mapping of an agent to a supply chain tier. For an agent denoted as $a_{j,i}$ holds $\tau(a_i) = j$ (cf. fig. 1).

Let $\boldsymbol{\mu} : B \to C \times C \times \cdots \times C$ be the mapping of a bunch to attributes.

Let $\nu : C \to Y$ be the mapping of an attribute to a value.

Let $\boldsymbol{\nu} : C \times C \times \cdots \times C \to Y \times Y \times \cdots \times Y$ be the mapping of multiple attributes to each a value $(\boldsymbol{\nu}(\mathbf{c}) = \mathbf{y} \Rightarrow y_i = \nu(c_i))$.

Let $Z \subseteq B^n$ be the set of all allocations over the bunches and agents. $\mathbf{z} \in Z$ is an allocation of bunches, where bunch z_i is allocated to the agent a_i.

Let $u : B \times A \to \mathbb{R}$ the function which determines the utility of a bunch for an agent.

Let $sw(z_1, \ldots, z_n) = \sum_{i=1}^{n} u(z_i, a_i)$ be the so called social welfare function.

The allocation mechanism aims to maximise the social welfare:
$(z_1^*, \ldots, z_n^*) = \underset{(z_1,\ldots,z_n)\in Z}{\arg\max}\ sw(z_1, \ldots, z_n)$.

In order to specify what is expected from the auction protocol, let

$$o = (b, \mathbf{c}, \mathbf{y}, t, a_1, a_2) \in O \subseteq B \times \boldsymbol{\mu} \times \boldsymbol{\nu} \times T \times A \times A$$

be an actual bid. The first three parameters are as explained above. t is a point of time in T. a_1 is the agent, that sends a bid message and a_2 is the agent that shall receive it. Let furthermore $O_{past}(t)$ be the set of all past offers at time t. Now, a characteristic function $\xi : O \times O_{past} \rightarrow \{0, 1\}$ can be constructed that decides whether a bid is valid or not. In addition, a rule for termination must be specified.

3 Research method

This paper describes a research plan along the guidelines for design science in the information systems research by Hevner et al. [HMPR04]. The following steps are conducted:

1. *Design as artifact:* The artifact is an auction protocol (artifact type: method).
2. *Problem relevance:* JiS is highly relevant for practitioners. In the automotive sector JiS is already established to avoid stocks and to achieve flexibility. The concept promises advantages for other industries, too.
3. *Design evaluation:* The evaluation will be done in parts analytically and – due to the high complexity – simulation studies will be conducted. The dependent variables will be the amount of wasted resources and the number of stagnations.
4. *Research contribution:* The research contribution is the artifact as stated above and knowledge about its usefulness.
5. *Research rigor:* Rigor is achieved by focusing on the knowledge base of the mechanism design theory. It also informs the evaluation using game theoretical properties, such as Pareto-efficiency, incentive compatibility and Nash-equilibrium.
6. *Design as search process:* The aim of the search process is the detection of appropriate means to reach desired ends. Since the ends have already been stated, the search for the corresponding means is still in progress.
7. *Communication of research:* The results are documented in a master thesis and communicated in outlets of the design science community.

4 The pavement construction example for evaluation

For conducting the simulation study, a pavement construction example is applied. Supply chains for the pavement construction are characterised by numerous independent economic entities like asphalt batch plants, logistics service providers and prime contractors that operate a fleet of construction machines like finishers and compactors. The problem that a construction manager faces on site is to order asphalt in a way that neither the finisher runs out of material nor the asphalt cools off too much. Both leads to road irregularities and thus to high costs for retouching work. The proposed auction mechanism is employed in the procurement and transportation of asphalt. The software agents representing the finishers act as the auctioneers, the agents representing the asphalt batch plants and the logistics service providers act as bidders.

References

[Bue06] Ricardo Buettner. A classification structure for automated negotiations. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 523–530, Hong Kong, 2006.

[CPF+99] Ye Chen, Yun Peng, Tim Finin, Yannis Labrou, Scott Cost, Bill Chu, Jian Yao, Rongming Sun, and Bob Wilhelm. A negotiation-based multi-agent system for supply chain management. In *Agent-Based Decision-Support for Managing the Internet-Enabled Supply-Chain*, pages 15–20, Seattle, 1999.

[HMPR04] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.

[HYH97] Kap Hwan Kim, Jun Yeob Song, and Ki Hong Wang. A negotiation based scheduling for items with flexible process plans. *Computers & Industrial Engineering*, 33(3-4):785–788, December 1997.

[KTV13] Piotr Krysta, Orestis Telelis, and Carmine Ventre. Mechanisms for multi-unit combinatorial auctions with a few distinct goods. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, pages 691–698, Saint Paul, Minnesota, USA, 2013.

[Mas08] Eric S. Maskin. Mechanism design: How to implement social goals. *The American Economic Review*, 98(3):567–576, 2008.

[Woo09] Michael Wooldridge. *An introduction to MultiAgent Systems*. Wiley, Chichester, West Sussex, 2 edition, 2009.

[WPK13] Tobias Widmer, Marc Premm, and Paul Karaenke. Energy-aware service allocation for cloud computing. In *Proceedings of the 11th International Conference on Wirtschaftsinformatik (WI 2013)*, pages 1147–1161, Leipzig, 2013.

[WSC11] Stephan M. Wagner and Victor Silveira-Camargos. Decision model for the application of just-in-sequence. *International Journal of Production Research*, 49(19):5713–5736, 2011.

[WWW01] Peter R. Wurman, Michael P. Wellman, and William E. Walsh. A parametrization of the auction design space. *Games and Economic Behavior*, 35:304–338, 2001.

Geschäftsprozessmodellierung durch Spracherkennung und Evaluation geeigneter Satzstrukturen

Tim Maurer

Karlsruher Institut für Technologie (KIT)
Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB)
Tim.Maurer@rolandberger.com

Art der Arbeit: Masterarbeit
Betreuer/in der Arbeit: Björn Keuter, Prof. Dr. Andreas Oberweis

Abstract: Eine Verbindung von gesprochener Sprache und Modellierungssprachen zur Erstellung von Geschäftsprozessmodellen bietet neue Wege in der Geschäftsprozessmodellierung. So kann unter Nutzung von Spracherkennung die Modellierung von Prozessen ohne weitere Eingabegeräte wie Tastatur oder Touchscreens erfolgen. Im Rahmen der Masterarbeit wird die Geschäftsprozessmodellierung durch Spracherkennung untersucht und ein Prototyp entworfen. Notwendige Voraussetzung vor der Umsetzung ist die Ermittlung geeigneter Satzstrukturen zur Beschreibung von Petri-Netz-Prozessmodellen, was einen Kernpunkt der Arbeit darstellt. Dazu werden Befragungen sowohl mit Petri-Netz-Experten als auch mit einer größeren heterogenen Personengruppe durchgeführt. Nach Abschluss der Befragungen liegen Satzstrukturen für die Beschreibung grundlegender Prozesselemente vor. Anschließend wird das Geschäftsprozessmodellierungswerkzeug Horus durch eine selbst implementierte Schnittstelle an eine exemplarische Spracherkennungssoftware angebunden.

1 Einführung

Spätestens seitdem das Unternehmen Apple im Oktober 2011 das iPhone 4S und damit gleichzeitig die Applikation Siri vorstellte, ist Spracherkennung in der breiten Öffentlichkeit ein Begriff. Apple warb mit aufwendigen Fernsehwerbespots für diese neue Funktionalität und erreichte damit ein Millionenpublikum. Seither müssen Programme auf dem Smartphone nicht mehr zwangsläufig von Hand gestartet werden, ein Sprachbefehl reicht hierfür aus. Fragen nach dem zukünftigen Wetter beantwortet Siri zuverlässig. Ganze E-Mails oder sonstige Nachrichten können auch per Spracheingabe geschrieben werden. Die Worte werden in das Mikrofon des Smartphones gesprochen und Siri übernimmt die Transformation der Sprache in einen Text.

Das Konzept der Spracherkennung wird innerhalb der Masterarbeit auf die Modellierung von Geschäftsprozessen übertragen. Bisher werden Prozessmodelle meist manuell auf

dem Papier oder elektronisch mit einem entsprechenden Computerprogramm erstellt. Die Modellierung erfordert Expertise in wenigstens einer Prozessmodellierungssprache. Als nützlich kann sich eine Alternative erweisen, bei der Prozessmodelle ohne Vorkenntnisse durch einfache Spracherkennung erstellt werden.

Am Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB) des Karlsruher Instituts für Technologie (KIT) wird an angrenzenden Themen bereits geforscht. Am Lehrstuhl für Betriebliche Informationssysteme wurden ein Schema und eine Implementierung entwickelt, um geschriebene Texte in Prozessmodelle zu überführen. Auch der Rückweg von Prozessmodellen in Texte ist möglich. Aufgrund ihrer mathematisch fundierten Grundlage eignen sich Petri-Netze besonders als Prozessmodellierungssprache. Sie ermöglichen eine automatische Weiterverarbeitung des Prozessmodells. Auf Basis der Textumwandlung in Petri-Netz-Prozessmodelle wird hier angesetzt. Der eingeschlagene Weg soll weiter verfolgt und erweitert werden, indem die Texte nicht mehr von Hand getippt werden müssen, sondern mittels Spracheingabe über ein Mikrofon erkannt werden können. Diese Idee und entsprechende Vorüberlegungen dazu werden in der Masterarbeit vorgestellt. Der Schwerpunkt liegt dabei auf der Evaluation geeigneter Satzstrukturen und Formulierungen für die sprachliche Beschreibung von Prozessabläufen.

