

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 co-operation 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 978-3-88579-446-2



Gesellschaft für Informatik (Hrsg.): Informatiktage 2013



GI-Edition

Lecture Notes in Informatics

Gesellschaft für Informatik (Hrsg.)

Informatiktage 2013

Fachwissenschaftlicher Informatik-Kongress

22. und 23. März 2013

**B-IT Bonn-Aachen International Center
for Information Technology, Bonn**



Gesellschaft für Informatik (Hrsg.)

Informatiktage 2013
Smart Life – dank Informatik

Fachwissenschaftlicher Informatik-Kongress

22. und 23. März 2013

**B-IT Bonn-Aachen International Center for Information
Technology in Bonn**

Gesellschaft für Informatik e.V. (GI)

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

Volume S-12

ISSN 1614-3213

ISBN 978-3-88579-446-2

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

Series Editorial Board

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

Dieter Fellner, Technische Universität Darmstadt, Germany

Ulrich Flegel, Hochschule für Technik, Stuttgart, Germany

Ulrich Frank, Universität Duisburg-Essen, Germany

Johann-Christoph Freytag, Humboldt-Universität zu Berlin, Germany

Michael Goedicke, Universität Duisburg-Essen, Germany

Ralf Hofestädt, Universität Bielefeld, Germany

Michael Koch, Universität der Bundeswehr München, Germany

Axel Lehmann, Universität der Bundeswehr München, Germany

Peter Sanders, Karlsruher Institut für Technologie (KIT), Germany

Sigrid Schubert, Universität Siegen, Germany

Ingo Timm, Universität Trier, Germany

Karin Vosseberg, Hochschule Bremerhaven, Germany

Maria Wimmer, Universität Koblenz-Landau, Germany

Dissertations

Steffen Hölldobler, Technische Universität Dresden, Germany

Seminars

Reinhard Wilhelm, Universität des Saarlandes, Germany

Thematics

Andreas Oberweis, Karlsruher Institut für Technologie (KIT), Germany

© Gesellschaft für Informatik, Bonn 2013

printed by Köllen Druck+Verlag GmbH, Bonn

Wissenschaftliche Tagungsleitung

Alfred Zimmermann, Reutlingen University

Programmkomitee

Ehrenvorsitz

Rul Gunzenhäuser – Universität Stuttgart

Otto Spaniol – RWTH Aachen

Karlheinz Blank – T-Systems Stuttgart
Hermann Engesser – Informatik-Spektrum
Klaus-Peter Fähnrich – Universität Leipzig
Ulrich Furbach – Universität Koblenz-Landau
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
Lars Lippert – Microsoft München
Florian Matthes – TU München
Martin Mähler – IBM Ehningen
Doris Nitsche-Ruhland – DHBW Stuttgart
Gunther Piller – Fachhochschule Mainz
Karl Prott – Capgemini Hamburg
Wilfried Reimann – Daimler Stuttgart
Kurt Sandkuhl – Universität Rostock
Karsten Schweichhart – Deutsche Telekom Bonn
Andreas Speck – Universität Kiel
Gottfried Vossen – Universität Münster
Gerhard Weber – TU Dresden
Alfred Zimmermann – Hochschule Reutlingen (Chair)
Wolf Zimmermann – Universität Halle

Workshops - Themen und Partner

Capgemini: Agile Softwareentwicklung in der Praxis

Hilti AG: Cloud Computing @ Hilti: Vision, Design und Implementierung

K & A/TU Dresden: Einsatz neuer SW-Technologien - Neue Wege in der Mobile App-Entwicklung

iteratec GmbH: Enterprise Architecture Management - Grundlagen und Best-Practices

msg systems ag: Play! With the Cloud

Übersicht über das Rahmenprogramm und Partner der Informatiktage 2013

Neben den Workshops wurde den Teilnehmerinnen und Teilnehmern ein attraktives Rahmenprogramm geboten. Wir bedanken uns bei unseren Partnern, ohne deren Unterstützung die Informatiktage nicht hätten stattfinden können.

Rahmenprogramm:

Wettbewerbspräsentationen des informatiCup 2013

Wissenschafts- und Absolventen-Workshop

Leitung: Prof. Dr. Alfred Zimmermann, Fachhochschule Reutlingen

Hauptvortrag:

Prof. Dr.-Ing. Ina Schieferdecker, Fraunhofer-Focus

Wie Informatik Städte attraktiver macht

Talk-Show des Computer Clubs 2

Postersessions

Workshops in drei Sessions

Moderation der Veranstaltung:

Prof. Dr. Alfred Zimmermann, Reutlingen University

Partner und Sponsoren der Informatiktage:

Partner:

B-IT Bonn-Aachen Center

Coremedia

Think-cell

Workshop-Partner:

Capgemini

Hilti Aktiengesellschaft

iteratec Gesellschaft für iterative Softwaretechnologien mbH

msg systems

Kölsch & Altmann

Medienpartner:

IX

Computer Club 2

Jobguide

Staufenbiel Media GmbH

Vorwort zum Tagungsband 2013

Liebe Teilnehmerinnen und Teilnehmer,

die Informatiktage waren schon immer auch ein Labor für Zukunftsfragen, deren Lösung durch Informationstechnologien dort diskutiert und analysiert wurde. Erstmals in diesem Jahr haben wir die Informatiktage unter eine Leitüberschrift gestellt, die die Fantasie von Menschen weltweit beflügelt. „Smart Life“ bündelt Hoffnungen und Erwartungen an die nachhaltige Verbesserung der Lebensbedingungen durch intelligenten Informatik-Einsatz. Die Anwendungsbereiche sind vielfältig und umfassen die Vernetzung aller Lebensbereiche. Unser Hauptvortrag beschäftigt sich mit „Smart City“, mit den Schwerpunkten Kommunikation, Verkehr und Mobilität, Energie, Sicherheit und öffentliche Verwaltung.

Wir beleuchten diese Themen sowohl in unserem Hauptvortrag als auch in der Talk-Runde des Computer Clubs 2 und erhoffen uns einen lebhaften Austausch zwischen Wissenschaft und Wirtschaft.

Mit den Informatiktagen der Gesellschaft für Informatik bieten wir alljährlich ausgewählten Studierenden Gelegenheit, ihre fachlichen Ideen und ihre Zukunftspläne mit erfahrenen Wissenschaftlern zu diskutieren. Aber nicht nur die Wissenschaft steht im Vordergrund der Veranstaltung, vielmehr auch der Kontakt zu den IT-Unternehmen, die im Rahmen der Informatiktage Workshops anbieten, welche die Unternehmenswirklichkeit in der IT-Welt abbilden. Ohne die Beteiligung von engagierten Unternehmen wären die Informatiktage nicht möglich. Ihnen gilt also mein besonderer Dank!

Ich begrüße Sie herzlich zu den 13. Informatiktagen. Allen Teilnehmerinnen und Teilnehmern wünsche ich eine interessante und erfolgreiche Veranstaltung.



Prof. Oliver Günther, Ph.D. (GI-Präsident)

Inhaltsverzeichnis

Grundlagen der Informatik

Maria Dostert

Die parametrisierte Komplexität des Student Sectioning Problems..... 15

Oscar Dustmann

Symbolic Execution of Discrete Event Systems with Uncertain Time 19

Florian Kliem

*Tangent-Linear and Adjoint Derivative Code for the Non-Linear
Least Squares Solver Ceres* 23

Marcus Pöckelmann

Zyklizität in empirischen und künstlichen Nahrungsnetzen..... 27

Steffen Rechner

*Sampling perfekter Matchings
in Bezug auf die Struktur des unterliegenden Graphen*..... 31

Künstliche Intelligenz

Aykut Bußian

*Indikatoren gestützte Betrugsprognose anhand von Jahresabschluss-
und Buchführungsdaten in einem probabilistischen wissensbasierten System*..... 35

Dominik Kreuz, Matthias Mielke, Rainer Brück

Echtzeiterkennung von Umgebungsgeräuschen mit Android 39

Lukas Georgieff

Improving Ontology Construction Using Crowdsourcing and Machine Learning.... 43

Marcus Marto

*Implementation of a Transformation System for
Knowledge Bases Consisting of Relational, Probabilistic Conditionals*..... 47

Benjamin Milde

Crowdsourcing slang identification and transcription in twitter language 51

Stephan Opfer

*Limits of SROIQ in the Context of Reasoning
Support for Multi-Agent Plan Modeling*..... 55

Softwaretechnik

Franziska Anders, Michael Däubler, Kolja Hick, Denis Krajcevic <i>Model-View-Controller Pattern in a distributed System</i>	59
Kai Beckmann <i>Integration modellbasierten Testens in die Entwicklungsprozesse eines mittelständischen Unternehmens der Automatisierungsbranche</i>	63
Alexander Diener, Thomas Zink, Oliver Haase, Juergen Waesch <i>Universelle Traversierung von NAT-Geräten</i>	67
Johannes Geismann <i>Quelltextgenerierung für LEGO Mindstorms-Roboter</i>	71
Manuel Grandheit, Christoph Romstück, Patrick Kreutzer <i>FASL: Skriptsprache zur Programmierung mobiler Geräte</i>	75
Thomas Gutenberger, Perihan Bulut, Meryem Cetinkaya, Yasemin Guruhan, Philip Maier, Buket Oezmen, Mehtap Yilmaz <i>Ein Pattern Katalog für Qualitätsbewertungsmethoden für Software Architekturen</i>	79
Norman Hansen, Dominik Franke <i>Reverse Engineering des Windows Phone Process Execution Models</i>	83
Daniel Krämer <i>Systematisches Reengineering eines Softwaresystems am Beispiel einer Sprachtherapie-Software</i>	87
Sebastian Kunze <i>Anwendung der Nutzwertanalyse zur Auswahl von Frameworks zur Unterstützung testgetriebener Softwareentwicklung</i>	91
Johannes Schauer <i>Bootstrapping of component based software systems</i>	95
Jochen Wierum <i>Aufbau und Auswertung wiederverwendbarer Strukturen aus Ticketssystem- und Versionskontrollsystem-Daten für multiple Analysen</i>	99

Mehtap Yilmaz, Andrew Page, Markus Riegger <i>Dokumentation von Software Architekturen</i>	103
--	-----

Dennis Ziegenhagen <i>Konzeption und Implementierung eines Risiko- und Chancen-Management-Moduls für die Projekt + Business Software admileo</i>	107
--	-----

Mensch-Computer-Interaktion

Tanja Blascheck, Michael Raschke, Thomas Ertl <i>eTaddy - Ein integratives Framework für die Erstellung, Durchführung und Analyse von Eyetracking-Daten</i>	111
---	-----

Sebastian Klingenbeck, Tayfur Coskun, Teodora Velikova, Johannes Roith, Gudrun Klinker <i>Fog of Triage: Usage of the Fog of War concept for the Triage in a Mass Casualty Incident (MCI)</i>	115
---	-----

Rüdiger Marwein <i>Koaktivitäts- und Interaktivitätsbewertung kooperativer Software-Werkzeuge</i>	119
---	-----

Jessica Millberg <i>Markerloses, modellbasiertes Echtzeit-Tracking für AR-Applikationen</i>	123
---	-----

Alexander Phleps <i>Ambient Assisted Living: Aktueller Stand und Ausblick</i>	127
---	-----

Carolin Reichherzer, Jens Epe <i>Mobile Augmented Reality Browser in Combination with Magazines: Problems Concerning Usability and Content</i>	133
--	-----

Johannes Roith, Teodora Velikova, Sebastian Klingenbeck, Tayfur Coskun, Gudrun Klinker <i>Gestairboard: A gesture-based touch typing keyboard using the Kinect camera</i>	137
---	-----

Nicolas Simon <i>Analysieren und Bewerten der Einsatzmöglichkeiten des EPOC Neuroheadsets von Emotiv</i>	141
--	-----

Benjamin Vogler, Jan Schäfer <i>Ein Fehlertoleranzkonzept für dynamische, verteilte OSGi-Anwendungen</i>	145
--	-----

Datenbanken und Informationssysteme

Felix Gessert, Florian Bücklers

ORESTES: ein System für horizontal skalierbaren Zugriff auf

Cloud-Datenbanken 149

Jan Kristof Nidzwetzki

Optimierte Speicherzuteilung an Operatoren in Datenbanksystemen 153

Fabian Ohler, Kai Schwarz

Bewertung von Diskriminierungsnetzwerken für regelbasierte Systeme 157

Alexander van Renen, Wolf Rödiger

Thesis: The Distributed Radix Join Algorithm -

Efficient Distributed Join Processing on Modern Hardware 161

Ricardo Usbeck

Analyse von Mikro-Blog-Daten 165

Gero Willmes

Query Optimization for Large Sets of Predicates in the

Database Management System SECONDO 169

Technische Informatik

Sebastian Berschneider

Modellbasierte Hardwareentwicklung am Beispiel eingebetteter

Prozessoren für die optische Messtechnik 177

Daniel Danner

SLOTH ON TIME: Effiziente Zeitsteuerung in einem unterbrechungsgesteuerten

Echtzeitbetriebssystem 181

Dzenan Dzafic, Dominik Franke

Entwicklung und Evaluation eines Navigationssystems für Elektrorollstühle 185

Thomas Falkenberg

Browser-basierte SIP-Telefonie auf Grundlage von WebRTC 189

Thim Strothmann

Self-Optimizing Binary Search Trees - A Game Theoretic Approach 193

Informatik in den Lebenswissenschaften

**Teodora Velikova, Tayfur Coskun, Sebastian Kligenbeck, Johannes Roith,
Eva Artinger, Gudrun Klinker**

*Animation-based requirements analysis of a map application user
interface to support critical MCI situations 197*

Natalie Wenz, Anja Rothbart

Konzeption und Implementierung einer medizinischen Multi-Touch-Anwendung 201

Graphische Datenverarbeitung

Christoph Drexler

*Abschätzung des Disparitätsraums für das Stereo-
Matching mithilfe von Bildmerkmalen 205*

Jorin Guderian

*Entwicklung einer Kleidungssimulation für
3D-Charaktere auf Basis eines Masse-Feder-Systems 209*

Stephan Helou

Computational Fluid Dynamics within LibGeoDecomp 213

David A. Plecher

Archäologie..... 217

Ruben Reiser

*Approximation von Reflektanzmessungen mit Vektorraummodellen
und Latent Semantic Indexing..... 221*