2 Vorgehensweise

Um geeignete Satzstrukturen für die Beschreibung von Prozessmodellen zu finden, wird eine Befragung mittels standardisiertem Fragebogen durchgeführt. Bevor der Fragebogen erstellt werden kann, sind allerdings Vorüberlegungen zu leisten. Im ersten Teil werden die zu beantwortenden Forschungsfragen hergeleitet, indem u. a. die wissenschaftlichen Arbeiten von [FrMP11], [LeMP12] und [Otte11] untersucht werden. Im zweiten Teil werden die Beschriftungen von Petri-Netzen standardisiert, um im weiteren Ablauf Einheitlichkeit zu gewährleisten. Der dritte Teil behandelt die Ermittlung von beschrifteten Petri-Netz-Prozessmodellen, welche innerhalb der Befragung eingesetzt werden sollen. Dazu werden gängige Ablaufstrukturen von Prozessen – diese orientieren sich an den Workflow-Patterns nach [AHKB03] – betrachtet und durch exemplarische Petri-Netz-Prozessmodelle ausgedrückt. Im vierten Teil werden die Prozessmodelle zusammen mit Petri-Netz-Experten durchgesprochen. Innerhalb dieser Leitfadeninterviews sollen die Experten Formulierungen vorgeschlagen, mit denen sie die Abläufe in den Prozessmodellen beschreiben würden. Auf ihren Satzstrukturen basieren die Antwortmöglichkeiten im später entwickelten Fragebogen. Der fünfte Teil handelt von der Konzeption des Fragebogens, mit welchem die Satzstrukturen evaluiert werden sollen. Nach einem Pretest und der Durchführung der Befragung werden die Daten ausgewertet. Zur Auswertung wird ein Algorithmus entwickelt, der die erhobenen Daten systematisch zusammenfasst und aus einzelnen Bestandteilen ganze Sätze kombiniert. Die einzelnen Satzteile mit Stimmenanzahl sind beispielhaft für die Eingabeverbindung in Tabelle 1 zu sehen.

Prä-Stelle	Stelle	Post-Stelle	Prä-Transition	Transition	Post-Transition
Sobald (33)	E-Mail erhalten	wurde, (48)	kann die Aktion (44)	E-Mail lesen	ausgeführt werden. (42)
Nachdem (30)		eingetreten ist, (19)	kann (18)		durchgeführt werden. (27)
Wenn (30)		ist Vorbedingung (15)	wird (18)		stattfinden. (14)
Das Ereignis (15)		erfüllt ist, (13)	für die Aktion (12)		. (13)
Falls (9)		vorliegt, (11)	dann (10)		schalten. (9)
		wird, (11)			

Tabelle 1: Satzteile zur Beschreibung der Eingabeverbindung

3 Ergebnis

Ziel der Studie war es, geeignete Satzstrukturen für die Beschreibung von Petri-Netz-Prozessmodellen zu finden und diese Satzstrukturen innerhalb des Konzepts der Geschäftsprozessmodellierung durch Spracherkennung praktisch anzuwenden. Zunächst wurde eine Befragung mittels standardisiertem Fragebogen unter 164 Personen durchgeführt. Die Befragungsteilnehmer waren zu mehr als drei Vierteln Studierende und besaßen einen wirtschaftswissenschaftlichen Hintergrund.

Durch die Befragung wurden u. a. die folgenden Ergebnisse erzielt. Eingabeverbindungen werden mit der Satzstruktur „Sobald [Stelle] wurde, kann die Aktion [Transition] ausgeführt werden.“ in geeigneter Form beschrieben, Ausgabeverbindungen mit der Satzstruktur „Wenn die Aktion [Transition] ausgeführt wurde, ist [Stelle].“ Die Satzstrukturen der Eingabe- und der Ausgabeverbindung wurden nach einem selbst entwickelten Algorithmus ausgewertet. Zusätzlich wurden die Satzstrukturen zur Beschreibung von Verzweigungselementen (paralleler Split, Synchronisation, alternative Auswahl, einfache Zusammenführung) und weiterer Prozesselemente analysiert. Für die Beschreibung von AND-Elementen ist die Satzstruktur „sowohl [Stelle] als auch [Stelle] als auch …“ geeignet. Für die Beschreibung von XOR-Elementen ist die Satzstruktur „entweder [Transition] oder [Transition] oder …“ geeignet. Mehr als zwei Drittel der Befragungsteilnehmer betrachteten die Geschäftsprozessmodellierung durch Spracherkennung als ein nützliches Konzept. Die Beschriftungen de Petri-Netz-Elemente basieren auf einem selbst entwickelten Standard. Stellen werden demnach mit Substantiv und einem Partizip Perfekt beschriftet, Transitionen mit einem Objekt und einem Verb im Infinitiv. Nach der Auswertung der Befragung wurde die Geschäftsprozessmodellierung durch Spracherkennung umgesetzt, wobei sich die sprachliche Beschreibung der Prozesse auf die zuvor evaluierten Satzstrukturen stützte. Neben der Implementierung der Satzstrukturen in das Modellierungswerkzeug Horus [SVOK11] wurde eine Schaltfläche

programmiert, mit welcher sich die Windows-Spracherkennung unmittelbar innerhalb des Modellierungswerkzeugs ausführen lässt. Darüber hinaus wurde die Spracherkennungssoftware mittels eines Lernprogramms trainiert, sodass das Sprachsignal des Anwenders meist zuverlässig und korrekt erkannt werden konnte.

4 Zusammenfassung und Ausblick

Es wurden systematisch geeignete Satzstrukturen ermittelt, die eine sprachbasierte Prozessmodellierung grundlegend ermöglichen. Diese Satzstrukturen wurden in einem Prototyp angewandt und zeigen auf, dass die Prozessmodellierung durch Spracheingabe ein nutzbares Konzept darstellt. Zur Verbesserung des Konzepts existieren offene Forschungsfragen. Für die Beschriftung von Petri-Netz-Elementen sollte ein allgemeingültiger Standard definiert werden. Dazu könnte untersucht werden, welche Vor- und Nachteile unterschiedliche Beschriftungsformen mit sich bringen.

Die in dieser Arbeit ermittelten Satzstrukturen klingen teilweise monoton. Hier könnten weitere Verbesserungen durchgeführt werden, indem beispielsweise Lexika oder Wörterbüchern eingesetzt werden, um den Numerus innerhalb der Elementbeschriftung zu ermitteln. Damit wäre die Möglichkeit gegeben, unterschiedliche Satzstrukturen für Singular- und Pluralformulierungen zu verwenden. Innerhalb des Modellierungswerkzeugs wurden bisher lediglich die Satzstrukturen implementiert, welche in der Befragung die meisten Stimmen auf sich vereinen konnten. In Zukunft könnten Synonyme implementiert werden, um die Vielseitigkeit der natürlichen Sprache abzubilden. Außerdem könnte das Programm erweitert werden, in dem die eingesprochenen Sätze direkt ohne Betätigen einer Transformations-Schaltfläche in ein Prozessmodell überführt würden.

Literaturverzeichnis

[AHKB03] Aalst, W. M. P. v. d.; Hofstede, A. H. M. t.; Kiepuszewski, B.; Barros, A. P.: Workflow Patterns. Springer, Berlin, 2003.

[FrMP11] Friedrich, F.; Mendling, J.; Puhlmann, F.: Process Model Generation from Natural Language Text. In: Mouratidis, H.; Rolland, C. (Hrsg.): Advanced Information Systems Engineering. Springer, London, Großbritannien, 2011, S. 482-496.

[LeMP12] Leopold, H.; Mendling, J.; Polyvyanyy, A.: Generating Natural Language Texts from Business Process Models. In: Advanced Information Systems Engineering. Springer, Danzig, Polen, 2012, S. 64-79.

[Otte11] Ottensooser, A.; Fekete, A.; Reijers, H. A.; Mendling, J.; Menictas, C.: Making Sense of Business Process Descriptions – An Experimental Comparison of Graphical and Textual Notations. Journal of Systems and Software, ISSN 0164-1212, 2011.

[SVOK11] Schönthaler, F.; Vossen, G.; Oberweis, A.; Karle, T.: Geschäftsprozesse für Business Communities – Modellierungssprachen, Methoden, Werkzeuge. Oldenbourg, München, 2011.

Verbesserung der Lehre durch Frameworking

Jan Czogalla (jan.czogalla@tu-dortmund.de)
Lehrstuhl für künstliche Intelligenz (LS 8)
Fakultät für Informatik, TU Dortmund

Abstract: Studierende lernen am besten, wenn Methoden und ihre Eigenschaften nicht nur in der Vorlesung vorgestellt, sondern auch in eigenen Implementierungen erfahren werden. Komplexe Methoden lassen sich nicht einfach im Rahmen von Übungen zu Vorlesungen implementieren, wenn man den Gesamtzusammenhang vermitteln will. Zur Verbesserung der Lehre ist eine Implementierung in einem vorbereiteten Zusammenhang daher sinnvoll.

Dazu habe ich eine Entwicklungsumgebung umgesetzt, die im Rahmen der Übungen zur Vorlesung „Maschinelles Lernen“ an der TU Dortmund zur Verbesserung der Lehre eingesetzt wird. Sie basiert auf den Werkzeugen Rapidminer und Eclipse, die zusammen eine gute Umgebung für Algorithmen des maschinellen Lernens bereitstellen.

1 Einleitung

Während des Informatikstudiums sollen Studierende neben theoretischen Grundlagen auch Verfahren, Algorithmen und Datenstrukturen aus verschiedensten Bereichen der Informatik kennen und verstehen lernen. Im Rahmen einer Vorlesung soll auch die Implementierung der Algorithmen beleuchtet werden. Oft können Einsichten (theoretisch und architektonisch) nur dadurch erlangt werden, dass wir Studierenden selbst vor die Aufgabe gestellt werden, die gelernten Inhalte umzusetzen. Bei komplexeren Inhalten erreichen diese Aufgabenstellungen aber schnell einen Rahmen, der vor allem an zeitliche Grenzen stößt.

Dieser Artikel befasst sich mit einem Beispiel zur Verbesserung der Lehre durch das Nutzen von bestehenden Werkzeugen und wie dieses Prinzip angenommen wurde.

2 Verbesserung der Lehre

Um Probleme und Eigenarten der Methode selbst zu erfahren, ist es für die Studierenden von Vorteil, zumindest den reinen Algorithmus selbst zu implementieren. Dabei werden allgemeine Konzepte des Software-Designs mit den spezifischen Vorlesungsinhalten verbunden.

Wie oben beschrieben, ist es wünschenswert, Studierende vor die Aufgabe zu stellen, einen Algorithmus, ein Verfahren oder eine Datenstruktur selbst zu implementieren. Um das

Umgesetzte aber auch zu testen, bzw. in Aktion zu erleben, ist meist ein viel größerer Programmieraufwand von Nöten, um alleine die Voraussetzungen für das Kernstück zu schaffen. Im Rahmen einer Übung ist das aber nicht nur für die Studierenden sehr zeitaufwändig, sondern auch für den Korrigierenden, wenn Fehler im Programm nachvollzogen werden müssen. Es ist also wünschenswert, dass die Kernimplementierung im Gesamtzusammenhang geschehen kann, ohne dass ein großer Mehraufwand für die Studierenden entsteht.

Im Folgenden soll anhand eines Beispiels verdeutlicht werden, wie sich ein Übungsverfahren aufbauen lässt, dass sowohl Studierende als auch Lehrende unterstützen kann.

2.1 Maschinelles Lernen an der TU Dortmund

An der TU Dortmund bietet Katharina Morik[1] in der Vorlesung Maschinelles Lernen Übungen mit dem Werkzeug RapidMiner[2] an. RapidMiner bietet neben vielen Data Mining Algorithmen auch mehrere Vorverarbeitungsmethoden und die Möglichkeit, verschiedenste Datenquellen einzulesen und wird durch eine große Palette an Erweiterungen ergänzt. Damit ist im Vergleich zu anderen Data Mining Tools wie KNIME[3] oder WEKA[4] ein größerer Umfang gegeben, zusammen mit einer einfach zu handhabenden GUI, um Data Mining Experimente zusamenstellen zu können.

Die Erweiterung von RapidMiner um einen neuen Operator ist dank eines Plugin-Systems und der zugehörigen Dokumentation einfach umzusetzen, womit sich die Implementierung der Übungen auf das Umsetzen in einen RapidMiner Operator beschränkt, der dann mit den schon vorhandenen Operatoren verknüpft werden kann. Für die Programmierarbeit kommt hierbei das Framework Eclipse[5] zum Einsatz, die IDE, die auch von den RapidMiner Entwicklern genutzt wird.

Damit liegt der Mehraufwand „nur“ noch darin, den Algorithmus in eine solche Operatorumgebung einzubetten. Der nächste Abschnitt befasst sich damit, wie auch das noch minimiert werden kann, damit sich die Studierenden auf die Implementierung der wichtigen Schritte der Lernverfahren konzentrieren können.

2.2 Unterstützung für Studierende

Um RapidMiner zu erweitern, ist es möglich, sogenannte Plugins nach gewissen Konventionen zu erstellen. Für die Vorlesung Maschinelles Lernen wurde ein Plugin für die Übungsaufgaben am Lehrstuhl 8 entwickelt, in dem verschiedene Operatorenrahmen für die einzelnen Übungsaufgaben zur Verfügung gestellt werden. Diese Operatorenrahmen sind so weit vorbereitet, dass lediglich Kenntnisse über die Datenstrukturen der Beispiel-

[1] www-ai.cs.uni-dortmund.de/PERSONAL/morik.html
[2] www.rapidminer.com
[3] www.knime.org
[4] http://www.cs.waikato.ac.nz/ml/weka/
[5] www.eclipse.org

mengen, Metriken und Statistiken von RapidMiner notwendig sind. Das Ganze liegt als Java-Projekt vor, sodass die Studierenden den Code anpassen können, anschließend das Plugin kompilieren und dann nach einem Neustart von RapidMiner ihr eigenes Programm mittels eines Tests nachvollziehen können.

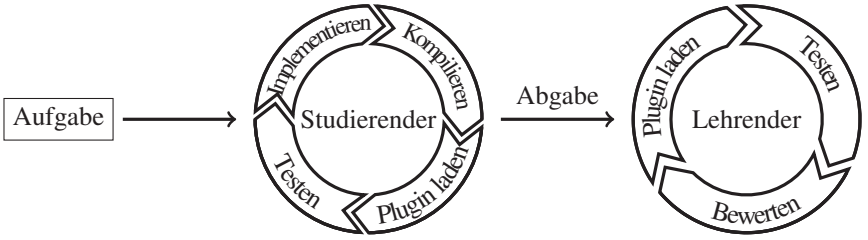


Abbildung 1: Ablauf des Übungsbetriebs

Abbildung 1 zeigt den generellen Ablauf des Übungsbetriebes mit diesem Plugin. Die Studierenden bekommen eine Aufgabe, implementieren diese im vorgegeben Coderahmen und testen ihre Implementierung, bis sie abgabefertig ist. Neben dem kompilierten Plugin wird auch der Sourcecode abgegeben. Der Lehrende kann dann diese Abgaben in RapidMiner laden und bewerten oder Verbesserungsvorschläge abgeben.

Dieses System hat sich im Wintersemester 2011/2012 bewährt, allerdings auch einige Schwachstellen aufgezeigt: Der Neustart von RapidMiner ist sehr zeitaufwändig, aber notwendig, um Änderungen an einem Plugin übernehmen zu können. Da die Studierenden aber sehr oft Änderungen am Code vornehmen, um Fehler zu beseitigen, dauert es oft sehr lange, bis die endgültige Version eines Algorithmus abgabefertig ist. Das führt allerdings zu Frustration, wenn für kleine Änderungen so viel Zeit benötigt wird. Auf der anderen Seite trifft das auch den Lehrenden: Wenn die Studierenden ihre Plugins abgeben, müssen diese jeweils einzeln in das RapidMiner Verzeichnis kopiert werden und zwischen dem Testen zweier Lösungen muss ein Neustart von RapidMiner erfolgen.

2.3 Kernerweiterung für RapidMiner

Um das Verfahren sowohl für Studierende als auch für Lehrende weiter zu vereinfachen, habe ich in Zusammenarbeit mit RapidMiner ein weiteres Plugin entwickelt, das das Neuladen von Plugins zur Laufzeit ermöglicht. Schon nach kurzer Zeit stellte sich heraus, dass dieses Plugin auch generell für Entwickler interessant ist, sodass es auch auf der RCOMM 2013[6] vorgestellt wurde [CM13]. Das Plugin stellt eine Verbindung zwischen Eclipse und RapidMiner her und sorgt so dafür, dass geänderte Plugins mit einem Klick ohne Neustart in RapidMiner geladen werden. Andersherum können Operatoren in RapidMiner per Menü in Eclipse geöffnet werden, um den Code näher zu untersuchen.

Das Entwicklungs-Plugin sorgt dafür, dass zu ersetzende RapidMiner Plugins mitsamt

[6] Jährliche Konferenz der RapidMiner Community

ihrer Operatoren zuerst aus dem laufenden RapidMiner System entfernt werden und anschließend von Grund auf neu hinein geladen werden. Hierbei wird ein neues Feature von Java 7 genutzt, dank dem es möglich ist, alle zu einem RapidMiner Plugin gehörigen Objekte zu verwerfen und aus der selben Datei die neuen Daten herauszulesen.

Um die Verknüpfung zwischen Eclipse und RapidMiner herstellen zu können, war es außerdem nötig, ein Plugin für Eclipse zu schreiben, dass die Verwaltung der zu den Plugins gehörenden Java Projekten übernimmt und RapidMiner anstößt, sobald Änderungen geladen werden müssen. Da Eclipse ebenso wie RapidMiner ein Open Source Projekt ist und eine sehr gut dokumentierte Plugin-API besitzt, hat dies das Verfahren unterstützt und mehr Zeitersparnis beigetragen als eine rein RapidMiner-seitige Erweiterung.

Ein geplantes Feature ist das Neuladen eines RapidMiner Plugins, indem eine Plugin-Datei angegeben wird, die lokal auf dem Rechner liegt oder als URL vorliegen kann. Damit soll es einerseits für den Lehrenden einfacher werden, die verschiedenen Abgaben leichter in RapidMiner laden zu können. Andererseits können so auch Plugins, die nicht öffentlich über den Marketplace verfügbar sein sollen, leicht an einen ganzen Arbeitskreis weitergegeben werden.

3 Fazit

Das Plugin ist im Wintersemester 2013/2014 erfolgreich im Einsatz und wurde, unabhängig von der Vorlesung, im RapidMiner Marketplace bisher über 800 mal heruntergeladen[7]. Der Ablauf der Übungen ist von den Studierenden laut der Lehre-Evaluation gut angenommen worden und hat ihnen die Möglichkeit gegeben, sich auf das Wesentliche zu konzentrieren. Für den Lehrenden wird das Korrigieren der Abgaben auch erleichtert werden, da mit dem nächsten Update das Laden von Plugin Dateien direkt in RapidMiner zur Verfügung stehen wird.

Die eigene Implementierung bekannter und viel genutzter Methoden führt bei Studierenden zu einem tieferen Verständnis der Materie. Um das Zusammenspiel der eigenen Umsetzung mit realen Anwendungsfällen besser erfahren zu können, ist es sinnvoll, sich an bestehenden Frameworks und Werkzeugen zu orientieren. So kann in diesem Fall ein selbst implementierter Lerner in einem RapidMiner Prozess mit den vorhandenen Operatoren kombiniert werden (z.B. Einlesen von Daten, Merkmalsauswahl oder Kreuzvalidierung) und mit bereits vorhandenen Lernern verglichen werden.

Literatur

[CM13] Jan Czogalla and Katharina Morik. Rapid Development of RapidMiner Extensions. In Simon Fischer, Ingo Mierswa, Jõao Mendes Moreira, and Carlos Soares, editors, *Proceedings of the 4th RapidMiner Community Meeting and Conference (RCOMM 2013)*, pages 51–57, 2013.