Kai Stenkamp

*Positionsabhängige perspektivische Darstellung von 3D-Inhalten auf
mobilen Endgeräten 225*

Jan Sutter

A Real-Time Visualization Infrastructure 229

Wirtschaftsinformatik

Adnan Ademovic, Christian Kovac, Denise Uslu, Betlul Acekza, Adil Sahiner

*Methode zur Evaluierung von Public Cloud IaaS Providern unter
Berücksichtigung individueller Anforderungen 233*

Michael Däubler, Franziska Anders, Kolja Hick, Denis Krajcevic, Philipp Maier, Daniel Stiefel	
<i>Service Technologien – SOA Integrationsmodell</i>	237
Christopher Jud	
<i>Effektivität von Multiagenten-Organisationstrukturen am Beispiel der Störgeräuscherkennung in Fahrzeugen</i>	241
Fabian Meyer	
<i>Wissensbasiertes automatisiertes IT-Management</i>	245
Dirk Stamer	
<i>Informationslogistisches Rahmenwerk für bedarfsgerechte Informationsversorgung in Unternehmen</i>	249
Daniel Stiefel, Eberhard Heber	
<i>Campus2Go - Content Enrichment Using Enterprise Integration Patterns.....</i>	253
Mehtap Yilmaz, Markus Riegger, Benjamin Hasimov, Usama Mir, Agim Imeraj, Dino Delic	
<i>Einführung in Business Process Management und die Entwicklung von Business Process Model & Notation Patterns</i>	257

Sicherheit – Schutz und Zuverlässigkeit

Alexander Beifuß	
<i>Improving Performance and Precision of UniLoG's Load Generation Process by Utilizing Realtime Enhancing Features of the Operating System</i>	261
André Gimbut	
<i>Verschlüsselte Sprachkommunikation zwischen Android-Smartphones</i>	265

Informatik und Ausbildung / Didaktik der Informatik

Michael Gruben, Johannes Freudenmann	
<i>Analyse und Vergleich von Autorensystemen für ein WBT zu Vorlesungsinhalten ...</i>	269
Michael Müller	
<i>Design and Implementation of the UULM Network Lab</i>	271

Die parametrisierte Komplexität des *Student Sectioning Problems*

Maria Dostert

M.Dostert@hochschule-trier.de, Abgabedatum: 01.01.2013

Betreuer: Prof. Dr. Heinz Schmitz (Hochschule Trier),

Prof. Dr. Peter Rossmanith (RWTH Aachen)

Master-Abschlussarbeit im Fachbereich Informatik der Hochschule Trier

GI-Fachbereich: Grundlagen der Informatik

Abstract: Das *Student Sectioning Problem (SSP)* stellt die Frage, ob alle Studierenden an den Übungen ihrer gewünschten Module konfliktfrei teilnehmen können. Die Analyse der parametrisierten Komplexität dieses Problems ergibt, dass sich das parametrisierte Problem (SSP, p) in der Komplexitätsklasse FPT befindet, wobei p die Gesamtzahl der Übungen ist. Des Weiteren wird die *para-NP-Vollständigkeit* für verschiedene andere Parametrisierungen von SSP nachgewiesen.

1 Einleitung

Die Studierenden einer Hochschule nehmen üblicherweise jedes Semester an einer Reihe von Lehrveranstaltungen und zugehörigen Übungen teil. Für die Hochschulen ergibt sich dadurch unter anderem das Problem, dass die Studierenden auf die verfügbaren Übungsgruppen aufzuteilen sind und zwar am besten so, dass die Studierenden problemlos an sämtlichen Übungen teilnehmen können. Restrangiert wird dieses Problem durch verschiedene Bedingungen, da zum Beispiel die Teilnehmerzahl der Übungsgruppen beschränkt ist. Bei der Studierendenverteilung sind die Termine der Übungsgruppen zu berücksichtigen, um Terminkonflikte für Studierende zu vermeiden. Daher basiert dieses Problem auf einem existierenden Stundenplan. Das *Student Sectioning Problem (SSP)* adressiert genau dieses Problem, denn es stellt die Frage, ob alle Studierenden an den Übungen ihrer gewünschten Module konfliktfrei teilnehmen können. In der Arbeit wird die parametrisierte Komplexität des SSP analysiert, in dem unterschiedliche Parametrisierungen des SSP untersucht werden.

2 Problemstellung

Es gibt mehrere verschiedene Varianten des SSP , die sich beispielsweise in der Eingabe oder in den Bedingungen unterscheiden. Das hier zu untersuchende SSP wird so gewählt,

dass es realitätsnah ist und somit beispielsweise Studierende ihre Module wählen können und Übungen an mehreren Terminen stattfinden dürfen.

Nachfolgend wird die Definition des *SSP* angegeben. Diese enthält den Begriff *Studierenden/Modul-Kombination*, der ein Tupel eines Studierenden mit einem seiner gewählten Module bezeichnet. Das *SSP* wird als Entscheidungsproblem definiert, wobei die Anzahl unverteilter *Studierenden/Modul-Kombinationen* bestenfalls minimiert wird.

Definition: *Student-Sectioning-Problem (SSP)*

Eingabe: Module $M = \{m_1, \dots, m_v\}$
 Studierende $S = \{s_1, \dots, s_r\}$
 Übungsgruppen $U = \{u_{11}, \dots, u_{vj}\}$
 Übungsgruppenkapazität $\tau: U \rightarrow N^+$
 Modulwahl $\beta: S \rightarrow 2^M \setminus \emptyset$
 Terminkonflikte $\delta: U \times U \rightarrow \{0, 1\}$
 Grenze unverteilter Studierenden/Modul-Kombinationen $k \in N$

Frage: Existiert eine Verteilung der Studierenden auf Übungsgruppen, so dass alle, bis auf maximal k , Studierenden/Modul-Kombinationen genau eine Übungsgruppe des Moduls zugewiesen bekommen, ohne die Kapazität einer Übungsgruppe zu überschreiten und Terminkonflikte zu erzeugen?

Wie bereits in der Einleitung erwähnt, basiert das *SSP* auf einem existierenden Stundenplan. Für das *SSP* sind die konkreten Termine einzelner Übungsgruppen irrelevant, denn entscheidend ist nur, welche Übungsgruppen sich terminlich überschneiden. Die in der Eingabe enthaltene Funktion δ stellt die Terminkonflikte dar, indem sie für je zwei Übungsgruppen 1 zurückliefert, falls sie mindestens einen gemeinsamen Termin haben, ansonsten 0. So wird die Anzahl der Termine einer Übungsgruppe nicht vorgegeben.

3 Parametrisierte Komplexität

Schwer lösbare Probleme haben die Eigenschaft, dass kein zugehöriger Algorithmus mit einer polynomiellen Laufzeit bekannt ist. Die Laufzeit eines Algorithmus ist in der parametrisierten Komplexitätsanalyse, im Gegensatz zur klassischen Komplexitätsanalyse, nicht nur von der Eingabelänge abhängig, sondern auch von einem angegebenen Parameter. Die Untersuchung der parametrisierten Komplexität des *SSP* zeigt somit, welche Auswirkung ein konkreter Parameter auf die Laufzeit eines Algorithmus für das *SSP* hat. Für die parametrisierte Komplexitätsanalyse sind daher vor allem schwer lösbare Probleme interessant. Ein parametrisiertes Problem (Q, κ) ist ein Tupel, bestehend aus einem Problem Q und einer Parametrisierung $\kappa(x): N \rightarrow N$ abhängig von der Eingabe x . Falls eine Funktion $f: N \rightarrow N$ und ein Polynom p existiert, so dass die Laufzeit eines parametrisierten Algorithmus durch $f(\kappa(x)) \cdot p(|x|)$ beschränkt werden kann, lässt sich dieses Problem effizient und exakt lösen, falls der Wert der Parametrisierung klein ist. Solche parametrisierten Algorithmen werden als *FPT-Algorithmen* bezeichnet. Alle parametrisierten

Probleme zu denen ein *FPT-Algorithmus* existiert, befinden sich in der Komplexitätsklasse *FPT*. Falls es zu einem parametrisierten Problem (Q, κ) einen nichtdeterministischen Algorithmus mit der Laufzeit $f(\kappa(x)) \cdot p(|x|)$ gibt, so gilt $(Q, \kappa) \in \text{para-NP}$. Da sich das Resultat der Analyse der parametrisierten Komplexität des *SSP* ausschließlich auf diese beiden Komplexitätsklassen bezieht, werden diese in der Arbeit einleitend vorgestellt, bevor nachfolgend auf die Analyse eingegangen wird.

Vor der Analyse der parametrisierten Komplexität wird gezeigt, dass $\text{SSP} \in \text{NP}$, denn daraus folgt $(\text{SSP}, \kappa) \in \text{para-NP}$ für eine beliebige Parametrisierung κ [FG06]. Da sich bereits die Eingabelänge des *SSP* mit den Parametern c (maximale Übungsgruppenkapazität), l (Anzahl Übungsgruppen) und k beschränken lässt, gilt $(\text{SSP}, \kappa) \in \text{FPT}$ mit $\kappa(x) = c + l + k$. Diese Parametrisierung ist allerdings nicht sehr hilfreich, denn für Instanzen mit einer großen Eingabelänge sind die Werte dieser Parameter ebenfalls groß.

Im zweiten Teil der Arbeit zeigen wir, dass eine Reduktion $3\text{-SAT} \leq_m^p \text{SSP}$ existiert, bei der der maximalen Übungsgruppenzahl eines Moduls (u), der maximalen Terminanzahl einer Übung (t), sowie c und k immer ein fester Wert zugeordnet wird ($u = 2, t = 3, c = 3, k = 0$). Hieraus ergibt sich die *para-NP-Vollständigkeit* aller parametrisierten *SSP*-Varianten, deren Parameter eine Teilmenge aus $\{u, t, c, k\}$ bilden [FG06].

In den folgenden Abschnitten der Arbeit wird das *SSP* mit dem Parameter l untersucht. *Hendrik W. Lenstra* zeigte 1983, dass das *ILPF* (= *Integer Linear Programming Feasibility Problem*) für eine feste Anzahl der Variablen des übergebenen *ILP* effizient lösbar ist [L83]. In der Arbeit gelingt eine Modellierung von *SSP* als *ILP* in der Art, dass die Variablenanzahl ausschließlich von l abhängt. Auf Grund des Resultats von *Lenstra* gilt somit $(\text{SSP}, l) \in \text{FPT}$ [R11]. Aus diesem Resultat folgt wiederum, dass sich alle parametrisierten *SSP*, die als Parameter unter anderem l enthalten, in *FPT* befinden.

Im letzten Teil der Arbeit geben wir eine Reduktion $3\text{-DM} \leq_m^p \text{SSP}$ an und analysieren ihre Eigenschaften. Für die Reduktion gilt, dass $k = \#T - m$ und da 3-DM für $m = \#T$ effizient lösbar ist, folgt aus dieser Reduktion die *NP-Vollständigkeit* des *SSP* für $k > 0$. Die Parameter c, u, t, s (mit $s = \#S$) erhalten durch die Reduktion einen festen Wert ($c = 1, u = 1, t = 3, s = 1$), woraus wiederum die *para-NP-Vollständigkeit* aller parametrisierten *SSP*, deren Parameter in $\{c, u, t, s\}$ enthalten sind, für $k > 0$ folgt.

Die nachfolgende Tabelle enthält alle erzielten Resultate. Die Spalten repräsentieren jeweils einen untersuchten Parameter und die Zeile das Ergebnis der Komplexität. Der entsprechende Zelleninhalt gibt an, wodurch dies bewiesen wurde.

	Max. Anzahl Studierender	Max. Anzahl Termine einer Übungsgruppe	Max. Anzahl Übungsgruppen eines Moduls	Max. Kapazität einer Übungsgruppe	k (Definition SSP)	Anzahl Übungsgruppen
FPT	(SSP, Anzahl Übungsgruppen) \in FPT					ILP
para-NP-vollständig	$3\text{-DM} \leq_m^p \text{SSP}$ (für $k > 0$)	$3\text{-SAT} \leq_m^p \text{SSP}$	$3\text{-SAT} \leq_m^p \text{SSP}$	$3\text{-SAT} \leq_m^p \text{SSP}$	$3\text{-SAT} \leq_m^p \text{SSP}$	
	$3\text{-SAT} \leq_m^p \text{SSP}$					
	$3\text{-DM} \leq_m^p \text{SSP}$ (für $k > 0$)					

Abbildung 1: Übersicht der Resultate

Literatur

Hendrik Willem jun. Lenstra. Integer programming with a fixed number of variables, *Math. Oper. Res.*, 1983.

Jörg Flum und Martin Grohe. Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series), *Springer-Verlag New York, Inc.*, 2006.

Rolf Niedermeier. Invitation to Fixed-Parameter Algorithm *OUP Oxford*, 2002.

Ravi Kannan. Improved algorithms for integer programming and related lattice problems *Math. Oper. Res.*, 1987.

Peter Rossmanith. Parameterized Algorithms. Vorlesungsskript, *RWTH Aachen*, 2011.

Arne Meier, Johannes Schmidt, Michael Thomas und Heribert Vollmer. On the parameterized complexity of the default logic and autoepistemic logic. In: Proceedings of the 6th international conference on Language and Automata Theory and Applications, LATA'12, *Berlin, Heidelberg, 2012, Springer-Verlag*

Heinz Schmitz. Algorithmik. Vorlesungsskript, *Hochschule Trier*, 2008.

Heinz Schmitz. Theoretische Informatik. Vorlesungsskript, *Hochschule Trier*, 2011.

Symbolic Execution of Discrete Event Systems with Uncertain Time

Oscar Soria Dustmann
Communication and Distributed Systems (ComSys)
RWTH Aachen University, Germany
oscar.dustmann@comsys.rwth-aachen.de

Abstract:

Distributed and networked systems are difficult to debug. The inherent concurrency and communication delay are the primary sources of many errors involving timing. However, whether by hand, randomised, rigorously, or heuristically, timing bugs are hard to find, understand and repair. Timing related issues tend to hide in obscure corner-cases; For example, a catastrophic protocol lock-up might manifest only for an unlikely packet reordering which the protocol designer did not expect. Nevertheless, especially in systems that are subjected to extreme conditions or run for an extraordinary long time span, such unforeseen consequences might arise at some point. Consequently, a system that seemed to perform flawlessly in previous experiments could then inexplicably fail in production.