[7] www.marketplace.rapid-i.com/UpdateServer/faces/product_details.xhtml?productId=rmx_ide2rm

Pytuts.com – Python und NoSQL Tutorials

Jewgeni Kovalev, Robert Mietusch, Joachim Schole
Hochschule Osnabrück
Fakultät Ingenieurwissenschaften und Informatik
jewgeni.kovalev@hs-osnabrueck.de
robert.mietusch@hs-osnabrueck.de
joachim.schole@hs-osnabrueck.de
http://pytuts.com

Art der Arbeit: Hausarbeit zum Modul „Fortgeschrittene Datenbanktechnologien“
Betreuer/in der Arbeit: Prof. Dr.-Ing. Heiko Tapken

Abstract: Im Rahmen der Hausarbeit im Modul „Fortgeschrittene Datenbanktechnologien“ wurde unter Betreuung von Prof. Dr.Ing. Heiko Tapken die Tutorialplattform „Pytuts.com” erstellt. Diese stellt umfangreiche Lernmaterialien zu den Themen Python und (NoSQL)Datenbanken bereit Die entstandenen Tutorials setzen keine vertieften Kenntnisse in den Bereich Pythonprogrammierung und Datenbanken voraus, führen jedoch gleichzeitig fundiert in die Materie ein. Zu jedem betrachteten Datenbankmanagementsystem (SQLite, MongoDB, Neo4j, Postgres, CouchDB und HBase) werden einführende Tutorials bereitgestellt, die dem Leser zu einer Einrichtung und einem ersten Testprogramm verhelfen, sowie weiterführende Tutorials, die die Besonderheiten der Datenbankmanagementsysteme präsentieren und u.a. APIs für Youtube Crawling, Google News und Facebook einbeziehen. Darüberhinaus sind Tutorials zu grundlegenden Funktionen und Besonderheiten von Python entstanden. Die Ergebnisse wurden in Form eines Wikis und eines Blogs auf http://pytuts.com veröffentlicht und dienen als Grundlage für aktuelle und zukünftige Arbeiten.

1. Einführung

Das praxisorientierte Wahlpflichtmodul Fortgeschrittene Datenbanktechnologien vermittelt den Studierenden im Rahmen praktischer Übungen Fähigkeiten zu verschiedenen aktuellen Datenbankmanagementsystemen (DBMS). Neben den relationalen DBMS SQLite (mit dem Fokus mobiler Anwendungen) und PostgreSQL (mit dem Fokus auf linguistische und phonetische Bibliotheken) stehen aktuelle NoSQLDatenbanken im Mittelpunkt. Dabei werden Einblicke in Funktion, Nutzung und Anwendungsgebiete verschiedener Graphdatenbanken (insb. Neo4j), Document Stores (MongoDB), Key/ValueStores und Wide Column Stores (HBase) gegeben. Im Rahmen praktischer Arbeiten werden die Javaschnittstellen der vorgestellten Datenbanken einschl. des MapReduceund Aggregation Frameworks des DBMS MongoDB vertieft betrachtet und anhand realistischer Beispiele (bspw. dem Einlesen eines Dumps der

deutschsprachigen Wikipedia) eingeübt.

Im Rahmen der Hausarbeit zu diesem Modul wurden Tutorials zum Thema Python und Datenbanken erstellt und auf der Website Pytuts.com präsentiert. Dabei wurde untersucht, inwiefern sich Tutorials als didaktisches Mittel eignen. Nach einer selbstständigen Einarbeitung in die Pythonschnittstellen ([@SQL14], [@PYM14], [@POST14], [@COU14], [@NEO14], [@HBA14]) der jeweiligen Datenbanksysteme wurden Anleitungen formuliert, die dem Leser zu einem schnellen Kennenlernen der Datenbanksysteme verhelfen. Die Ergebnisse wurden in einem Wiki festgehalten und in einem Blog veröffentlicht. Darüberhinaus wurde ein YouTube Channel für Videotutorials erstellt. Folgende DBMS wurden betrachtet: SQLite, MongoDB, Neo4j, PostgreSQL, CouchDB und HBase.

2. Entscheidungen zu den Inhalten

2.1 Wahl der Programmiersprache

Um die praktischen Erfahrungen in der Arbeit mit Datenbanken unter einem anderen Blickwinkel zu verfestigen, fiel der Entschluss, eine andere Programmiersprache als Java einzusetzen. Alle betrachteten Datenbanksysteme bieten eine Pythonschnittstelle ([@SQL14], [@PYM14], [@POST14], [@COU14], [@NEO14], [@HBA14]), allerdings mangelt es in vielen Fällen an Lernmaterialien zu der Schnittstellenprogrammierung, weswegen der Entschluss fiel, Tutorials anzufertigen und gesammelt zu veröffentlichen. Als höhere Programmiersprache bietet Python die Möglichkeit, schnell zu ersten Ergebnissen zu kommen, und eignet sich damit gut, um sich in die verschiedenen Datenbankmanagementsysteme einzuarbeiten.

2.2 Wahl der betrachteten Datenbankmanagementsysteme

Um einen möglichst umfassenden Einblick in die Diversität der Datenbanksysteme zu bekommen, wurden sechs verschiedene Datenbanksysteme ausgewählt, worunter zwei relationale (SQLite, PostgreSQL), eine spaltenbasierte (HBase), zwei dokumentorientierte (CouchDB, MongoDB) und eine graphenbasierte (Neo4j) Datenbank zu finden sind. Besondere Aufmerksamkeit verdienten die Datenbanken, die entweder nur knapp oder gar nicht in der Vorlesung behandelt wurden. Darunter fallen Neo4j und CouchDB.

3. Themen der Tutorials

Die weiterführenden Tutorials sollen dem Leser einen tieferen Einblick in das jeweilige Datenbanksystem bieten und ihn auf Besonderheiten und Einsatzgebiete der Datenbanken aufmerksam machen. Darüberhinaus sind einige grundlegende PythonTutorials enstanden: darunter ein Tutorial zum Arbeiten mit URLRessourcen und

zum Parsen von großen XMLDateien. Außerdem wurden Anleitungen angefertigt, die eine schrittweise Installation der Datenbanken und der benötigten Bibliotheken zeigen. Die nachfolgenden kurz skizzierten Tutorials vermitteln einen ersten Eindruck; für die weiteren sei auf die Tutorialplattform verwiesen.

3.1 Python + Neo4j YouTube Crawler Tutorial

In diesem Tutorial wird ein Programm erstellt, das über die YouTube API Schritt für Schritt Videos sucht und in der Graphdatenbank speichert. Als Beziehungen zwischen den Videos dienen die jeweiligen Empfehlungen. In Ergänzung zur schriftlichen Anleitung ist ein Videotutorial entstanden. Dieses ist auf YouTube öffentlich verfügbar.

3.2 Python + PostgreSQL Google News Importieren Tutorial

In diesem Tutorial wird die Nutzung des auf Representational State Transfer (REST)basierende Google Feedzilla API vermittelt. Dabei wird aufgezeigt, wie sich die aktuellsten Google News Artikel automatisiert analysieren lassen.

3.3 Python + SQLite Facebook API

Dieses Tutorial vermittelt die Erstellung eines Python Programms, welches über die Facebook API Daten in eine SQLite Datenbank importiert und auswertet. Am Ende des Tutorials erstellt der Leser ein Programm zur Analyse der Likes von FacebookBeiträgen.

3.4 Python + MongoDB Einführung Tutorial

Mit Hilfe dieses Tutorials bekommt der Leser einen ersten Einblick in die Arbeit mit MongoDB in Python. Es versetzt den Leser in der Lage, eine Verbindung zu MongoDB aufzubauen und wesentliche (CRUD)Operationen auf ihr auszuführen. Außerdem wird ein Programmbeispiel präsentiert, dass die gelernten Materialien zusammenfasst.

4. Ergebnisse

Das wichtigste Ergebnis der Arbeit ist die Tutorialsammlung, die in Form eines Wikis und eines Blogs öffentlich zugänglich ist. In dem Wiki sind sämtliche Tutorials strukturiert gespeichert und können von allen Teammitgliedern aktualisiert und vervollständigt werden. Das Wiki wird fortlaufend aktualisiert, um den stetigen Entwicklungen der DBMS Rechnung zu tragen. Ferner werden weitere Materialien erstellt. Der Blog kündigt jedes neue Tutorial mit einem Eintrag an. Ein YouTube Channel wurde eingerichtet um Videotutorials zur Verfügung zu stellen. Der Einsatz von Tutorials als didaktisches Mittel hat sich bewährt. Es wurde eine Plattform erschaffen, die eine schnelle und unkomplizierte Veröffentlichung von Inhalten zulässt.
Die Arbeit mit der Programmiersprache Python erwies sich als sehr Vorteilhaft. Wie

vorher erwartet, ist die Sprache leicht zu erlernen und dennoch sehr mächtig. Dies macht Python zum optimalen Werkzeug um sich schnell in neue Technologien einzuarbeiten.

Des Weiteren bot die Hausarbeit tiefere Einblicke in die vorgestellten Datenbanksysteme. Besonders fördernd war die selbstständige Arbeit an den DBMS, die nicht in der Vorlesung vorgestellt wurden, da hier intensive Recherche gefordert war.

Das Erstellen von Lernmaterialien bot einen deutlichen Gewinn an didaktischen Fähigkeiten. Der zukünftige Leser musste stets im Hinterkopf behalten werden, was zu einer gründlicheren Arbeit führte.

5. Ausblick

Die Mitglieder arbeiten weiterhin aktiv an der Vervollständigung und Aktualisierung der Tutorials. Um einen breiteren Leserkreis zu erreichen, werden die Lernmaterialien in die englische Sprache übersetzt und so international zugänglich gemacht. Außerdem ist die Erstellung und Veröffentlichung von Videotutorials von wachsender Priorität.

Die Breite der vorgestellten Themen soll zukünftig durch Aufnahme von weiteren Datenbanksystemen wachsen, doch auch bereits vorgestellte Datenbanken werden mit weiteren Tutorials versehen.