Prevalent tools for analysing distributed software systems with regard to their timing properties exhaustively explore different timing conditions to achieve coverage of execution paths. When checking an actual implementation, this methodology attempts to enumerate a large collection of possible behaviours in the expectation to eventually stumble upon erroneous ones. For example, in light of packet reordering, searching bugs with brute force is inefficient, rendering rigorous state-exploration tools unsuited for this particular class of errors.

In this paper, we present SymTime — a novel technique to analyse timing related issues in distributed systems. We employ the symbolic execution paradigm to model the uncertainty in timed events and apply this approach to the analysis of discrete event systems. The key idea of SymTime is to run systems under symbolic time rather than concrete time and thus non-redundantly execute paths representing equivalence classes of temporal behaviour. As the main contribution of this work, we provide the formal foundation and the core algorithms as well as their implementation in the context of a state of the art symbolic execution tool.

1 Introduction

Over the last decade communication between individual systems evolved in efficiency, flexibility and robustness. The employment of distributed systems has become an almost ubiquitous commodity in most areas of every-day life, as well as in the professional domain. Researchers all over the globe have worked together on designing increasingly effective networking protocols that enable several isolated hosts to cooperate. The overall resilience of systems that are the result of such joint efforts has since increased the perceived reliability of information technologies.

It is the aim of our group to research how such systems can be subjected to rigorous analysis with specialised evaluation tools [SLAW08, SLA⁺10, SD10, SSDK⁺11]. In this work, we present a formal and practical model to describe how a program can be executed in a fashion to cover all possible concretisations of an event execution time that is uncertain. This means that the execution time of an event, if uncertain, is in fact the representation of a collection of event times that could occur in a proper, concrete run of the deployed system.

2 Symbolic Time

As in previous research into *Symbolic Distributed Execution (SDE)* [SSDK⁺11], it is useful to represent a distributed system with n nodes as a set of n independent states governed by an execution engine. This approach decomposes the distributed system into its n partitions, allowing to employ highly effective overhead reduction algorithms. The composition of such n states is referred to as (network) *scenario*. Observe that the execution engine might choose to *fork* any state at any moment in order to reflect the branching of two different execution paths the code might follow. Due to this decomposition, transmissions between separate nodes must be handled by the engine since the transmitted packet crosses the boundaries of individual states [SD10, SSDK⁺11]. The basic achievement there is the significant reduction of the state space, resulting in a total speed up by orders of magnitude. This is accomplished by allowing individual states to represent any number of scenarios while keeping network communication consistent.

Previous research into SDE always assumed instantaneous data transmission between neighbouring nodes, while deferring the simulation of network latency to the implementation of the client's network model. Consequently, packets were considered to take immediate effect. Due to the compressed nature of all scenarios, this effect may introduce conflicts when and if a state represents scenarios where a packet was sent and scenarios where that particular packet was not sent. Such conflicts can be resolved by forcibly forking the respective receiver state and delivering the packet to only one copy.

2.1 Introduction of Uncertain Time

As we are interested in allowing packet transmissions to be simulated with an uncertain delay, we must generalise the model of SDE to take a packet's delay into account. This generalisation results in a potential temporal separation of cause and effect, i.e., sending and receiving a packet cease to coincide.

A collection of packets, or more generally pending events, can be governed by a priority or calendar queue, which allows efficient extraction of the earliest event at any given time. Such data structures are the corner stone of classical *Discrete Event Simulation (DES)* engines. However, in the context of events with uncertain timestamps, the implicitly used strict total order of events, induced by their concrete, fixed timestamps is no longer given.

For instance, given concrete events e_0, e_1 and e_2 with timestamps $t_0 = 4, t_1 = 7$ and $t_2 = 42$ respectively, a classical DES will keep these three events ordered as (e_0, e_1, e_2) . Note that classical DES break ties in some arbitrary fashion, e.g., by drawing random numbers or by sorting coincident events in the order in which they were added. However, we assume now uncertainty in the timestamp of events, for instance by allowing a discrepancy of ± 2 units of time for every timestamp. This can be modelled by generalising the associated timestamps from fixed cardinal numbers to sets of numbers: $T_0 = \{2, \dots, 6\}$, $T_1 = \{5, \dots, 9\}$ and $T_2 = \{40, \dots, 44\}$. As one can easily see, there is no meaningful way to decide which of the events with timestamps T_0 and T_1 is to be run first.

This situation can be resolved by what is known as *Interval Branching* [PML08]; Since there are two eligible events to be run, the state on which both events are scheduled is forcibly forked. For each copy, a distinct event ordering will be determined by running e_0 (at time T_0) on one copy and e_1 (at time T_1) on the other. The respectively unexecuted event remains in the event queue in both cases.

2.2 Temporal Violation

We analyse one of these states individually: Consider the state where the events are executed in the order (e_0, e_1, e_2) and ignore the state where the event ordering (e_1, e_0, e_2) is chosen. As we model the uncertainty in time symbolically, e_0 runs at all times T_0 simultaneously and afterwards e_1 is run on all T_1 . This however has the potential to violate temporal monotonicity, as it includes the concrete case where e_0 is run at time 6 and e_1 is run at time 5 — before e_0 .

Hence, additional effort must be invested to ensure that the code run on this state will never follow such an invalid path. Symbolic expressions, including symbolic timestamps like T_0 , are described by a number of read operations from memory locations, any number of operators and a set of *constraints*. For example, the timestamp $T_0 = \{2, \dots, 6\}$ would be represented as a read operation from the memory location where the execution time for e_0 is stored, say a_0 , and would be denoted as $\text{read}(a_0)$. Additionally, the corresponding state would carry two constraints: $2 \leq \text{read}(a_0)$ and $\text{read}(a_0) \leq 6$. That is, constraints are represented as boolean expressions, with the implication of always evaluating to true.

To determine admissible values for a given symbolic expression, including symbolic timestamps like T_0 , the execution engine queries a solver, passing both the given expression and its private set of constraints.

This technical representation can be used to ensure the determination of ambiguous event orderings to be causally consistent, avoiding the previous example where e_1 included a concrete timestamp preceding e_0 . One possible solution is to introduce a new constraint for each additional (ambiguous) event. That is, if the timestamp of event e_i is given as the expression x_i , evaluating to the set of times T_i , we can reflect a determined ordering of e_i before e_j by introducing a new constraint $x_i \leq x_j$ to the constraint set of the respective state. Clearly, this eliminates any execution path which narrows down the symbolic timestamps of events to contradicting values.

Acknowledgments

This paper is mainly based on firstly the results of the author's master thesis of the same name, concluded in December, 21st 2012 [SD12]. Secondly, a peer reviewed workshop paper highlighting the ongoing research on the topic which was published in April 2012 [SDSW12].

I thank Raimondas Sasnauskas and Klaus Wehrle for their support, both in general and specifically during writing of my theses.

References

- [PML08] Patrick Peschlow, Peter Martini, and Jason Liu. Interval Branching. In *22nd Workshop on Principles of Advanced and Distributed Simulation*, PADS '08, pages 99–108. IEEE, 2008.
- [SD10] Oscar Soria Dustmann. Scalable Symbolic Execution of Distributed Systems, 2010. Bachelor's Thesis.
- [SD12] Oscar Soria Dustmann. Symbolic Execution of Discrete Event Systems with Uncertain Time, 2012. Master's Thesis.
- [SDSW12] Oscar Soria Dustmann, Raimondas Sasnauskas, and Klaus Wehrle. Symbolic System Time in Distributed Systems Testing. In *Testing: Academic and Industrial Conference - Practice and Research Techniques*, TAIC PART, pages 893–894, 2012.
- [SLA⁺10] Raimondas Sasnauskas, Olaf Landsiedel, Muhammad Hamad Alizai, Carsten Weise, Stefan Kowalewski, and Klaus Wehrle. KleeNet: discovering insidious interaction bugs in wireless sensor networks before deployment. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '10, pages 186–196. ACM, 2010.
- [SLAW08] Raimondas Sasnauskas, J6 Ágila Bitsch Link, Muhammad Hamad Alizai, and Klaus Wehrle. KleeNet: automatic bug hunting in sensor network applications. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, pages 425–426. ACM, 2008.
- [SSDK⁺11] Raimondas Sasnauskas, Oscar Soria Dustmann, Benjamin Lucien Kaminski, Klaus Wehrle, Carsten Weise, and Stefan Kowalewski. Scalable Symbolic Execution of Distributed Systems. In *Proceedings of the 2011 31st International Conference on Distributed Computing Systems*, ICDCS '11, pages 333–342. IEEE, 2011.

Tangent-Linear and Adjoint Derivative Code for the Non-Linear Least Squares Solver Ceres

Florian Kliem

florian.kliem@rwth-aachen.de

Abstract: For the non-linear least squares solver *Ceres*, algorithmic differentiation can be changed from tangent-linear dual numbers to an adjoint mode implementation. *dco/c++* is used to do so. Results show that in typical applications, this produces favorable, i.e. faster, computations of the Jacobian matrix.

1 Motivation: Least Squares

Ceres is a non-linear least squares solver developed by Google Inc. [AM12]. This thesis had the aim to investigate it and improve its algorithmic differentiation (AD) implementation by inclusion of the *dco/c++* overloading AD tool currently developed at STCE [Nau11].

This task seemed promising because a large amount of run time for numeric solvers is typically spent on the calculation of derivatives. Especially the Jacobian matrix $J = \nabla F(\mathbf{x}_k)$ of a cost function $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ evaluated at interim solutions $\mathbf{x}_k \in \mathbb{R}^n$ takes a central place in the solver algorithms. Its efficient computation therefore has a major influence on the run time performance of any numerical solver. For better understanding, consider

$$F: \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad \mathbf{y} = F(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^m. \quad (1)$$

Using the method of least squares, the aim becomes to solve a linearized problem

$$\min_{\Delta \mathbf{x}} \frac{1}{2} \|J(\mathbf{x})\Delta \mathbf{x} + F(\mathbf{x})\|^2, \quad (2)$$

where $\Delta \mathbf{x} \in \mathbb{R}^n$ is the discrete step of a trust region method and $J(\mathbf{x})$ denotes the Jacobian matrix of F at point \mathbf{x} , thus $J_{ij}(\mathbf{x}) = \frac{\partial F_i}{\partial x_j}$, $J \in \mathbb{R}^{m \times n}$.

2 Algorithmic Differentiation

In general, one might assume $m \approx n$. The calculation of the Jacobian can be accelerated by *exploitation of structure*, e.g. if the number of outputs locally becomes $\hat{m} \ll n, m$.

In Ceres, this is possible because the interface for formulating the cost functions F requires that the resulting problems are partially separable. It is therefore possible to change the core AD mechanism from Ceres' intrinsic forward mode AD to a reverse mode available from `dco/c++`. This change is very good for problems where a high-dimensional input $\mathbf{x} \in \mathbb{R}^n$ produces a very low-dimensional output $\mathbf{y} \in \mathbb{R}^{\hat{m}}$ locally. The Jacobian can be computed faster because reverse mode only needs \hat{m} evaluations of F , whereas forward mode needs n calls of the function.

Ceres' intrinsic AD uses *dual numbers* similar to [Che94]. A dual number $d \in \mathbb{D}$ is hereby considered as $d = x + \epsilon x'$ where ϵ fulfills $\epsilon \neq 0$, but $\epsilon^2 = 0$. Formally, this leads to the introduction of dual numbers as a superset $\mathbb{D} \supset \mathbb{R}$ where $\mathbb{D} = \{(x, x') | x, x' \in \mathbb{R}\}$. The reverse holds because $\mathbb{R} = \{d = (x, x') | x' = 0\}$. The concept could be compared to the complex numbers, where $i^2 = -1$.

The dual number approach is in fact a *tangent-linear model*, represented as

$$\mathbf{y}^{(1)} = \nabla F(\mathbf{x}) \cdot \mathbf{x}^{(1)}. \quad (3)$$

By sequentially setting $\mathbf{x}^{(1)}$ to all Cartesian basis vectors of \mathbb{R}^n and evaluating an overloaded $F^{(1)}$, J can be accumulated in a complexity of $\mathcal{O}(n) \cdot \text{Cost}(F)$.

Adjoint mode AD, on the other hand, forms partial derivatives in *reverse mode*. Instead of the forward differentiation shown in (3), this gives the adjoint Jacobian $J^T = (\nabla F)^T$ via

$$\mathbf{x}_{(1)} = (\nabla F(\mathbf{x}))^T \cdot \mathbf{y}_{(1)}. \quad (4)$$

The full Jacobian can be accumulated by passing over all Cartesian base directions. While the tangent-linear code had to seed all possibilities within $\mathbb{R}^n \ni \mathbf{x}^{(1)}$, the adjoint seed takes place on $\mathbf{y}_{(1)} \in \mathbb{R}^m$. Computational complexity becomes $\mathcal{O}(m) \cdot \text{Cost}(F)$. In general, for most problems requiring a gradient (i.e. $m = 1$), reverse mode AD is preferable.

Implementational changes to Ceres' AD interface composed the core contribution of this work: As mentioned above, the existing implementation uses a tangent-linear data type, `ceres::Jet`, and its driver class `ceres::internal::AutoDiff`. In a first step, this driver class was replaced by counterparts that were based on the tangent-linear types `dco::t1s` (scalar) and `dco::t1v` (vectorized).

Afterwards, the tangent-linear parts were completely replaced by adjoint modelling. This was realized by providing derivative code based on the default class `dco::a1s` and using standard tape functionalities. For more details about the `dco/c++` AD tool, reference should be made to [Nau11].

3 Experimental Analysis

3.1 Sum of X Squared

The implemented changes were tested for two problems, with the first being an easy textbook example from [Nau11, Equation 1.2]:

$$F : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad F(\mathbf{x}_i) = f(\mathbf{x}_i) - \mathbf{y}_i = \left(\sum_{j=0}^{n-1} (\mathbf{x}_{i_j})^2 \right)^2 \quad (5)$$

where $\mathbf{x}_i \in \mathbb{R}^n$ should be regarded as parameter vectors for which each f is minimized, and $f(\mathbf{x}_i) \in \mathbb{R}$ is an one-dimensional output. i counts various, different observations.

3.2 Bundle Adjustments for Dubrovnik

According to its manual [AM12, Chapter 10], "one of the main reasons for writing Ceres" was the wish to reconstruct huge visual sceneries (e.g. the touristic city centres of Rome or Dubrovnik [ASS⁺09]) through the use of large image collections for construction of so-called bundle adjustment problems.

A detailed description of bundle adjustments in general should be taken from [TMHF99]. The underlying camera model comes from [HZ03, Chapter 6]. In the context of this paper, it is sufficient to note that for each observation j , the model yields

$$F : \mathbb{R}^{12} \rightarrow \mathbb{R}^2, \quad r_{x|y} = F(\mathbf{c}, \mathbf{p}). \quad (6)$$

\mathbb{R}^{12} comes from the nine-parameter camera model $\mathbf{c} \in \mathbb{R}^9$ and a real world 3D coordinate point $\mathbf{p} \in \mathbb{R}^3$. r is the residual, i.e. the 2D reprojection error of the observation.

(6) has to be solved for every observation $j \in \{1..k\}$ seperately.

3.3 Results

The different performance can be seen in direct comparison of differential times.

The positive effects of adjoint show best in the log-log plot of Figure 1: For (5), the tangent-linear approach of Ceres' intrinsic `AutoDiff` implementation has an exponential complexity, and run times increase drastically according to $\mathcal{O}(n^2)$. This can be recognized by the slope within the plot. On the other hand, the adjoint mode of `DCO_Driver_Als` with its complexity of $\mathcal{O}(m) \cdot Cost(F)$ remains less dependent from the problem size – its slope indicates the expected simple linear increase with regard to input dimension n , which actually comes from $Cost(F) \in \mathcal{O}(n)$.

Figure 2 visualizes the direct comparison of time spent for differential evaluation in the Dubrovnik bundle adjustment dataset. Although (6) only has $n/m = 12/2$, which might not be as favorable for adjoint mode, `DCO_Driver_Als` still outperforms `AutoDiff` at all problems by roughly 5–10%.

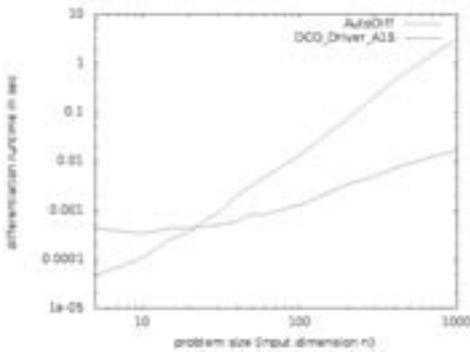


Figure 1: x_squared: Overall time for differentiation with regard to input dimension n . The red line denotes Ceres' AutoDiff, green DCO_Driver_A1S.

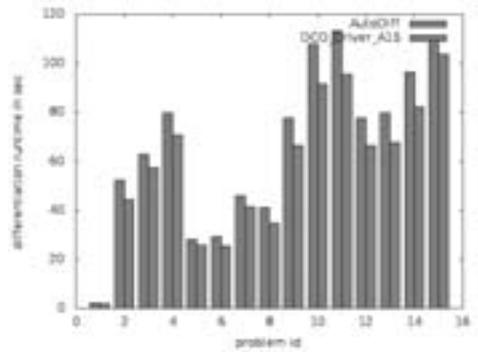


Figure 2: Dubrovnik: Overall time for differentiation while solving the given bundle adjustments no. 1–15. Red denotes AutoDiff, green DCO_Driver_A1S.

References

- [AM12] Sameer Agarwal and Keir Mierle. *Ceres Solver: Tutorial & Reference (Version 1.4.0)*. Google Inc., November 2012.
- [ASS⁺09] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building Rome in a Day. In *International Conference on Computer Vision*, Kyoto, Japan, 2009.
- [Che94] Harry H. Cheng. Programming with Dual Numbers and its Applications in Mechanisms Design. *Engineering with Computers*, 10(4):212–229, 1994.
- [HZ03] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2003.
- [Nau11] Uwe Naumann. *The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation*. SIAM, Philadelphia, PA, 2011.
- [TMHF99] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle Adjustment – A Modern Synthesis. In *Vision Algorithms*, pages 298–372, 1999.

Zyklizität in empirischen und künstlichen Nahrungsnetzen

Marcus Pöckelmann

Martin-Luther-Universität Halle-Wittenberg

Institut für Informatik

marcus.poeckelmann@gmail.com

Abstract: In der hier in Teilen umrissenen, noch nicht fertig gestellten Masterarbeit stehen gerichtete Kreise in empirischen Nahrungsnetzen im Fokus. Dazu werden drei Maße zum Bestimmen des Grads der Zyklizität vorgestellt und untersucht. Neben dem klassischen *Loop* nutzt das zweite Maß starke Zusammenhangskomponenten. Das dritte Maß beruht auf dem NP-vollständigen Problem der minimalen *Feedback Vertex Sets*, für deren Bestimmung verschiedene Lösungsansätze diskutiert werden. Auf Grundlage der darauf aufbauenden Erkenntnis, dass einige der betrachteten Nahrungsnetze im hohen Maße von gerichteten Kreisen durchsetzt sind, werden Ursachen und Bedeutung der Zyklizität untersucht. Unter anderem steht die noch zu prüfende Hypothese im Raum, dass gerichtete Kreise einen positiven Einfluss auf ein weiteres bekanntes Maß der Netzwerkanalyse - die Stabilität von Nahrungsnetzen - haben. Dieser Zusammenhang ist sowohl im Bereich der Netzwerkanalyse von algorithmischer Relevanz als auch für Biodiversitätsaspekte im Bereich der Ökologie.

1 Einleitung und Datengrundlage

Nahrungsnetze geben Auskunft darüber, welche Spezies miteinander interagieren und lassen so Rückschlüsse auf eine Vielzahl struktureller Eigenschaften zu. In dieser Arbeit werden sie als gerichtete Graphen $D = (V, A)$ aufgefasst. Dabei repräsentieren die Knoten V funktionale Gruppen von Spezies, die so genannten *trophischen Spezies* [WM00], und die gerichteten Kanten A Interaktionen zwischen diesen. Die Interaktion $a_{i,j}$ bedeutet hierbei, dass *Konsument* i die *Ressource* j frisst bzw. in anderer Form (z.B. Parasitismus, Symbiose) mit ihr interagiert. Dabei liegt das Hauptaugenmerk auf der Struktur. Ignoriert werden hier dynamische Aspekte, wie Tages- oder Jahreszeit, Temperatur und Populationsgrößen.

Das Erstellen detaillierter empirischer Datensätze ist ein schwieriges Unterfangen, da sich das Vorkommen vieler Spezies oft erst nach langwierigen Feldstudien nachweisen lässt. Entsprechend umfangreich fallen auch die Zeiträume zum Beobachten der Fressgewohnheiten aus. Trotz des Arbeitsaufwandes stehen als Datenquellen neben der *EcoWeb*¹, auch die geteilten² bzw. öffentlich zugänglichen³ Datensätze zweier Forschungsgruppen zur Verfügung. Diese Arbeit beschränkt sich auf 21 Datensätze, welche besonders detailliert

¹Verfügbar bei der *Rockefeller University* unter <http://hdl.handle.net/10209/306>

²E-Mail-Korrespondenz mit dem *PEaCE Lab*, siehe <http://foodwebs.org>

³Auf der Website des *SEEL* unter <http://people.uncw.edu/borretts/research.html>

oder Grundlage anderer Publikationen sind. Das Nahrungsnetz *El Verde* mit $|V| = 155$ und $|A| = 1509$ zählt bspw. zu den größten Datensätzen. Im Zuge der Untersuchung der Zyklizität wurden ausgehende Kanten von totem organischen Material (Detritus) entfernt, um diese triviale Ursache gerichteter Kreise (im Folgenden kurz “Kreise”) auszuschließen.

2 Zyklizität und Stabilität

Den zentralen Untersuchungsgegenstand stellt die Zyklizität der Nahrungsnetze dar, die als Kenngröße für das Auftreten von Kreisen im Graphen dienen soll. Eine nahe liegende Vermutung ist, dass Nahrungsnetze weitgehend kreisfrei sind. Dennoch treten Kreise in Nahrungsnetzen auf, die bspw. auf Altersunterschieden der sich gegenseitig jagenden Spezies oder Parasitismus beruhen. Zur Bewertung der Zyklizität werden drei Kenngrößen vorgestellt und diskutiert. Neben dem häufig genutzten Maß *Loop* (Anteil der Spezies, die sich in Kreisen befinden), wurde ein *Komponentenmaß* (Anzahl der starken Zusammenhangskomponenten normiert mit $|V|$) untersucht. Das *Komponentenmaß* spiegelt im Vergleich zu *Loop* die Stärke der Azyklizität wider und lässt einige Rückschlüsse auf die Struktur der Kreise zu. Dennoch besteht zwischen beiden ein direkter mathematischer Zusammenhang, was sich in ähnlichen Vor- und Nachteilen äußert. Sie eignen sich gut für Graphen mit vielen kleinen disjunkten Kreisen, haben aber das Problem, dass bereits ein einzelner Kreis mit vielen Knoten, ausschlaggebend für den Wert der Kenngrößen sein kann. Weitere Kreise zwischen diesen Knoten haben dann keinen Einfluss mehr, wie Abbildung 1 veranschaulicht.

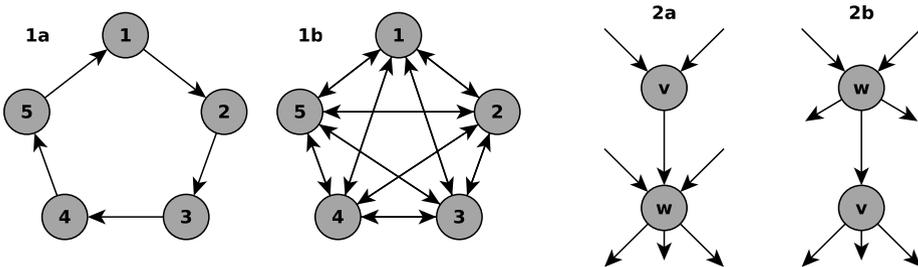


Figure 1: Der einfache Kreis (1a) wird von *Loop* und dem *Komponentenmaß* genauso bewertet, wie ein vollständiger Graph gleicher Größe (1b).

Knoten v besitzt nur eine ausgehende (2a) bzw. eine eingehende (2b) Kante. In beiden Fällen wird v von Knoten w *dominiert*, d.h. alle Kreise, die v beinhalten, beinhalten auch w .

Eine Möglichkeit diesen Strukturen verstärkt Rechnung zu tragen, ist das komplexe Problem die Kreise zu zählen ($\#\text{CYCLE} \in \mathbf{FP} \Rightarrow \mathbf{P} = \mathbf{NP}$) [AB09]. In dieser Arbeit wurde mit den *Feedback Vertex Sets*, kurz *FVS* ein alternatives Vorgehen gewählt. Ein *FVS* ist eine Knotenmenge, durch deren Entfernen ein Graph azyklisch wird. Interessant sind dabei die minimalen *FVS*, die gewissermaßen die Spezies beinhalten, die die größte Rolle

beim Entstehen der Kreise spielen. Das Problem ist NP-vollständig [ENSS95], sodass als Teil dieser Arbeit verschiedene Lösungsansätze diskutiert werden. Ziel ist es, alle minimalen *FVS* eines Nahrungsnetzes zu bestimmen, um zum einen der Zyklizität Ausdruck zu verleihen (Kardinalität der minimalen *FVS* normiert mit $|V|$) und zum anderen über die verursachenden Spezies Rückschlüsse auf die ökologische Bedeutung der Kreise zu ermöglichen. Trotz geringer Größe der Datensätze kommt ein naiver Brute-Force-Ansatz, der alle Kombinationen von Knoten abarbeitet, nicht in Frage. Bei *El Verde* müssten im Worst Case bspw. $\binom{155}{77}$ Kombinationen geprüft werden (geschätzte Laufzeit > 100 Jahre).

Doch es gibt Graphenklassen mit effizienten exakten Lösungsansätzen, die für die vorliegenden Nahrungsnetze überprüft wurden. Zum einen sind die betrachteten Datensätze in der Regel keine planaren Graphen, sodass ein entsprechend effizienter Algorithmus [Hac97] mit linearer Laufzeit hier nicht angewendet werden kann. Zum anderen hat sich gezeigt, dass es sich mit einer Ausnahme nicht um *Reducible Flow Graphs* handelt. Für Instanzen dieser relativ leicht zu bestimmenden Graphenklasse [Tar73], hätte sich ein minimales *FVS* in der polynomiellen Laufzeit von $\mathcal{O}(|A| \cdot |V|^2 \cdot \log(|V|^2/|A|))$ berechnen lassen können [Ram88].

Es gibt eine Vielzahl weiterer Ansätze mit polynomieller Laufzeit, bei denen allerdings keine exakte Lösung des Problems sicher gestellt ist. Die einfache *Big Degree*-Heuristik [FWY09] bspw., bei der stets der Knoten mit den meisten aus- und eingehenden Kanten entfernt wird, benötigt für *El Verde* 22 Knoten bis zur Kreisfreiheit. Daneben wurden eine Heuristik [ELS93] mit der Laufzeit $\mathcal{O}(|A|)$ sowie verschiedene Approximationsalgorithmen [ENSS95] für diese Arbeit in Betracht gezogen.

Doch mit Hilfe einiger Reduktionsregeln, welche sich z.T. bereits in der Literatur finden [FWY09], kann die Problemgröße deutlich verringert werden. Zwei erwiesen sich hier als besonders erfolgreich. Zum einen die Beschränkung auf die starken Zusammenhangskomponenten, da die übrigen Knoten per Definition kein Bestandteil von Kreisen sein können, und zum anderen das Ausschließen *dominierter* Knoten aus der eigentlichen Berechnung. Wie in Abbildung 1 veranschaulicht, besitzt ein *dominierter* Knoten v nur eine aus- bzw. eingehende Kante, sodass alle Kreise, die v beinhalten, auch stets den selben adjazenten Knoten w beinhalten. So kann der Knoten v bei der Berechnungen der minimalen *FVS* durch w repräsentiert und der Graph so “geschrumpft” werden. Für die resultierenden *FVS* wird anschließend geprüft, ob w wieder durch v ersetzt werden kann, um wirklich alle minimalen *FVS* zu erhalten. Nach Anwendung der Reduktionsregeln war die Problemgröße der betrachteten Nahrungsnetze stets klein genug, um alle minimalen *FVS* exakt zu bestimmen (bspw. konnte *El Verde* mit $\binom{58}{11}$ zu prüfenden Kombinationen in unter 23 Std. berechnet werden) und den Fokus auf die Analyse der Ergebnisse zu legen.