Pytuts.com soll zu einer möglichst inhaltsreichen Sammlung von Tutorials zum Thema Python und Datenbanken ausgebaut werden und als Startpunkt zum Kennenlernen von neuen Technologien für den zukünftigen Leser dienen.

Literaturverzeichnis

[@COU14] Python library for working with CouchDB: https://pypi.python.org/pypi/CouchDB, letzter Abruf: 26.01.2014

[@HBA14] HappyBase: http://happybase.readthedocs.org/en/latest/index.html, letzter Abruf: 26.01.2014

[@NEO14] Py2neo: http://book.py2neo.org/en/latest/, letzter Abruf 26.01.2014

[@POST14] PostgreSQL Python tutorial: http://zetcode.com/db/postgresqlpythontutorial/, letzter Abruf: 26.01.2014

[@PYM14] PyMongo 2.6.3 Documentation: http://api.mongodb.org/python/current/, letzter Abruf: 26.01.2014

[@SQL14] DBAPI 2.0 interface for SQLite databases: http://docs.python.org/2/library/sqlite3.html, letzter Abruf: 26.01.2014

Interactive Educational Modules Illustrating Sparse Matrix Computations and their Corresponding Graph Problems

M. Ali Rostami, H. Martin Bücker

{a.rostami,martin.buecker}@uni-jena.de

Abstract: Applications of graph theory are ubiquitous in many different scientific areas. Therefore various software tools are available that aim at explaining graph concepts and graph algorithms. Graphs are particularly important and common in sparse matrix computations. However, there is currently no educational software demonstrating the intimate connection between sparse matrix problems and their corresponding graph problems. The relation between sparse matrix problems and their graph theoretical counterparts is often not easy to catch for students. We describe two interactive educational modules for classroom teaching. The first module explains how to group columns of a sparse matrix in a certain way and shows its connection to graph coloring. The second module illustrates Cholesky factorization of a sparse matrix and clarifies its relation to vertex elimination in some undirected graph. The goal of these modules is to give students the opportunity to interactively explore the underlying phenomena from the point of view of both, linear algebra and graph theory.

1 Introduction

Various areas of scientific computing deeply involve sparse matrix computations. In particular, numerical techniques for the solution of partial differential equations eventually lead to sparse matrices. A matrix is called sparse if it is advantageous to make use of the number and/or position of its nonzero elements. Sophisticated numerical techniques heavily depend on a thorough understanding of the structure of the sparse matrix. The term "structure" typically refers to the pattern of nonzeros which is commonly represented by a graph. Therefore, there is an intimate connection between some algorithm on a matrix and its corresponding algorithm on a graph. For many sparse matrix problems, the understanding of the correspondence between matrix and its graph representation is the key to designing efficient algorithms. Therefore, there is urgent need to clarify this connection in classroom. We argue that a carefully designed visualization of the matrix problem and its corresponding graph problem improves the quality of education in scientific computing.

There are various interactive tools aiming at teaching graph theory. In [SH02], an extensive tool for graph algorithms is presented with an emphasis on algorithm animation. In another tool [CLM96], students select the graph and the algorithm interactively, following the execution of the algorithm step by step. However, to the best of our knowledge, the only educational tool with a clear focus on graphs resulting from scientific computing is EXPLAIN [LLB10, BRL13]. In contrast to other software, this tool shows the intimate

connection between graphs and matrices that arises from the combinatorial aspects in scientific computing.

2 Finding a seed matrix via graph coloring

Given a computer program to evaluate some mathematical function $f : R^n \to R^m$, techniques of automatic differentiation [GW08] generate another computer program capable of evaluating f as well as its $m \times n$ Jacobian matrix J. More precisely, given a user-chosen $n \times q$ matrix S referred as the *seed matrix*, programs generated by automatic differentiation compute the product $J \cdot S \in R^{m \times q}$ without explicitly assembling the Jacobian J. Compared to the program for f, the time and storage requirement of the program for $J \cdot S$ increase by a factor of q, the number of columns of the seed matrix.

If J is sparse with a known sparsity pattern, linearly combining the columns of J can reduce the time and storage to compute all nonzero entries in J. The aim of such a column compression technique is to find a seed matrix with a minimal number of columns, $q \ll n$, and thus reduce the computational effort significantly. The requirement when finding S with a minimal q is that all nonzero elements of J also appear in the product $J \cdot S$. Since a matrix-vector product corresponds to a linear combination of the matrix columns, the above problem boils down to partition the columns of J into a minimal number q of groups such that no two columns in one group have a nonzero element in the same row [GW08].

Coleman and Moré [CM83] reformulated the problem of finding a seed matrix as a coloring problem on the column intersection graph $G = (V, E)$ as follows. The set of vertices is given by $V = \{v_1, v_2, ..., v_n\}$ where a vertex v_i represents the ith column of the Jacobian J. The set of edges is used to encode the condition that no two columns of J that belong to the same group have a nonzero element in the same row. More precisely, there is an edge $(v_i, v_j) \in E$ if and only if the columns represented by v_i and v_j have a nonzero element in a same row position. So, finding a seed matrix with a minimal number of columns is equivalent to finding a coloring of the column intersection graph G with a minimal number of colors.

3 Cholesky factorization via vertex elimination

Systems of linear equations are fundamental building blocks of a large variety of techniques in scientific computing. Often, the coefficient matrix A is symmetric positive definite in which case the Cholesky factorization [Dav06] is the method of choice. In this method, the matrix is decomposed into the product of the form $A = LL^T$ where the Cholesky factor L is lower triangular. The decomposition consists of a sequence of operations on the rows and columns of A which is overwritten by L. If this matrix A is sparse this procedure may generate nonzero elements at matrix positions of L which were zero in A. This phenomenon is called fill-in. The number of fill-in elements depends on the ordering of the rows and columns. A reordering of the rows and columns of A before

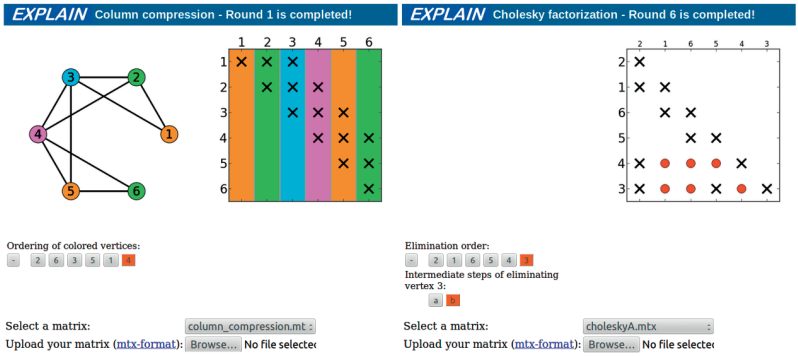


Figure 1: Educational modules for column compression (left) and Cholesky factorization (right).

applying the Cholesky factorization can dramatically decrease the number of fill-in elements. To decrease time and storage requirement, one is therefore interested in finding a reordering of the rows and columns such that the number of fill-in elements is minimized.

Any reordering of the rows and columns of the matrix A corresponds to the class of symmetric permutations of A. This class of matrices is represented by a graph $G = (V, E)$ whose vertices represent rows/columns of A. The edges represent the nonzero elements in A. So, there is an edge between vertex v_i and vertex v_j if and only if there is a nonzero element in A at the row represented by vertex v_i and the column represented by vertex v_j. This graph model dates back to the 1960s [Par61]. Each step of the Cholesky factorization is then nothing but an elimination of a vertex from the graph G. More precisely, a vertex and all its incident edges are removed. At the same time, all the neighbors of this vertex are connected to a clique. If this vertex elimination generates a new edge, this edge corresponds to a fill-in element. Therefore, finding a reordering of the rows and columns that minimizes the number of fill-in elements is equivalent to finding an elimination ordering of the vertices from G that minimizes the number of additionally generated "fill-in edges."

4 EXPLAIN: EXPLoring Algorithms INteractively

An interactive software called *EXPLoring Algorithms INteractively* (EXPLAIN) [LLB10, BRL13] is currently being developed. It is intended for classroom use, not for self study. The modules provide students a visualization of the matrix and the graph side by side. The student can interactively work with the graph and see the corresponding modification in the matrix related to the specific algorithm. In Figure 1, the two modules associated with the two problems discussed in the last two sections are depicted. The software provides an interactive interface offering the student various options for exploring the algorithms such as clicking on a vertex. In the column compression module, a click on a vertex corresponds to the choice of the next column to be colored. In the Cholesky module, a click on a vertex

corresponds to the next vertex to be eliminated. It is also possible to return to previous steps of the algorithms, allowing to change the choice of a vertex. The software supports uploading graphs so that students can explore the algorithms on problem instances that they could choose on their own. The unique feature is that the software allows the student to see the changes in the graph and matrix view simultaneously.

5 Conclusion

The EXPLAIN software is currently being designed as an extensible collection of educational modules specifically designed for combinatorial problems arising from scientific computing. Its implementation is completely web-based. As a result, the effort needed for software testing and maintenance is minimal. Currently, there are two interactive educational modules available: a module to explore the problem of finding a seed matrix in automatic differentiation and another module to analyze the problem of fill-in involved in sparse Cholesky factorization. These modules are easy to access, designed for classroom use, and allow students to better understand the connection between (a) column compression and graph coloring as well as (b) Cholesky factorization and vertex elimination.

References

[BRL13] H. M. Bücker, M. A. Rostami, and M. Lülfesmann. An Interactive Educational Module Illustrating Sparse Matrix Compression via Graph Coloring. In *2013 International Conference on Interactive Collaborative Learning (ICL), Kazan, Russia, September 25–27, 2013*, pages 330–335. IEEE, 2013.

[CLM96] Y. Carbonneaux, J.-M. Laborde, and R. Madani. *CABRI-Graph: A tool for research and teaching in graph theory*, volume 1027 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1996.

[CM83] T. F. Coleman and J. J. Moré. Estimation of Sparse Jacobian Matrices and Graph Coloring Problems. *SIAM Journal on Numerical Analysis*, 20(1):187–209, 1983.