Die bisherigen Berechnungen haben gezeigt, dass Kreise in Nahrungsnetzen keine Seltenheit sind, sondern im Gegenteil große Strukturen, wie starke Zusammenhangskomponenten mit 70 Knoten, bilden können. In diesem Kontext ist der Einfluss der Kreise auf die Stabilität von Nahrungsnetzen ein wichtiger Untersuchungsgegenstand. Ein geeignetes Stabilitätsmaß ist dabei die *strukturelle Robustheit* [DWM02], bei welcher die Auswirkungen des Entfernens von Spezies, sogenannter *primary extinctions* (pe), auf andere Spezies bestimmt werden. Verliert eine Spezies auf diese Weise all ihre Ressourcen, findet eine sogenannte *secondary extinction* (se) statt und sie wird ebenfalls entfernt, was

wiederum zu weiteren *se* führen kann. Dabei wurden die zu entfernenden Spezies in der Originalarbeit nach den drei Kriterien (1) höchster Knotengrad, (2) zufällig und (3) niedrigster Knotengrad gewählt und die Auswirkungen getrennt voneinander untersucht. Nun lassen sich die Spezies der minimalen *FVS* als viertes Kriterium ergänzen. Die noch zu prüfende Hypothese lautet dabei, dass dieses Kriterium zu signifikant häufigeren *se* führt als die übrigen und somit der Einfluss der Kreise auf die Stabilität der Nahrungsnetze belegt werden kann. Auch ganze Nahrungsnetze können, wie in der Originalarbeit geschehen, miteinander verglichen werden, indem die mit $|V|$ normierte Anzahl an *pe* bestimmt wird, die zu einem Gesamtspeziesverlust von über 50% führt. Entsprechend lässt sich die strukturelle Robustheit von stark zyklischen und azyklischen Netzen miteinander vergleichen und, so die Hypothese, die Bedeutung der Kreise auf die Stabilität untermauern. Ferner ist die Bestimmung der minimalen Knotenmenge von Interesse, deren Entfernung den genannten Verlust von 50% auslöst. Angenommen wird, dass es eine große Schnittmenge zwischen dieser und den minimalen *FVS* gibt.

Ein weiterer Aspekt der Arbeit, der an dieser Stelle nur erwähnt wird, ist die Validierung des häufig verwendeten *Nischen-Modells* zum Erzeugen künstlicher Nahrungsnetze von Martinez und Williams [WM00] bezogen auf die vorgestellten Zyklizitätsmaße.

References

- [AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.
- [DWM02] Jennifer A. Dunne, Richard J. Williams, and Neo D. Martinez. Network structure and biodiversity loss in food webs: robustness increases with connectance. *Ecology Letters*, 5(4):558–567, 2002.
- [ELS93] Peter Eades, Xuemin Lin, and W. F. Smyth. A fast and effective heuristic for the feedback arc set problem. *Inf. Process. Lett.*, 47(6):319–323, October 1993.
- [ENSS95] Guy Even, Joseph (Seffi) Naor, Baruch Schieber, and Madhu Sudan. Approximating Minimum Feedback Sets and Multi-Cuts in Directed Graphs. In *Proceedings of the 4th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 14–28, London, UK, UK, 1995. Springer-Verlag.
- [FWY09] Rudolf Fleischer, Xi Wu, and Liwei Yuan. Experimental Study of FPT Algorithms for the Directed Feedback Vertex Set Problem. In *Algorithms - ESA 2009*, volume 5757, pages 611–622. Springer Berlin Heidelberg, 2009.
- [Hac97] W. Hackbusch. On the feedback vertex set problem for a planar graph. *Computing*, 58(2):129–155, 1997.
- [Ram88] Vijaya Ramachandran. Finding a minimum feedback arc set in reducible flow graphs. *J. Algorithms*, 9(3):299–313, 1988.
- [Tar73] Robert Tarjan. Testing flow graph reducibility. In *Proceedings of the fifth annual ACM symposium on Theory of computing*, STOC '73, pages 96–107, New York, NY, USA, 1973. ACM.
- [WM00] Richard J. Williams and Neo D. Martinez. Simple rules yield complex food webs. *Nature*, 404(6774):180–183, March 2000.

Sampling perfekter Matchings in Bezug auf die Struktur des unterliegenden Graphen

Steffen Rechner

Martin-Luther-Universität Halle-Wittenberg
Institut für Informatik
steffen.rechner@gmail.com

Abstract: In dieser Arbeit sollen Markov Chain Monte Carlo Verfahren zur approximativen Berechnung der Anzahl perfekter Matchings in bipartiten Graphen untersucht und implementiert werden. Ziel der Arbeit ist eine empirische Analyse der Mischzeiten der Verfahren in Abhängigkeit verschiedener struktureller Eigenschaften des unterliegenden Graphen. Eine theoretische Analyse sowie der Vergleich mit experimentell ermittelten Mischzeiten sollen Aufschlüsse über die praktische Verwendbarkeit der Methoden liefern sowie zu möglichen Verbesserungen bekannter oberer Schranken führen.

1 Einleitung

Die Berechnung der Anzahl perfekter Matchings eines bipartiten Graphen $G = (V, E)$ stellt ein Problem dar, das im allgemeinen schwer ist. Leslie Valiant hat 1979 gezeigt, dass dieses Problem *#P-vollständig* ist und daher wahrscheinlich nicht in polynomieller Zeit zu lösen ist [Val79]. Tatsächlich besitzt der beste bekannte exakte Algorithmus, der 1963 von Herbert J. Ryser vorgestellt wurde [Rys63], eine asymptotische Laufzeit von $\Theta(|V| \cdot 2^{|V|})$ und ist damit bereits für kleine Graphen praktisch nicht mehr anwendbar.

Ein naheliegender Schritt ist daher der Versuch einer Approximation des Problems. Ein von Mark Jerrum und Alistair Sinclair im Jahr 1989 veröffentlichtes Verfahren beschreibt ein Approximationsschema, mit welchem die Anzahl der perfekten Matchings mit beliebiger Genauigkeit geschätzt werden kann [JS89]. Jerrum und Sinclair haben zudem bewiesen, dass die Laufzeit dieses Algorithmus für einige Graphklassen polynomiell in der Eingabegröße sowie des gewünschten Fehlers beschränkt ist und damit für diese Klassen ein Polynomialzeit-Approximationsschema gefunden. Jedoch wurde ebenfalls gezeigt, dass das Verfahren für andere Klassen eine exponentielle Laufzeit aufweist.

Es blieb noch für einige Jahre ungewiss, ob ein Polynomialzeit-Approximationsschema für den allgemeinen Fall existiert. Erst im Jahr 2004 publizierten Jerrum, Sinclair und Vigoda eine Weiterentwicklung des 1989 vorgestellten Verfahrens, mit dem sich die Anzahl der perfekten Matchings in allen bipartiten Graphen approximieren lässt und erhielten dafür im Jahre 2006 den Fulkerson-Preis [JSV04].

2 Sampling perfekter Matchings

Grundlage und Kernstück der 1989 und 2004 vorgestellten Verfahren bildet ein Sampler für perfekte Matchings. Damit wird ein randomisierter Algorithmus bezeichnet, der zu einem gegebenen bipartiten Graphen einen zufälligen Vertreter aus der Menge aller perfekter Matchings dieses Graphen konstruiert. Sollte der Samplingprozess in polynomieller Zeit funktionieren, dann kann auch die Anzahl der perfekten Matchings in Polynomialzeit approximiert werden.

Der Hauptanteil der Arbeiten von Jerrum und Sinclair [JS89] sowie von Jerrum, Sinclair und Vigoda [JSV04] ist daher die Konstruktion eines solchen Samplingverfahrens. Ein für diese Zwecke oftmals verwendetes Konzept ist das Markov Chain Monte Carlo Verfahren. Dieses Prinzip basiert auf der Konstruktion und der Entwicklung ergodischer *Markov-Ketten*, deren Zustandsraum aus der Menge der zu generierenden Objekte besteht, in diesem Fall also aus der Menge aller perfekten Matchings. Um eine ergodische *Markov-Kette* zu erhalten werden dem Zustandsraum zudem alle fast-perfekten Matchings hinzugefügt. Dies sind Matchings, in denen genau zwei Knoten ungepaart bleiben. Man erhält dadurch einen Sampler für perfekte und fast-perfekte Matchings. Der Anteil der fast-perfekten Matchings muss durch die Einführung von Gewichten $w(M)$ ausgeglichen werden. Mit den nachfolgend beschriebenen Zustandsübergängen definieren Jerrum, Sinclair und Vigoda eine *Markov-Kette* zum Samplen perfekter und fast-perfekter Matchings:

Ausgehend von einem Matching M werden folgende Transitionen definiert:

1. Falls M ein perfektes Matching ist: Wähle eine Kante $e \in M$ und entferne sie. Man erhält ein fast-perfektes Matching M' .
2. Falls M ein fast-perfektes Matching ist, in dem die Knoten u und v ungepaart sind: Wähle einen Knoten z zufällig und gleichverteilt aus der Menge aller Knoten V .
 - (a) Falls $z = u$ oder $z = v$ und eine Kante $\{u, v\} \in E$ existiert: Füge die Kante $\{u, v\}$ zu M hinzu. Man erhält ein perfektes Matching M' .
 - (b) Falls eine Kante $\{u, z\} \in E$ existiert, z jedoch bereits zu einer Matchingkante $\{x, z\} \in M$ gehört: Entferne die Kante $\{x, z\}$ aus M und füge die Kante $\{u, z\}$ hinzu. Man erhält ein fast-perfektes Matching M' .
 - (c) In allen anderen Fällen setze $M' \leftarrow M$.
3. Mit einer Wahrscheinlichkeit von $\min\{1, w(M')/w(M)\}$ geht die *Markov-Kette* in den Zustand M' über, sonst verbleibt sie im Zustand M .

Es lässt sich zeigen, dass diese *Markov-Kette* zu einer Verteilung $\pi(M)$ konvergiert, die sich proportional zu den Gewichten $w(M)$ verhält [JSV04]. Das zufällige Durchlaufen des Zustandsraumes liefert daher nach genügend vielen Schritten ein Matching, dessen Wahrscheinlichkeit zu seinem Gewicht korrespondiert, wodurch sich jede gewünschte Verteilung ansteuern lässt. Um den Anteil der fast-perfekten Matchings auszugleichen, werden die Gewichte daher so gewählt, dass der Menge der perfekten Matchings eine ausreichend hohe Wahrscheinlichkeit zugewiesen wird.

3 Mischzeitabschätzungen

Jerrum, Sinclair und Vigoda zeigten in ihrer Arbeit [JSV04], dass die Anzahl an Schritten, die die *Markov-Kette* durchlaufen muss, um sich einer gewünschten Verteilung bis auf ein ϵ anzunähern, polynomiell in der Größe $|V|$ des zugrundeliegenden bipartiten Graphen sowie der Fehlertoleranz ϵ^{-1} beschränkt ist. Die Mischzeit wird dort mit einer oberen Schranke von $\mathcal{O}(|V|^{10} \cdot (\log(\pi_{min}^{-1} + \log \epsilon^{-1})))$ abgeschätzt [JSV04, Theorem 3.1 + Lemma 4.1]. Tatsächlich ist die Mischzeit sogar noch etwas höher, da die Gewichte $w(M)$ im Vorfeld approximiert werden müssen. In ihrem Beweis geht es den Autoren in erster Linie um einen nachvollziehbaren Beweis der Polynomialzeitbeschränkung und weniger um möglichst niedrige obere Schranken. Aus diesem Grund werden an einigen Stellen recht großzügige Abschätzungen getroffen, wodurch die gewonnene obere Schranke die reelle Mischzeit in der Praxis oft um ein Vielfaches übersteigt. Da diese die Laufzeit des Samplers und damit die Laufzeit des Approximationsschemas dominiert, würde eine schärfere Schranke die Laufzeit des Algorithmus deutlich verbessern.

Alistair Sinclair liefert in seiner Arbeit von 1992 einen Überblick über die gebräuchlichsten Methoden zur Abschätzung von Mischzeiten [Sin92]. Die dort vorgestellten Methoden nutzen verschiedene Eigenschaften der Markov-Kette aus und liefern unterschiedlich gute Abschätzungen. Einen Ansatz liefert der wohlbekannt Zusammenhang zwischen dem *Spectral Gap* einer *Markov-Kette* und deren Mischzeit. Viele Verfahren beruhen daher auf der Abschätzung der Eigenwerte der Übergangsmatrix einer *Markov-Kette*. Im Rahmen dieser Arbeit werden einige dieser Methoden implementiert sowie experimentell miteinander verglichen. Abbildung 1 zeigt einen Vergleich zwischen der experimentell bestimmten, tatsächlichen Mischzeit und verschiedenen Abschätzungsmethoden eines beispielhaften Graphen.

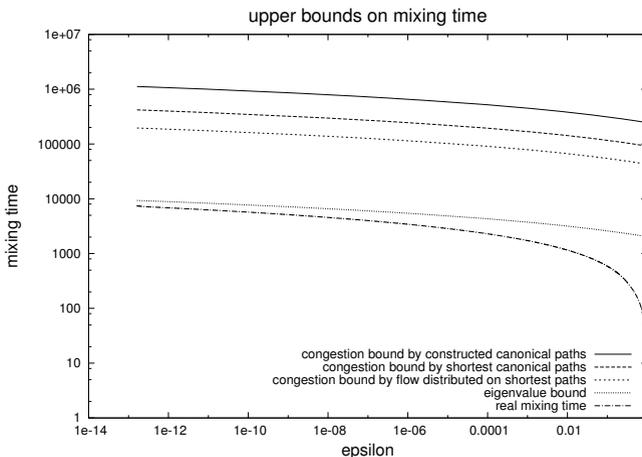


Abbildung 1: Mischzeit (untere Kurve) und obere Schranken eines Graphen mit $|V| = |E| = 28$

Für ein beliebiges $\epsilon \in [0, 1]$ zeigt die Grafik die notwendige Anzahl von Schritten, um sich der Wahrscheinlichkeitsverteilung π bis auf ϵ anzunähern. Die untere Kurve zeigt die experimentell bestimmte, tatsächliche Mischzeit des Graphen, während die anderen Kurven verschiedene Methoden zu deren Abschätzung zeigen. Die oberste Kurve repräsentiert etwa die Methode, welche in [JS89] und [JSV04] benutzt wird, um die Polynomialzeitbeschränkung des Samplers zu beweisen. Es ist deutlich zu erkennen, dass diese im Vergleich zu der tatsächlichen Mischzeit entschieden größer ist und daher durch die Anwendung raffinierterer Methoden noch Potenzial für bessere Abschätzung vorhanden ist.