[Dav06] T. A. Davis. *Direct Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2006.

[GW08] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Number 105 in Other Titles in Applied Mathematics. SIAM, Philadelphia, PA, 2nd edition, 2008.

[LLB10] M. Lülfesmann, S. R. Lessenich, and H. M. Bücker. Interactively exploring elimination ordering in symbolic sparse Cholesky factorization. *International Conference on Computational Science, ICCS 2010*, 1:867–874, 2010.

[Par61] S. Parter. The Use of Linear Graphs in Gauss Elimination. *SIAM Review*, 3(2):pp. 119–130, 1961.

[SH02] A. Schliep and W. Hochstättler. Developing Gato and CATBox with Python: Teaching graph algorithms through visualization and experimentation. *Multimedia Tools for Communicating Mathematics*, pages 291–310, 2002.

Contrastive Co-occurrence Analysis on Twitter for the German Election 2013

Uli Fahrer
uli.fahrer@googlemail.com

Supervised by: Prof. Dr. Chris Biemann
Kind of work: Bachelor Thesis

Abstract: This paper describes an approach based on word co-occurrence that contrasts two separate keywords regarding their strongly associated words. This approach is used to investigate how real-world events are reflected in Twitter. Furthermore we present an HTML Dashboard that allows real-time interaction to explore and visualize the data for analyses. Based on a case study about the German election, we perform a contrastive analysis that shows differences and commonalities between two politicians. Results show that the overlap in our analysis is an indicator for key political events. We also found that the Twitter stream reflects real-world events well, and is in high accordance with the daily press.

1 Introduction

The Internet changed over the years in terms of how people interact with each other. Online social networking and microblogging services like Twitter were key factors in this transition. Recent uprisings in the Ukraine or the Arab Spring have shown that Twitter is a tool for sharing information about the protests with the rest of the world. This raises the question of how real-world events are reflected in Twitter. We present an approach based on word co-occurrence that enables us to retrieve words that are strongly associated with a particular keyword and to perform a contrastive analysis on two separate keywords. Besides the investigation of only one keyword, the contrastive analysis can be used to show the differences and commonalities between two keywords and how they are reflected in Twitter. The paper tackles the following research questions:

(1) How do two given keywords differ with respect to their strongly associated words?
(2) How does Twitter reflect real-world events?

We are conducting our research on a case study about the German federal election, which took place on 22nd September 2013.

2 Related Work

Much research has already been done on word association measures in natural language texts. Dunning [Dun93] introduced the log-likelihood measure and Church and Hanks [CH89] the point mutual information measure. Also see Evert and Krenn [EK01] for an overview and an evaluation of co-occurrence measures. Tumasjan et al. [TSSW10] performed a sentiment analysis on the German election 2009 and found that the sentiment of Twitter messages closely corresponds to candidate profiles. Blenn et al. [BCD12] presented a polarization analysis of commonly used words with two keywords based on general Twitter messages. Their system arranges the resulting associated words according to their overall polarity strength. We combine statistical significance measures with this approach to contrast two keywords with respect to their strongly associated words.

3 Data Acquisition and Preprocessing

The data we use in our study was collected from Twitter between August 2, 2013 and October 9, 2013, which includes the German election 2013. We used the Python Tweepy module[1] as an interface for the Twitter Search API[2], where we set the language parameter to German. Further, we defined the 6 parties represented in the German parliament with their top candidates as search terms. Overall, we collected a corpus of 6,163,367 tweets. For the tokenization, we employed the Twitter tokenizer from Owoputi et al. [OOD+13]. We also removed function words, as well as punctuation from the output. In addition we employed the unsupervised POS tagging system from Biemann [Bie06] to annotate the tokens with word classes. Based on a word list consisting of all electable German politicians we also tag words as named entities. To determine the words strongly co-occurring with a given word, we use the log-likelihood measure [Dun93]. We apply this measure to rank the vocabulary according to descending values. For the representation we use a weighted and undirected co-occurrence graph $G(V, E)$, where each vertex $v(\mathrm{id}, \mathrm{freq}, \mathrm{word\_class}) \in V$ is a triple. The triples consist of a unique identifier used to represent the word, the frequency of the word in the corpus and its word class respectively. Each edge $e(w_1, w_2, \mathrm{likelihood}, \mathrm{weight}) \in E$ is a 4-tuple consisting of the two connected words, the significance measure and the edge weight. In addition, we index tweets by words and their co-occurrences to be able to display the reason for the association.

Figure 1: Context: Showing tweets that contain keywords Brüderle, Trittin and #dreikampf, as queried from Figure 2.

[1] http://pythonhosted.org/tweepy/html/
[2] https://dev.twitter.com/docs/api/1/get/search

4 Visualization

We designed an HTML dashboard, which allows real-time user interaction and offers several parameters to affect the generation of the co-occurrence chart:

- **Keywords:** One or two keywords that are used to query the system.
- **Measure:** Variation of the statistical significance measures e.g likelihood.
- **Minimal edge weight:** A threshold on the edge weight that specifies how often the given keyword and its co-occurring words have to occur together in the corpus.
- **Display limit:** Sets the number of displayed words associated with the keywords.
- **Include tweets after date:** Check to include Twitter posts after the election day.
- **Named entities only:** Check to visualize only named entities.
- **Part-of-speech option:** Select word classes that should be included in the chart.

Figure 2 shows a chart for the keywords *Brüderle* and *Trittin* , which are both top candidates for minor parties. We removed user names and excluded tweets after the election day. The left side of the graph shows words only co-occurring with the keyword *Brüderle* and the right side only co-occurring words with *Trittin* . The overlap in the middle indicates words that are co-occurring with both terms. We call this a contrastive analysis. To obtain the context related to the given keyword and a particular co-occurring word the user can select the context view. Figure 1 represents a context for the co-occurring word #dreikampf.

5 Case Study

The fact that Twitter is used as a platform for political deliberation does not necessarily mean that meaningful information can be extracted, or that the distribution of opinion tweets reflects the distribution of opinions in the population. To investigate how Twitter reflects real-world events, we show a contrastive analysis with the keywords *Brüderle* and *Trittin* to examplify the capabilities of our software (see Figure 2). The words in the overlap are related to the clash between both politicians during a TV duel that was being discussed on Twitter under the hashtag #dreikampf. The third politician in the three-way-battle was *Gregor Gysi*, whose name is also found in the overlap. This indicates that the overlap might be a reflection for key political events. In order to compare the results to the daily press, we use the ”Wörter des Tages”[3] platform from the University of Leipzig, which shows terms that are particularly relevant for a day with respect to different daily newspapers. From these terms we identified relevant topics that match our keywords. We found out that about 60% of the co-occurring words related to *Brüderle* are reflected in these topics. For the keyword *Trittin* we even found an overlap of about 70%. As this small example demonstrates, the Twitter stream reflects real-world events well, and is in high accordance with the daily press. We found similar ranges of overlap for other events such as the election, the Stinkefinger affair, the pedophilia discussion, and many more.

[3] http://wortschatz.uni-leipzig.de/wort-des-tages/

6 Conclusion and further research

We presented an approach that visualizes co-occurring words in Twitter messages in an intuitive and flexible way that allows contrastive analysis. With this approach we analyzed over 6 million tweets mentioning parties or politicians in the weeks leading up to the German election 2013. Overall, we found out that the overlap in our contrast analysis gives a sensible reflection of key political events, and that associations characterize single words very well. In addition most of the relevant newspaper topics regarding our contrastive analysis are reflected in Twitter. In our further work, we aim to extend our system to investigate how particular opinions (word-associations) change over time. In addition, it would be a natural extension to also include documents that are linked to from tweets into our corpus.

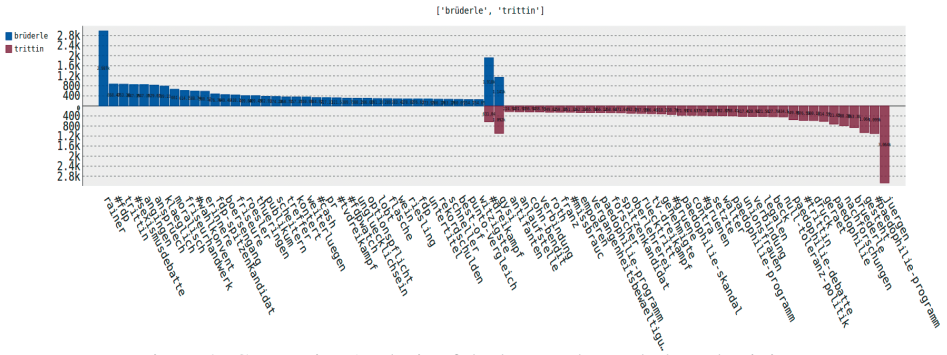


Figure 2: Contrastive Analysis of the keywords Brüderle and Trittin.

References

[BCD12] Norbert Blenn, Kassandra Charalampidou, and Christian Doerr. Context-Sensitive Sentiment Classification of Short Colloquial Text. In *Proc. IFIP'12*, pages 97–108, Prague, Czech Republic, 2012.

[Bie06] Chris Biemann. Unsupervised Part-of-speech Tagging Employing Efficient Graph Clustering. In *Proc. ACL/COLING-2006 SRW*, pages 7–12, Sydney, Australia, 2006.

[CH89] Kenneth W. Church and Patrick Hanks. Word Association Norms, Mutual Information, and Lexicography. In *Proc. ACL-1989*, pages 76–83, Vancouver, Canada, 1989.

[Dun93] Ted Dunning. Accurate Methods for the Statistics of Surprise and Coincidence. *Comp. Ling.*, 19(1):61–74, 1993.

[EK01] Stefan Evert and Brigitte Krenn. Methods for the Qualitative Evaluation of Lexical Association Measures. In *Proc. EACL-2001*, pages 188–195, Toulouse, France, 2001.

[OOD+13] Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT*, pages 380–390, 2013.

[TSSW10] Andranik Tumasjan, Timm Sprenger, Philipp Sandner, and Isabell Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proc. AAAI-2010*, pages 178–185, Atlanta, GA, USA, 2010.

GI-Edition Lecture Notes in Informatics

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 Lan-guage 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 Magenheim (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 Mahni_ (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 – Er-fahrungen 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: Biometrics 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-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing

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.): Persistance, 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, Wofgang 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 Datensicherheit – 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 Adhoc-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-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für inforrmatische Bildung

P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen

P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit

P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and ist Applications

P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005

P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolffried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web

P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik

P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)

P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)

P-69 Robert Hirschfeld, Ryszard Kowalcyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005

P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)

P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005

P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment

P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): “Heute schon das Morgen sehen“

P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology

P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture

P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz

P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006

P-78 K.-O. Wenkel, P. Wagner, M. Morgenstern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel

P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006

P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce

P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS´06

P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006

P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics

P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications

P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems

P-86 Robert Krimmer (Ed.): Electronic Voting 2006

P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik

P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODe 2006, GSEM 2006

P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur

P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006

P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006

P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1

P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2

P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen

P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies

P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices

P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik

P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik

P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten

P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments

P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)

P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme

P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007

P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops

P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications

P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures

P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1

P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2

P-111 Christian Eibl, Johannes Magenheim, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik

P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis

P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1st Conference on Social Semantic Web (CSSW)

P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3rd International Conference on IT-Incident Management & IT-Forensics

P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walther (Eds.) German conference on bioinformatics GCB 2007

P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1st International Working Conference on Business Process and Services Computing BPSC 2007

P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and manegement The 4th International Conference on Grid Service Engineering and Management GSEM 2007

P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007

P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications

P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges

P-121 Korbinian Herrmann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik

P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Pousttchi, Klaus Turowski (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)

P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)
9th Workshop on Parallel Systems and Algorithms (PASA)
Workshop of the GI/ITG Speciel Interest Groups PARS and PARVA

P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)
Unternehmens-IT:
Führungsinstrument oder Verwaltungsbürde
Referate der 28. GIL Jahrestagung

P-126 Rainer Gimnich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)
10th Workshop Software Reengineering (WSR 2008)

P-127 Thomas Kühne, Wolfgang Reisig, Friedrich Steimann (Hrsg.)
Modellierung 2008

P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)
Sicherheit 2008
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
2.-4. April 2008
Saarbrücken, Germany

P-129 Wolfgang Hesse, Andreas Oberweis (Eds.)
Sigsand-Europe 2008
Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems

P-130 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
1. DFN-Forum Kommunikations-technologien Beiträge der Fachtagung

P-131 Robert Krimmer, Rüdiger Grimm (Eds.)
3rd International Conference on Electronic Voting 2008
Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting. CC

P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)
DeLFI 2008:
Die 6. e-Learning Fachtagung Informatik

P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 1

P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 2

P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)
Didaktik der Informatik –
Aktuelle Forschungsergebnisse

P-136 Andreas Beyer, Michael Schroeder (Eds.)
German Conference on Bioinformatics GCB 2008

P-137 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2008: Biometrics and Electronic Signatures

P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)
Synergien durch Integration und Informationslogistik
Proceedings zur DW2008

P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)
Industrialisierung des Software-Managements
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschaftsinformatik

P-140 Oliver Göbel, Sandra Frings, Detlef Günther, Jens Nedon, Dirk Schadt (Eds.)
IMF 2008 - IT Incident Management & IT Forensics

P-141 Peter Loos, Markus Nüttgens, Klaus Turowski, Dirk Werth (Hrsg.)
Modellierung betrieblicher Informations-systeme (MobIS 2008)
Modellierung zwischen SOA und Compliance Management

P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung

P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)
Software Engineering 2009
Fachtagung des GI-Fachbereichs Softwaretechnik

P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)
Datenbanksysteme in Business, Technologie und Web (BTW)

P-145 Knut Hinkelmann, Holger Wache (Eds.)
WM2009: 5th Conference on Professional Knowledge Management

P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)

P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)
Business Process, Services Computing and Intelligent Service Management
BPSC 2009 · ISM 2009 · YRW-MBP 2009

P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)
9th International Conference on Innovative Internet Community Systems
I^{2}CS 2009

P-149 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
2. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.)
Software Engineering
2009 - Workshopband

P-151 Armin Heinzl, Peter Dadam, Stefan Kirn, Peter Lockemann (Eds.)
PRIMIUM
Process Innovation for Enterprise Software

P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 3rd Int'l Workshop EMISA 2009

P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.)
Lernen im Digitalen Zeitalter
DeLFI 2009 – Die 7. E-Learning Fachtagung Informatik

P-154 Stefan Fischer, Erik Maehle Rüdiger Reischuk (Hrsg.)
INFORMATIK 2009
Im Focus das Leben

P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2009:
Biometrics and Electronic Signatures
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-156 Bernhard Koerber (Hrsg.)
Zukunft braucht Herkunft
25 Jahre »INFOS – Informatik und Schule«

P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.)
German Conference on Bioinformatics 2009

P-158 W. Claupein, L. Theuvsen, A. Kämpf, M. Morgenstern (Hrsg.)
Precision Agriculture Reloaded – Informationsgestützte Landwirtschaft

P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.)
Software Engineering 2010

P-160 Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.)
Software Engineering 2010 – Workshopband
(inkl. Doktorandensymposium)

P-161 Gregor Engels, Dimitris Karagiannis Heinrich C. Mayr (Hrsg.)
Modellierung 2010

P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.)
Vernetzte IT für einen effektiven Staat
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2010

P-163 Markus Bick, Stefan Eulgem, Elgar Fleisch, J. Felix Hampe, Birgitta König-Ries, Franz Lehner, Key Pousttchi, Kai Rannenberg (Hrsg.)
Mobile und Ubiquitäre Informationssysteme
Technologien, Anwendungen und Dienste zur Unterstützung von mobiler Kollaboration

P-164 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2010: Biometrics and Electronic Signatures Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-165 Gerald Eichler, Peter Kropf, Ulrike Lechner, Phayung Meesad, Herwig Unger (Eds.)
10th International Conference on Innovative Internet Community Systems (I^2CS) – Jubilee Edition 2010 –

P-166 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
3. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-167 Robert Krimmer, Rüdiger Grimm (Eds.)
4th International Conference on Electronic Voting 2010
co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-168 Ira Diethelm, Christina Dörge, Claudia Hildebrandt, Carsten Schulte (Hrsg.)
Didaktik der Informatik
Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik

P-169 Michael Kerres, Nadine Ojstersek Ulrik Schroeder, Ulrich Hoppe (Hrsg.)
DeLFI 2010 - 8. Tagung der Fachgruppe E-Learning der Gesellschaft für Informatik e.V.

P-170 Felix C. Freiling (Hrsg.)
Sicherheit 2010
Sicherheit, Schutz und Zuverlässigkeit

P-171 Werner Esswein, Klaus Turowski, Martin Juhrisch (Hrsg.)
Modellierung betrieblicher Informationssysteme (MobIS 2010)
Modellgestütztes Management

P-172 Stefan Klink, Agnes Koschmider Marco Mevius, Andreas Oberweis (Hrsg.)
EMISA 2010
Einflussfaktoren auf die Entwicklung flexibler, integrierter Informationssysteme
Beiträge des Workshops der GI-Fachgruppe EMISA (Entwicklungsmethoden für Informationssysteme und deren Anwendung)

P-173 Dietmar Schomburg, Andreas Grote (Eds.)
German Conference on Bioinformatics 2010

P-174 Arslan Brömme, Torsten Eymann, Detlef Hühnlein, Heiko Roßnagel, Paul Schmücker (Hrsg.)
perspeGKtive 2010
Workshop „Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“

P-175 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 1

P-176 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 2

P-177 Witold Abramowicz, Rainer Alt, Klaus-Peter Fähnrich, Bogdan Franczyk, Leszek A. Maciaszek (Eds.)
INFORMATIK 2010
Business Process and Service Science – Proceedings of ISSS and BPSC

P-178 Wolfram Pietsch, Benedikt Krams (Hrsg.)
Vom Projekt zum Produkt
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschafts-informatik (WI-MAW), Aachen, 2010

P-179 Stefan Gruner, Bernhard Rumpe (Eds.)
FM+AM`2010
Second International Workshop on Formal Methods and Agile Methods

P-180 Theo Härder, Wolfgang Lehner, Bernhard Mitschang, Harald Schöning, Holger Schwarz (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW)
14. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme“ (DBIS)

P-181 Michael Clasen, Otto Schätzel, Brigitte Theuvsen (Hrsg.)
Qualität und Effizienz durch informationsgestützte Landwirtschaft, Fokus: Moderne Weinwirtschaft

P-182 Ronald Maier (Hrsg.)
6th Conference on Professional Knowledge Management
From Knowledge to Action

P-183 Ralf Reussner, Matthias Grund, Andreas Oberweis, Walter Tichy (Hrsg.)
Software Engineering 2011
Fachtagung des GI-Fachbereichs Softwaretechnik

P-184 Ralf Reussner, Alexander Pretschner, Stefan Jähnichen (Hrsg.)
Software Engineering 2011
Workshopband
(inkl. Doktorandensymposium)

P-185 Hagen Höpfner, Günther Specht, Thomas Ritz, Christian Bunse (Hrsg.)
MMS 2011: Mobile und ubiquitäre Informationssysteme Proceedings zur 6. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2011)

P-186 Gerald Eichler, Axel Küpper, Volkmar Schau, Hacène Fouchal, Herwig Unger (Eds.)
11th International Conference on Innovative Internet Community Systems (I^2CS)

P-187 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
4. DFN-Forum Kommunikations-technologien, Beiträge der Fachtagung 20. Juni bis 21. Juni 2011 Bonn

P-188 Holger Rohland, Andrea Kienle, Steffen Friedrich (Hrsg.)
DeLFI 2011 – Die 9. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. 5.–8. September 2011, Dresden