Ein weiterer Aspekt dieser Arbeit ist die Untersuchung des Zusammenhangs zwischen der Mischzeit einer *Markov-Kette* und der Struktur des darunterliegenden bipartiten Graphen. Unter anderem wird bestimmt, wie sich die Mischzeit in Verhältnis zur Dichte des bipartiten Graphen verhält. Neben dieser Eigenschaft werden verschiedene weitere Eigenschaften untersucht und in Zusammenhang zur Mischzeit gesetzt. Aufbauend auf dieser Analyse werden verschiedene Graphklassen vorgestellt sowie untersucht.

4 Zusammenfassung und Ausblick

In dieser Arbeit werden Markov Chain Monte Carlo Verfahren analysiert und implementiert, mit welchen die Anzahl der perfekten Matchings in bipartiten Graphen beliebig genau approximativ werden kann. Anhand verschiedener Graphklassen unterschiedlicher Struktur wird eine Reihe verschiedenartiger Abschätzungsmethoden verwendet, um die tatsächliche Mischzeiten nach oben hin zu beschränken. Dabei liefern verschiedene Methoden unterschiedlich gute Ergebnisse. Eine tieferegehende theoretische Analyse erscheint daher lohnenswert und bietet Raum für weitere Forschung.

Literatur

- [JS89] M. Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, December 1989.
- [JSV04] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A Polynomial-Time Approximation Algorithm for the Permanent of a Matrix with Non-Negative Entries. *Journal of the ACM*, pages 671–697, 2004.
- [Rys63] Herbert J. Ryser. *Combinatorial mathematics*. Carus mathematical monographs. Mathematical Association of America; distributed by Wiley [New York, 1963.
- [Sin92] Alistair Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. In Imre Simon, editor, *LATIN '92*, volume 583 of *Lecture Notes in Computer Science*, pages 474–487. Springer Berlin Heidelberg, 1992.
- [Val79] Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

Indikatoren gestützte Betrugsprognose anhand von Jahresabschluss- und Buchführungsdaten in einem probabilistischen wissensbasierten System

Aykut Bußian

Fernuniversität Hagen, Wissensbasierte Systeme (Student)
aykut.bussian@t-online.de

Art der Arbeit: Bachelorarbeit Informatik; Betreuer: Prof. Dr. Christoph Beierle

Abstract: Die Erkennung betrieblichen Betruges basiert regelmäßig auf unvollständigem Wissen des Benutzers; zudem sehen die bisherigen Prüfprogramme oftmals ein tabellarisches, standardisiertes Vorgehen zur Betrugsbeurteilung vor, ohne die unterschiedliche Erfahrung der Benutzer transparent zu machen. Probabilistische Expertensysteme bieten eine geeignete Sprachsemantik zur Darstellung unvollständigen Wissens und ermöglichen durch Angabe von Wahrscheinlichkeiten die Abbildung unterschiedlichen Expertenwissens. Mittels der Expertensystem-Shell SPIRIT wird im Rahmen dieser Arbeit die probabilistische Wissensbasis FRAUD DETECTION konzipiert, die sowohl zur Betrugsprognose als auch –diagnose eingesetzt werden kann. Als weiteres Einsatzgebiet wäre insbesondere die Insolvenzfrüherkennung denkbar.

1 Einleitung

Die Schwierigkeiten der Erkennung betrieblichen Betrugs („Fraud“) liegen regelmäßig darin, dass die Prozesse und Strukturen des Unternehmens sowohl Folge wirtschaftlicher Gegebenheiten (Branchentyp, Größe des Unternehmens) sein können als auch Ausdruck einer erhöhten Betrugsanfälligkeit. Zudem sehen die gängigen Prüfprogramme meist katalogartige Aufzählungen von Indikatoren vor, die entweder vorbeugend („Red Flag“) oder nachsorgend („Kennzahlen“) das mögliche/realisierte Betrugsrisiko erfassen wollen, ohne die Indikatoren im Hinblick auf ihre Relevanz bzw. Intensität zu quantifizieren [Co09, Ra86]. Probabilistische Expertensysteme zeichnen sich durch eine reichhaltige Sprachsemantik aus, mit der u.a. Konditionalausdrücke der Form „Wenn die Geschwindigkeit des Fahrzeugs sehr hoch ist oder sie ist hoch und der Fahrer ist übermüdet, dann besteht ein erhöhtes Unfallrisiko“. Zudem können die Konditionalausdrücke (Regeln, bestehend aus Prämissen und einer Konklusion) sowie die darüber hinaus möglichen Fakten (Variablen oder logische Verknüpfungen von Variablen) jeweils mit einer Wahrscheinlichkeit α (Schlussstärke) belegt werden, welche sowohl aus statistischen Daten gewonnen werden können oder auch aus den Einschätzungen des Experten/ Benutzers. SPIRIT als probabilistisches Expertensystem verwendet zudem die maximale Entropie zur Inferenz, wodurch fehlende Wahrscheinlichkeiten in informationstheoretisch optimaler Weise ergänzt werden [Ah05, Ro97, Ro96].

Vor diesem Hintergrund ist der Gegenstand dieser Arbeit eine Fallstudie zur Eignung probabilistischer Expertensysteme zur Prognose und Diagnose des betrieblichen Betrugsrisikos. Hierzu wurde mittels der Expertensystem-Shell SPIRIT die Wissensbasis FRAUD DETECTION erstellt, deren Regeln und Fakten aus der betriebswirtschaftlichen Fraud-Literatur [AC12, Ho08, ID11, Lu07, Sc12, We10, We11] gewonnen wurden und deren Schlussstärke der Anfangseinschätzung des Autors entstammen und im Fortgang der Arbeit variiert wurden. Insgesamt sollen folgende Hypothesen überprüft werden:

- das Domänenwissen der Betrugserkennung ist durch probabilistische Konditionale und Fakten darstellbar,
- unterschiedliche Erfahrungsgrade des Experten bzw. der Relevanz der Konditionale können durch unterschiedliche Schlussstärken berücksichtigt werden,
- durch probabilistische Expertensysteme können Branchen- und Betriebsspezifika erfasst werden.

Die Ergebnisse der Arbeit können Aufschluss geben über die grundsätzliche Einsatzfähigkeit probabilistischer Expertensysteme bei einer Vielzahl betriebswirtschaftlicher und volkswirtschaftlicher Fragestellungen, in denen –abseits des üblichen modellbasierten Vorgehens- eine empirische, auf Expertenschätzungen und statistischen Daten basierende Problemeinschätzung und –lösung angestrebt wird.

Im folgenden Abschnitt wird zunächst auf die Modellierung von gängigen Verfahren der Betrugserkennung durch probabilistische Regeln eingegangen, daran anschließend erfolgt die Beurteilung der Regelmenge zur Prognose („wenn die Regeln X,Y und Z sowie die Fakten 1-5 vorliegen, wie hoch ist dann das Risiko für Warendiebstahl?“) und zur Diagnose („Es waren Warendiebstahle zu verzeichnen. Auf welche betrieblichen Faktoren kann dies zurückzuführen sein?“).

2 Betrugserkennung: Wissensmodellierung und -auswertung

Die Wissensdomäne wird abgegrenzt durch die zu erfassenden Fraud-Tatbestände (hier: Vermögensdelikte und Rechnungslegungsdelikte in verschiedenen Varianten) sowie die Indikatoren und Kennzahlen, die gemäß der gängigen Fraud-Literatur zur Betrugserkennung verwendet werden.

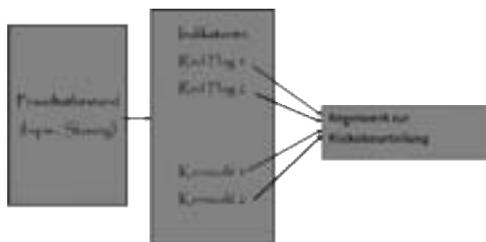


Abbildung 1: Konzept der Wissensbasis

Ein Fraud-Tatbestand wie „Skimming“ (=Gelddiebstahl an der Verkaufsstelle durch den Verkäufer) wird in folgenden Schritten in die probabilistische Wissensbasis überführt: 1.Auswahl geeigneter Indikatoren, 2.Überführung der Indikatoren in Variablen, 3.Formulierung der funktionalen Zusammenhänge in probabilistischen Regeln. Insgesamt wurden in FRAUD DETECTION neun Fraud-Tatbestände abgebildet, welche mittels 56 Regeln und 18 Variablen prognostiziert bzw. diagnostiziert werden. Dabei kann ein Fraud-Tatbestand die Ausprägung „high“, „middle“ oder „low“ aufweisen. Das Vorgehen der Regelbildung wird am Beispiel des Skimming-Risikos illustriert.

1.In Variablen überführte Indikatoren

Bezeichnung	Typ	Ausprägungen
SkimmingR	Nominal	high/middle/low: Intensität des Betrugsrisikos.
RevenueDev ¹	Boolesch	True, sofern die Umsatzerlöse gestiegen sind.
SegCashAcc ²	Boolesch	True, sofern die Kassenführung und die Buchhaltung von verschiedenen Mitarbeitern wahrgenommen werden.
SegAccWar ³	Boolesch	True, sofern die Buchhaltung und die Lagerführung von verschiedenen Mitarbeitern wahrgenommen werden.

2.Entwicklung einer probabilistischen Regel:

$$[0.8] (SkimmingR = high \mid \neg SegAccWar \wedge \neg SegCashAcc)$$

Es liegt ein hohes Skimming-Risiko vor, wenn keine Funktionstrennung zwischen Kasse, Buchhaltung und Lagerhaltung vorliegt. In diesem Fall kann der Mitarbeiter das Skimming weitgehend verschleiern, indem er der Kasse Bargeld entnimmt und den Warenstand in der Lagerhaltung fälscht, und somit vermeidet, dass die Verkaufstransaktion überhaupt von der Buchführung erfasst wird. Allerdings kann die geringe Funktionstrennung z.B. schlicht auf Personalmangel oder auf Effizienzgründe zurück zu führen sein, daher wird die Regel mit einer Schlussstärke von „0,8“ versehen, statt mit „1,0“. Um sowohl die Flexibilität als auch die Aussagekraft der ermittelten Betrugswahrscheinlichkeiten durch FRAUD DETECTION zu beurteilen, wurden anschließend mehrere Szenarien definiert, welche die Wissensbasis an die Besonderheiten von Unternehmen des Handelssektors, des produzierenden Gewerbes und der Bankwirtschaft anpassen. Darüber hinaus wurden Szenarien für kleine Unternehmen und Unternehmen in einer wirtschaftlichen Krise entwickelt. Die Würdigung der Wissensauswertung erfolgte in folgenden Schritten:

¹ RevenueDev = Revenue Development (Umsatzentwicklung).

² SegCashAcc = Segregation of Cash and Accounting.

³ SegCashAccWar= Segregation of Accounting and Warehousing.

Zunächst wurde je eine Hypothese über das zu erwartende Betrugsrisiko pro Betrugstatbestand formuliert („das Skimming-Risiko bei einer Bank ist gering“, „das Risiko gefälschter Rechnungen bei Unternehmen in Krisensituationen ist hoch“ etc.), anschließend wurden diese mit den tatsächlichen Systemauswertungen bei der Betrugsprognose und –diagnose verglichen und auf Abweichungen hin untersucht.

3 Ergebnisse und Ausblick

Sowohl hinsichtlich der Wissensrepräsentation als auch der –auswertung konnte FRAUD DETECTION weitgehend die in den obigen Hypothesen formulierten Anforderungen erfüllen. Die wesentlichen Vorteile gegenüber tabellarisch formulierten Prüfprogrammen bestehen in der Berücksichtigung des Expertenwissens verschiedener Benutzer oder des veränderten Erfahrungswissens desselben Benutzers (Anpassung von Regeln/ Fakten/ Schlussstärken), in den flexiblen Auswertungsmöglichkeiten durch die Definition von Szenarien und in der Messbarkeit des Lernprozesses durch die von SPIRIT ermittelte absolute Entropie. Der Entropiewert kann damit auch als Substitut der „Erfahrungsreife“ der Wissensbasis herangezogen werden (je kleiner, desto höher ist die Informationsdichte bzw. geringer ist die verbliebene Unsicherheit, [Ro96]). Ein weiterer Vorteil des Expertensystems liegt in der Möglichkeit der Berücksichtigung statistischer Daten, die auch aus der laufenden Buchhaltung gewonnen werden können. Als weiterer Schritt ist denkbar, den Einsatzbereich auf vergleichbare ökonomische Problemfelder mit unvollständigem Wissen zu erweitern; naheliegend wäre die Insolvenzprognose, die vielfältige Schnittstellen zur Betrugsprognose aufweist.

Literaturverzeichnis

- [AC12] ACFE, Report To The Nations On Occupational Fraud And Abuse; Austin (USA): ACFE (2012).
- [Ah05] Ahuja, André: Projektrisikomanagement mit Hilfe prob. Wissensv., Berlin: Logos Verlag (2005).
- [Co09] Coderre, David; Computer Aided Fraud Prevention and Detection, New Jersey (USA): John Wiley& Sons (2009).
- [Ho08] Hofmann, Stefan; Handbuch Anti-Fraud-Management, Berlin: Erich Schmidt Verlag (2008).
- [ID11] IDW; IDW Prüfungsstandards Band I, IDW PS 210: Zur Aufdeckung von Unregelmäßigkeiten im Rahmen der Abschlussprüfung; Düsseldorf: IDW Verlag (2011).
- [Lu07] Lüdenbach, Norbert/ Hoffmann, Wolf Dieter; Haufe IFRS-Kommentar, Freiburg: Rudolf Haufe Verlag (2007)
- [Ra86] Rauch-Hindin, W.B., Artificial intelligence in business, science, and industry, New Jersey (USA): Prentice Hall (1986).
- [Ro97] Rödder, Wilhelm/ Kern-Isberner, Gabriele; Léa Sombé und entropie-optimale Informationsverarbeitung mit der Expertensystem-Shell SPIRIT, OR Spektrum, (1997) Ausgabe 19.
- [Ro96] Rödder, Wilhelm/ Meyer, Carl-Heinz; Coherent Knowledge Processing at Maximum Entropy by Spirit. Proc. UAI 1996, pp. 470- 476, Morgan Kaufmann, 1996.
- [Sc12] Schiel, Andreas: Risikobeurteilung von Bilanzmanipulationen, Dortmund: Gabler Verlag (2012).
- [We11] Wells, Joseph T.; Corporate Fraud Handbook, Austin (USA): John Wiley& Sons (2011).
- [We10] Wells, Joseph T.; Principles of Fraud Examination, Austin (USA): John Wiley& Sons (2010).

Echtzeiterkennung von Umgebungsgeräuschen mit Android

Dominik Kreuz, Matthias Mielke, Rainer Brück

Universität Siegen
Lehrstuhl Mikrosystementwurf
dominik.kreuz@student.uni-siegen.de

Art der Arbeit: Bachelorarbeit

Abstract: In diesem Beitrag wird eine Smartphone-Applikation vorgestellt, die in der Lage ist Umgebungsgeräusche in Echtzeit zu erkennen. Dabei dient das Erkennen von Martinshörnern als Referenzbeispiel. Umgesetzt wird diese Art der Mustererkennung mit Hilfe von Mel Frequency Cepstral Coefficients (kurz: MFCCs) als Merkmalsbeschreibung und einem neuronalen Netz als Klassifikator.

1 Motivation

Für viele hörgeschädigte Menschen stellt die Teilnahme am Straßenverkehr ein Problem dar. In der Studie von Marion Hersh wurde eine von 45 gehörlosen Personen beantwortete Befragung ausgewertet. [1] Die Auswertung hat ergeben, dass sich nahezu alle Befragten durch die Hörbeeinträchtigung weniger sicher fühlen. Sie haben Probleme den Straßenverkehr wahrzunehmen, selbst herannahende Busse und LKWs werden nicht bemerkt.

Um diese Menschen im Straßenverkehr unterstützen zu können, wurde eine Applikation für mobile Geräte mit dem Betriebssystem Android entwickelt. Mit Hilfe von Verfahren der Mustererkennung soll eine Erkennung von Umgebungsgeräuschen ermöglicht werden.

2 Mustererkennung

Der Ablauf einer Mustererkennung wird in Abbildung 1 gezeigt. Die Aufnahme wandelt die analogen Signale des Mikrofons in digitale Werte um. Die Vorverarbeitung filtert und segmentiert die Daten, um eine bessere Datenkonsistenz zu erhalten. Anschließend werden Merkmale, die die Audiodatei beschreiben, berechnet. Die Merkmalsreduktion versucht die am besten geeigneten Werte zu identifizieren, die dann im Klassifikator den einzelnen Klassen zugeordnet werden. All diese Schritte zusammen bilden die Arbeitsphase. In der vorausgehenden Lernphase werden den Merkmalen Stichproben entnommen und mit ihrer Hilfe die Klassengebiete angepasst.

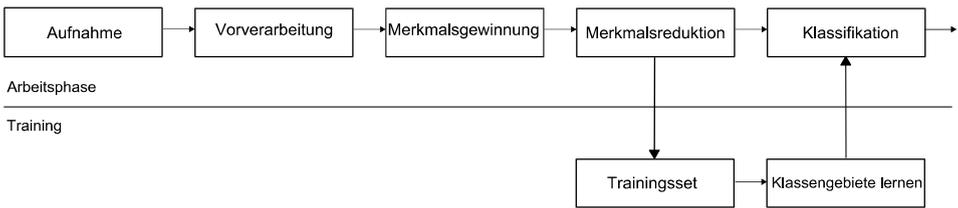


Abbildung 1: Ablauf einer Mustererkennung

2.1 Merkmalsgewinnung

Als Merkmale des Audiosignals werden MFCCs verwendet. Sie bilden eine kompakte Darstellung des Frequenzspektrums eines Signals. Mel steht dabei für eine Maßeinheit, welche die wahrgenommene Tonhöhe kompakt abbildet. MFCCs liefern besonders gute Ergebnisse für die Merkmalsgewinnung im Bereich der Spracherkennung. [2] Auch in der Umgebungsgeräuscherkennung können gute Ergebnisse erzielt werden. [3] Abbildung 2 stellt den Ablauf der Berechnung dar.

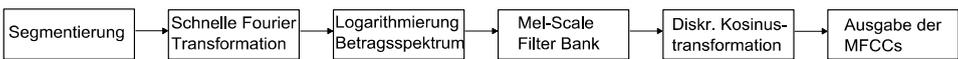


Abbildung 2: Ablauf einer Mustererkennung

2.2 Klassifikation

Als Klassifikator werden künstliche neuronale Netze benutzt. Künstliche neuronale Netze sind informationsverarbeitende Systeme, die dem menschlichen Gehirn in ihrer Arbeitsweise sehr ähnlich sind und strukturell aus Neuronen bestehen. Sie bilden eine Recheneinheit, die aktiviert wird, Daten verarbeitet und an die Umwelt weiterleitet. Die Gesamtheit der einzelnen Neuronen bildet ein neuronales Netz dessen Informationen in Form von elektrischen Signalen über Verbindungen übertragen werden. [4]

Das Besondere an einem neuronalen Netz ist, dass es trainiert und nicht programmiert wird. Seine Eigenschaften sind also nicht von vornherein vorgegeben, sondern es passt sich erst auf Grund von Trainingselementen an die jeweiligen Bedürfnisse an.

3 Implementierung und Ergebnisse

Die Applikation ist frameworkartig aufgebaut, somit ist die Wiederverwendbarkeit und die Möglichkeit zur Weiterentwicklung der Software gewährleistet. Die realisierte Applikation eröffnet die Möglichkeit im Training Audiodaten aus einer WAVE-Datei oder in der Arbeitsphase direkt vom Mikrofon einzulesen und zu verarbeiten. Die Daten können auch gefiltert und segmentiert werden.

Um feststellen zu können, ob die Applikation den Anforderungen der Echtzeiterkennung, hier die Erkennung innerhalb einer halben Sekunde, gerecht wird, wurde ein zweifaches Profiling durchgeführt.

Zunächst wurde die Systemzeit vor dem Start und nach Ablauf einer gewissen Anzahl an Durchläufen der Klassifizierung festgehalten. Auf diese Weise kann die Anzahl der Berechnungen pro Sekunde bestimmt werden. In diesem Fall liegt sie bei etwa 31, damit ist die Echtzeitfähigkeit gewährleistet.

Zum anderen wurde das von Android zur Verfügung gestellte Profiling-Programm "Traceview" benutzt. [5] Hier werden die aufgerufenen Methoden der zu ihrer Berechnung benötigten Zeit gegenübergestellt, wobei für jeden Prozess der prozentuale Anteil an der Gesamtberechnungszeit angegeben wird. Dabei wird zwischen CPU-Zeit und Real-Zeit unterschieden. Dies ist nötig da die CPU nicht ausschließlich für die Berechnungen dieser Applikation eingesetzt wird, sondern vielmehr alle laufenden Prozesse verwaltet. Wie Tabelle 1 zeigt, verbraucht die Ermittlung der MFCCs (MFCC.cepstrum) die meiste Zeit (64%). Die Berechnung des neuronalen Netzwerkes benötigt 13,7% (NeuralNetwork.calculate), dies beinhaltet die Berechnung der einzelnen Layer mit 13,3% (Layer.calculate) und die der Neuronen mit 10,8% (Neuron.calculate). Die Methode „Math.pow“, die eine gegebene Zahl potenziert, verbraucht ca. 20% der Zeit.

Prozess	CPU Time %	CPU Time	Real Time %	Real Time
MFCC.cepstrum	64,30%	1453,3	48,60%	1557,6
Math.pow	20,40%	460,9	15,80%	507,1
NeuralNetwork.calculate	13,60%	308,4	10,10%	324,3
Layer.calculate	13,20%	298,1	9,80%	313,2
Neuron.calculate	10,60%	240,7	7,90%	253,1

Tabelle 1: Traceview eines Galaxy Nexus S

Weitere im Rahmen dieser Arbeit durchgeführte Tests belegen, dass die Applikation schon jetzt unter Laborbedingungen sehr akzeptable Ergebnisse erzielt. Angewandt auf ein Testset von 6 Minuten und 11 Sekunden mit dem Signal eines Martinshorns und 6 Minuten 11 Sekunden mit dem von Straßengeräuschen, lag die Erkennungsrate für Martinshörnern bei über 70%, die Nichterkennungsrate von Straßengeräuschen bei 100%.

4 Ausblick

Bereits jetzt wird das Echtzeitkriterium erreicht und die Erkennungsraten unter Laborbedingungen sind annehmbar. Trotzdem ließen sich sicherlich noch wesentlich bessere Erkennungsraten erzielen. Eine erste naheliegende Möglichkeit wäre die Verbesserung und Vergrößerung des Trainingssets, eine weitere die Optimierung des neuronalen Netzes (Architektur, Wahl der Parameter). Eine dritte Option wäre es andere Merkmale aus den Audiodaten zu extrahieren. Ebenso wäre es wünschenswert andere Methoden der Klassifikation umzusetzen und diese in Kombination mit den verschiedenen Merkmalen zu testen.

Literaturverzeichnis

- [1] MARION HERSH ET AL.: Investigating Road Safety Issues and Deaf People in the United Kingdom: An Empirical Study and Recommendations for Good Practice. In: Journal of Prevention & Intervention in the Community, 2010
- [2] HEINRICH NIEMANN: Klassifikation von Mustern. Springer, 2003
- [3] FRANCESCO BERITELLI, ET AL.: An automatic emergency signal recognition system for the hearing impaired. In: Digital Signal Processing Workshop, 12th - Signal Processing Education Workshop, 4th-edition, 2006
- [4] RUDOLF KRUSE, ET AL. : Computational Intelligence - Eine methodische Einfuehrung in Kuenstliche Neuronale Netze, Evolutionaere Algorithmen, Fuzzy-Systeme und Bayes-Netze. 2012
- [5] GOOGLE: Profiling with Traceview and dmtracedump. <http://developer.android.com/tools/debugging/debugging-tracing.html> Abgerufen: Januar 2013

Improving Ontology Construction Using Crowdsourcing and Machine Learning

Lukas Georgieff

Technische Universität Darmstadt
lukas.georgieff@hotmail.com

Kind of work: research seminar paper
Supervisor: Chris Biemann

Abstract: In the last years, the term ontology became more and more popular in computer science. Especially in the semantic web and language based applications, ontologies are used to improve the understanding of freely used language, so user queries and statements can be processed with higher accuracy. Each processed document, therefore, must provide ontology information that must be carefully created by hand for every single domain. In the past, several approaches were introduced to generate ontologies automatically from unstructured text. Hearst [He92] introduced the automatic acquisition of hyponyms by using so-called Hearst Patterns. The so extracted hyponyms and hypernyms can be used for creating ontological hierarchies. This paper focusses on improving the accuracy of the Hearst Patterns by using crowdsourcing platforms and machine learning (ML).

1 Introduction

With the publication of the Semantic Web idea, the concept of ontologies became very popular in computer science. Especially, applications focused on language processing often use the ontology concept to structure language and knowledge. Thus, data requires to include ontology information to be processed. Each such application, therefore, must define or make use of an existing ontology.

Generating an ontology means to reflect the domain knowledge into an abstract model including all concepts, a concept hierarchy (cf. [Ca99, CS05]), all concept features and relations between these concepts. For the most parts, ontologies are created manually, which is very time consuming. Thus, there is a strong interest in automatically creating ontologies with minimal error rates.

This paper focusses on improving the so-called Hearst Pattern accuracy in three steps: (1) Extract patterns from unstructured text. (2) The extracted information is monitored by crowd workers. (3) The results of step 2 are processed by a ML tool to recognize bad matches and improve the Hearst Pattern accuracy.

2 Related Work

Hearst [He92] showed how to extract hyponym-hypernym information from unstructured text using specific lexical patterns. In contrast of interpreting the entire text with all its detailed information, she discovered a few simple patterns which express the *is-a/type-of* relation. The problem of identifying these patterns lies in their requirements being highly accurate but still generic.

In contrast to the linguistic basics of this work, Chamberlain et al. [CKP09] demonstrate the strength of collaborative work. Instead of using a few expensive experts to process a task, it is proposed to use collaborative systems with a high amount of participants for this task.

Continuing the idea of Hearst Patterns and collaborative work, the next step is to apply these patterns on a crowdsourcing platform with the benefit of reaching a large amount of people while spending relatively little time creating the tasks for the crowd workers.

3 Crowdsourcing for Hearst Patterns

As previously described, improving Hearst Pattern accuracy is realized as a three-step procedure. Initially, unstructured text must be processed to extract hyponyms-hypernym matches. Afterwards, the extracted data is monitored by crowd workers. Finally, the data is fed to a machine learning tool.

We use the Leipzig Corpora Collection (LCC)¹ [Bi07] which is made of newspaper texts and randomly collected web sources. The first processing step is to extract all sentences per Hearst Pattern with its hypo- and hypernyms. Therefore, a java application extracts all six patterns defined by Hearst, e.g. NP_0 such as $\{NP_1, NP_2 \dots, (and | or)\} NP_N$ which for example matches the sentence “Programming languages such as F# and Scala use functional programming style.” with the hypernym: *Programming languages* and the hyponyms: *F#* and *Scala*.

The task for the crowdsourcing platform Crowdfunder² presents a sentence and a Hearst Pattern match. Finally, an assertion about the correctness of the match is presented. Therefore, the user may select the answer *True*, *False*, *Don't know* or *Bad extraction* can be selected. The value *True* is to be selected when the assertion is true, i.e. the Hearst Pattern match is correct, otherwise *False* must be selected. If a crowd worker is not able to decide on the assertion, *Don't know* allows to ignore his answer. The value *Bad extraction* covers extraction errors made by the application generating the test data.