P-189 Thomas, Marco (Hrsg.)
Informatik in Bildung und Beruf
INFOS 2011
14. GI-Fachtagung Informatik und Schule

P-190 Markus Nüttgens, Oliver Thomas, Barbara Weber (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2011)

P-191 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2011
International Conference of the Biometrics Special Interest Group

P-192 Hans-Ulrich Heiß, Peter Pepper, Holger Schlingloff, Jörg Schneider (Hrsg.)
INFORMATIK 2011
Informatik schafft Communities

P-193 Wolfgang Lehner, Gunther Piller (Hrsg.)
IMDM 2011

P-194 M. Clasen, G. Fröhlich, H. Bernhardt, K. Hildebrand, B. Theuvsen (Hrsg.)
Informationstechnologie für eine nachhaltige Landbewirtschaftung
Fokus Forstwirtschaft

P-195 Neeraj Suri, Michael Waidner (Hrsg.)
Sicherheit 2012
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 6. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)

P-196 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2012
Proceedings of the 11th International Conference of the Biometrics Special Interest Group

P-197 Jörn von Lucke, Christian P. Geiger, Siegfried Kaiser, Erich Schweighofer, Maria A. Wimmer (Hrsg.)
Auf dem Weg zu einer offenen, smarten und vernetzten Verwaltungskultur
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2012

P-198 Stefan Jähnichen, Axel Küpper, Sahin Albayrak (Hrsg.)
Software Engineering 2012
Fachtagung des GI-Fachbereichs Softwaretechnik

P-199 Stefan Jähnichen, Bernhard Rumpe, Holger Schlingloff (Hrsg.)
Software Engineering 2012
Workshopband

P-200 Gero Mühl, Jan Richling, Andreas Herkersdorf (Hrsg.)
ARCS 2012 Workshops

P-201 Elmar J. Sinz Andy Schürr (Hrsg.)
Modellierung 2012

P-202 Andrea Back, Markus Bick, Martin Breunig, Key Pousttchi, Frédéric Thiesse (Hrsg.)
MMS 2012:Mobile und Ubiquitäre Informationssysteme

P-203 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
5. DFN-Forum Kommunikations-technologien
Beiträge der Fachtagung

P-204 Gerald Eichler, Leendert W. M. Wienhofen, Anders Kofod-Petersen, Herwig Unger (Eds.)
12th International Conference on Innovative Internet Community Systems (I2CS 2012)

P-205 Manuel J. Kripp, Melanie Volkamer, Rüdiger Grimm (Eds.)
5th International Conference on Electronic Voting 2012 (EVOTE2012)
Co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-206 Stefanie Rinderle-Ma, Mathias Weske (Hrsg.)
EMISA 2012
Der Mensch im Zentrum der Modellierung

P-207 Jörg Desel, Jörg M. Haake, Christian Spannagel (Hrsg.)
DeLFI 2012: Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.
24.–26. September 2012

P-208 Ursula Goltz, Marcus Magnor, Hans-Jürgen Appelrath, Herbert Matthies, Wolf-Tilo Balke, Lars Wolf (Hrsg.) INFORMATIK 2012

P-209 Hans Brandt-Pook, André Fleer, Thorsten Spitta, Malte Wattenberg (Hrsg.) Nachhaltiges Software Management

P-210 Erhard Plödereder, Peter Dencker, Herbert Klenk, Hubert B. Keller, Silke Spitzer (Hrsg.) Automotive – Safety & Security 2012 Sicherheit und Zuverlässigkeit für automobile Informationstechnik

P-211 M. Clasen, K. C. Kersebaum, A. Meyer-Aurich, B. Theuvsen (Hrsg.) Massendatenmanagement in der Agrar- und Ernährungswirtschaft Erhebung - Verarbeitung - Nutzung Referate der 33. GIL-Jahrestagung 20. – 21. Februar 2013, Potsdam

P-212 Arslan Brömme, Christoph Busch (Eds.) BIOSIG 2013 Proceedings of the 12th International Conference of the Biometrics Special Interest Group 04.–06. September 2013 Darmstadt, Germany

P-213 Stefan Kowalewski, Bernhard Rumpe (Hrsg.) Software Engineering 2013 Fachtagung des GI-Fachbereichs Softwaretechnik

P-214 Volker Markl, Gunter Saake, Kai-Uwe Sattler, Gregor Hackenbroich, Bernhard Mit schang, Theo Härder, Veit Köppen (Hrsg.) Datenbanksysteme für Business, Technologie und Web (BTW) 2013 13. – 15. März 2013, Magdeburg

P-215 Stefan Wagner, Horst Lichter (Hrsg.) Software Engineering 2013 Workshopband (inkl. Doktorandensymposium) 26. Februar – 1. März 2013, Aachen

P-216 Gunter Saake, Andreas Henrich, Wolfgang Lehner, Thomas Neumann, Veit Köppen (Hrsg.) Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband 11. – 12. März 2013, Magdeburg

P-217 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.) 6. DFN-Forum Kommunikations- technologien Beiträge der Fachtagung 03.–04. Juni 2013, Erlangen

P-218 Andreas Breiter, Christoph Rensing (Hrsg.) DeLFI 2013: Die 11 e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI) 8. – 11. September 2013, Bremen

P-219 Norbert Breier, Peer Stechert, Thomas Wilke (Hrsg.) Informatik erweitert Horizonte INFOS 2013 15. GI-Fachtagung Informatik und Schule 26. – 28. September 2013

P-220 Matthias Horbach (Hrsg.) INFORMATIK 2013 Informatik angepasst an Mensch, Organisation und Umwelt 16. – 20. September 2013, Koblenz

P-221 Maria A. Wimmer, Marijn Janssen, Ann Macintosh, Hans Jochen Scholl, Efthimios Tambouris (Eds.) Electronic Government and Electronic Participation Joint Proceedings of Ongoing Research of IFIP EGOV and IFIP ePart 2013 16. – 19. September 2013, Koblenz

P-222 Reinhard Jung, Manfred Reichert (Eds.) Enterprise Modelling and Information Systems Architectures (EMISA 2013) St. Gallen, Switzerland September 5. – 6. 2013

P-223 Detlef Hühnlein, Heiko Roßnagel (Hrsg.) Open Identity Summit 2013 10. – 11. September 2013 Kloster Banz, Germany

P-224 Eckhart Hanser, Martin Mikusz, Masud Fazal-Baqaie (Hrsg.) Vorgehensmodelle 2013 Vorgehensmodelle – Anspruch und Wirklichkeit 20. Tagung der Fachgruppe Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik (WI-VM) der Gesellschaft für Informatik e.V. Lörrach, 2013

P-225 Hans-Georg Fill, Dimitris Karagiannis, Ulrich Reimer (Hrsg.) Modellierung 2014 19. – 21. März 2014, Wien

P-226 M. Clasen, M. Hamer, S. Lehnert, B. Petersen, B. Theuvsen (Hrsg.) IT-Standards in der Agrar- und Ernährungswirtschaft Fokus: Risiko- und Krisenmanagement Referate der 34. GIL-Jahrestagung 24. – 25. Februar 2014, Bonn

P-227 Wilhelm Hasselbring,
Nils Christian Ehmke (Hrsg.)
Software Engineering 2014
Fachtagung des GI-Fachbereichs
Softwaretechnik
25. – 28. Februar 2014
Kiel, Deutschland

P-228 Stefan Katzenbeisser, Volkmar Lotz,
Edgar Weippl (Hrsg.)
Sicherheit 2014
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 7. Jahrestagung des
Fachbereichs Sicherheit der
Gesellschaft für Informatik e.V. (GI)
19.–21. März 2014, Wien

GI-Edition Lecture Notes in Informatics – Seminars

S-1 Johannes Magenheim, Sigrid Schubert (Eds.):
Informatics and Student Assessment
Concepts of Empirical Research and Standardisation of Measurement in the Area of Didactics of Informatics

S-2 Gesellschaft für Informatik (Hrsg.)
Informationstage 2005
Fachwissenschaftlicher Informatik-Kongress

S-3 Gesellschaft für Informatik (Hrsg.)
Informationstage 2006
Fachwissenschaftlicher Informatik-Kongress

S-4 Hans Hagen, Andreas Kerren, Peter Dannenmann (Eds.)
Visualization of Large and Unstructured Data Sets
First workshop of the DFG's International Research Training Group "Visualization of Large and Unstructured Data Sets – Applications in Geospatial Planning, Modeling and Engineering"

S-5 Gesellschaft für Informatik (Hrsg.)
Informationstage 2007
Fachwissenschaftlicher Informatik-Kongress

S-6 Gesellschaft für Informatik (Hrsg.)
Informationstage 2008
Fachwissenschaftlicher Informatik-Kongress

S-7 Hans Hagen, Martin Hering-Bertram, Christoph Garth (Eds.)
Visualization of Large and Unstructured Data Sets

S-8 Gesellschaft für Informatik (Hrsg.)
Informatiktage 2009
Fachwissenschaftlicher Informatik-Kongress

S-9 Gesellschaft für Informatik (Hrsg.)
Informatiktage 2010
Fachwissenschaftlicher Informatik-Kongress

S-10 Gesellschaft für Informatik (Hrsg.)
Informatiktage 2011
Fachwissenschaftlicher Informatik-Kongress

S-11 Gesellschaft für Informatik (Hrsg.)
Informatiktage 2012
Fachwissenschaftlicher Informatik-Kongress

S-12 Gesellschaft für Informatik (Hrsg.)
Informatiktage 2013
Fachwissenschaftlicher Informatik-Kongress

S-13 Gesellschaft für Informatik (Hrsg.)
Informatiktage 2014
Fachwissenschaftlicher Informatik-Kongress

The titles can be purchased at:

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

Gesellschaft für Informatik e.V. (GI)

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
• seminars
• proceedings
• dissertations
• thematics
current topics are dealt with from the vantage point of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure high quality contributions.

The volumes are published in German or English.

Information: http://www.gi.de/service/publikationen/lni/

ISSN 1614-3213
ISBN 978-3-88579-447-9