Before judgments of a crowd worker are added to the final result each worker must pass several items for which the correct answer is known already. Since most answers are *True* it is important to select many *False* items to catch malicious workers.

¹ <http://corpora.informatik.uni-leipzig.de/>

² <http://crowdfunder.com/>

4 Learning from the Crowdsourced Data

The ML tool WEKA³, a data mining tool written in Java, is used for creating a Support Vector Machine (SVM) model. In the final SVM model, four features are used. In the first scenario, all part of speech (PoS) tags in a predefined word window before the hyponym part of the matched pattern are used as features. In the second scenario, the actual words are used as features. A third scenario contains a mixture of both as independent features, i.e. words and PoS tags are separate strings. The fourth scenario uses a list of PoS tagged words as features. The window width (WW) is changed from 1 to 10, i.e. a window of 2 uses the last two words/PoS tags in front of the matched hyponyms as separated features. After several runs, the optimal WW is identified as 4.

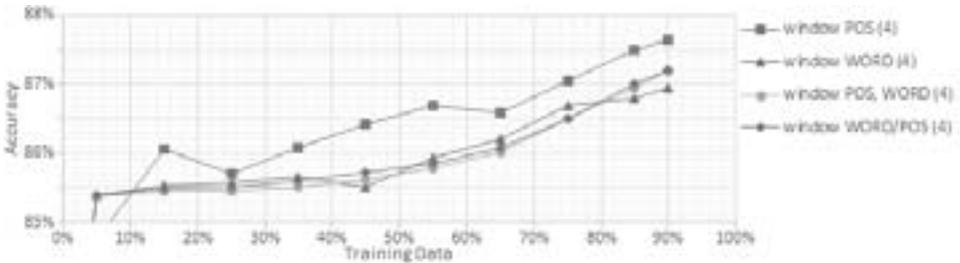


Fig.1: Learning curve, using 1% - 90% of 4500 items for training, rest for testing

Fig.1 shows the accuracy of the ML algorithm with a WW of 4 using the features described before. The accuracy tends to increase with a growing amount of data for model creation. In general, the accuracy starts with a high value, since the fraction of *True* datapoints is about 82% in the training data, so the SVM-model can reach a high accuracy by always choosing the most frequent class (i.e. *True*). Increasing the overall accuracy should be achieved by collecting even more data, especially more *False* pattern matches. The curves demonstrate that especially PoS tags, but also words and several combinations of words and PoS tags highlight the extraction quality of Hearst Patterns.

5 Conclusions

This paper presents a method for improving the accuracy of ontology creation using crowdsourcing and ML. Instead of processing the training data for the ML algorithms by experts, crowd workers yield similar quality but are more efficient considering time and costs. Compared to the original Hearst Patterns, the introduced concept improves the accuracy of building terminological ontologies from unstructured text by taking the PoS context into account. It might not cover all perspectives of creating ontological relations, nevertheless, this approach is an improvement to costs, efficiency and accuracy for creating ontological hierarchies and opens doors for further development.

³ <http://weka.wikispaces.com/>

References

- [Bi12] Biemann, C.: Creating a system for lexical substitutions from scratch using crowdsourcing, *Language Resources and Evaluation*, Springer, 2012
- [Bi07] Biemann, C., Heyer, G., Quasthoff, U., Richter, M.: The Leipzig Corpora Collection - Monolingual corpora of standard size, *Proceedings of Corpus Linguistics 2007*, Birmingham, UK, 2007
- [Ca99] Caraballo, S.: Automatic construction of a hypernym-labeled noun hierarchy from text, *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pp. 120–126, 1999
- [CS05] Cimiano, P., Staab S.: Learning concept hierarchies from text with a guided agglomerative clustering algorithm, *Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods (OntoML 05)*, Bonn, Germany, 2005
- [CKP09] Chamberlain, J., Kruschwitz U., Poesio, M.: Constructing An Anaphorically Annotated Corpus With Non-Experts: Assessing The Quality Of Collaborative Annotations, *Proceedings of the 2009 Workshop on The People’s Web Meets NLP: Collaboratively Constructed Semantic Resources (People’s Web)*, 2009
- [He92] Hearst, M.: Automatic Acquisition of Hyponyms from Large Text Corpora, *Proceedings of the Fourteenth International Conference on Computational Linguistics*, Nantes, France, 1992
- [Mi90] Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to WordNet: An on-line lexical database, *International Journal of Lexicography*, 3(4):235-244, 1990

Implementation of a Transformation System for Knowledge Bases Consisting of Relational, Probabilistic Conditionals

Marcus Marto
marcus.marto@gmail.com

Lehrgebiet Wissenbasierte Systeme, FernUniversität Hagen

Type of Work: Masterthesis, Advisor: Prof. Dr. C. Beierle

1 Introduction

First-order probabilistic conditional logic (FO-PCL) [Fis10, ch. 6-7] is a first-order extension to (propositional) probabilistic conditional logic (PCL) [RKI96] and combines first-order logic with probability theory to model uncertain knowledge. An FO-PCL knowledge base consists of a list of conditionals, and a conditional $\langle (\phi | \psi) [\xi], C \rangle$ consists of a premise ϕ , a conclusion ψ , a probability value ξ and a constraint formula C . Premise and conclusion are first-order formulas over a function-free, many-sorted signature Σ . The constraint formula consists of equations between two variables and a variables and a constant, it is used to restrict the set of ground instances that can be created from a conditional. The semantics of FO-PCL are defined via the ground instances of the conditionals in a knowledge base. Example 1.1 shows how uncertain knowledge can be expressed in FO-PCL.

Example 1.1 (Misanthrope):

The misanthrope example (adapted from Example 6.5.3 in [Fis10]) models the friendship relations for a group of people. Usually, if a person V likes a person U , then U also likes V . However, 'a' is a misanthrope and the chances that 'a' likes someone is considerably low. A possible choice of conditionals is:

$$M_1 : \langle (\text{likes}(U, V) | \text{likes}(V, U)) [0.9], U \neq V \rangle$$

$$M_2 : \langle (\text{likes}(a, V)) [0.05], V \neq a \rangle$$

A model of an FO-PCL knowledge base \mathcal{R} is a joint probability function and the method of maximum entropy is used to select the unique and most unbiased model, provided the knowledge base is consistent. The calculation of this probability distribution leads to an optimization problem with one free parameter for each *ground instance* of every conditional in the knowledge base, which is computational impracticable for larger knowledge bases. However, if the knowledge base is *parametrically uniform*, i.e. all ground instances

of a conditional share the same parameter value, the number of optimization parameters depends on the number of conditionals $|\mathcal{R}|$ only.

In [KB12] the syntactic criterion of *intra-rule* and *inter-rule interaction* is introduced together with a set of transformation rules \mathcal{PU} which allow to transform any consistent knowledge base into a semantically equivalent and parametrically uniform knowledge base with the same maximum entropy model. Interactions identify constellations that prevent a knowledge base from being parametrically uniform and can occur within a single conditional (intra) or between two conditionals (inter). The recognition of an interaction yields a pair (V, X) of a variable V and a variable or constant X . Each (sub-)type of an interaction has a specific transformation rule that is instantiated with a conditional R and the pair (V, X) . The application of a rule replaces the conditional R with two new conditionals $\nu(R')$ and $\nu(R'')$, thereby removing the interaction. $R' = \sigma(R)$ is the result of a variable substitution $\sigma = \{V/X\}$ applied to R , and $R'' = \bar{\sigma}(R)$ is the result of adding the constraint $V \neq X$ to the constraint formula of R . The operator ν transforms a conditional in constraint normal form, a normal form required for the recognition of interactions. The exhaustive application of \mathcal{PU} removes all interactions and always terminates. While \mathcal{PU} is correct and complete, it is not confluent as different orders of rule application can create different results.

The misanthrope example contains two interactions which can be solved by replacing M_1 (the interaction yields the pair (U, a)) with the conditionals

$$M_{1.1} : \langle \langle \text{likes}(a, V) \mid \text{likes}(V, a) \rangle [0.9], V \neq a \rangle$$

$$M_{1.2} : \langle \langle \text{likes}(U, V) \mid \text{likes}(V, U) \rangle [0.9], U \neq V, U \neq a \rangle$$

and $M_{1.2}$ (the interaction yields the pair (V, a)) with the conditionals

$$M_{1.2.1} : \langle \langle \text{likes}(U, a) \mid \text{likes}(a, U) \rangle [0.9], U \neq a \rangle$$

$$M_{1.2.2} : \langle \langle \text{likes}(U, V) \mid \text{likes}(V, U) \rangle [0.9], U \neq V, U \neq a, V \neq a \rangle.$$

In this work, a transformation system for \mathcal{PU} has been designed and implemented for an integration in KReator [FT12], which is an integrated development environment for relational probabilistic logic. Algorithms for the transformation in constraint-normal form, the recognition of interactions and the application of transformation rules have been developed and implemented.

2 Implementation

The basic architecture of the implemented transformation system is illustrated in Figure 1. The input knowledge base is parsed into an abstract syntax tree (AST) from which an object structure is created [Par07]. Thereby it is ensured that the types of variables and constants in a conditional are conform with the signature of the knowledge base (type consistency) and that the variables of the constraint formula are used in either the premise or the conclusion of the conditional. The key operations of the transformation phase are the

transformation of conditionals in constraint normal form, the recognition of interactions and the application of the transformation rules. These operation operate directly on the object structure. After the transformation phase, the output knowledge base is created from the resulting object structure.

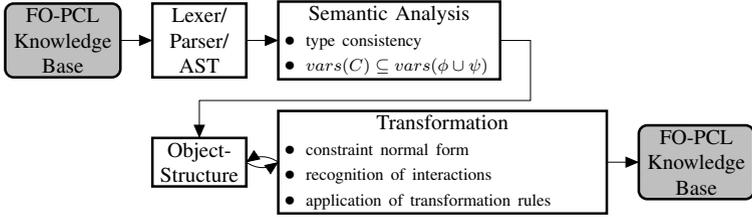


Figure 1: Conceptual pipeline of the transformation system.

2.1 Transformation Modes

The transformation systems offers three different transformation modes to transform a knowledge base: *Automatic*, *DFS* and *Interactive*. The *Automatic* mode applies the transformation rules in a fixed order and creates one resulting knowledge base. The *DFS* transformation mode creates all results that are obtainable by applying every possible order of rule application. For this purpose, a depth-first search is used. A node in the search tree is a knowledge base and has child nodes for every rewrite rule obtained from \mathcal{PU} . The child nodes are created by applying the corresponding rewrite rule to their parent node. A leave in the tree contains no interactions, i.e. it is parametrically uniform, and is written to file. The search stops when all nodes have been visited, i.e. all possible results have been created. A core feature of the transformation system is the *Interactive* mode, which allows to control, monitor and trace single steps of a transformation process via a graphical user interface. All transformation rules that are applicable are listed and can be applied separately.

2.2 User Interface

A transformation process can be started with KReator by executing a KReator-Script. All transformation parameters (e.g. transformation mode) can be set either by using a graphical user interface or within the script itself. The transformation systems supports batch processing for multiple knowledge base files.

A screenshot of the *Interactive* transformation mode is shown in Figure 2. The right table displays the conditionals of the knowledge base to be transformed and is updated accordingly after a transformation rule has been applied. All interactions that have been recognized are listed in a short form notation in the left table. An interaction can be selected to apply the corresponding transformation rule or to view more detailed information in the bottom pane about the reasons for the interaction.

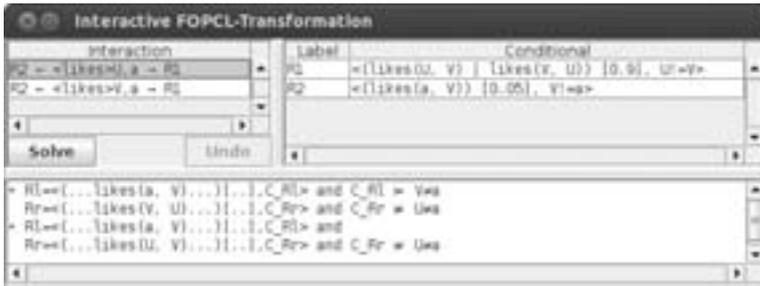


Figure 2: Interactive FO-PCL transformation mode.

3 Conclusion and Further Work

The implemented transformation system, which is based on the transformation rules \mathcal{PU} , allows to transform any consistent knowledge base to a semantically equivalent and parametrically uniform knowledge base. The implementation was successfully tested with various examples discussed in [KB12, Krä11]. An open question that was not addressed in this work is whether \mathcal{PU} can be complemented with additional transformation rules aiming to create a confluent set of transformation rules. A possible strategy to investigate this further could be to analyse results obtained by the *DFS* transformation mode, which creates all results that can be obtained by different orders of rule application. Our current work includes the transformation of larger knowledge bases to obtain more empirical data about the number of interactions that exist during a transformation process.

References

- [Fis10] J. Fisseler. *Learning and Modeling with Probabilistic Conditional Logic*, volume 328 of *Dissertations in Artificial Intelligence*. IOS Press, Amsterdam, 2010.
- [FT12] Marc Finthammer and Matthias Thimm. An integrated development environment for probabilistic relational reasoning. *Logic Journal of the IGPL*, 20(5):831–871, 2012.
- [KB12] A. Krämer and C. Beierle. On Lifted Inference for a Relational Probabilistic Conditional Logic with Maximum Entropy Semantics. In T. Lukasiewicz and A. Sali, editors, *Foundations of Information and Knowledge Systems (FoIKS 2012)*, volume 7153 of *LNCS*, pages 224–243. Springer, 2012.
- [Krä11] Annika Krämer. *Transformation rules for lifted inference in relational probabilistic logic knowledge bases*. B.Sc. Thesis. Dept. of Computer Science, FernUniversität in Hagen, Germany, 2011.
- [Par07] Terence Parr. *The Definitive ANTLR Reference: Building Domain-Specific Languages*. Pragmatic Bookshelf, 2007.
- [RKI96] Wilhelm Rödder and Gabriele Kern-Isberner. Representation and Extraction of Information by Probabilistic Logic. *Inf. Syst.*, 21(8):637–652, 1996.

Crowdsourcing slang identification and transcription in twitter language

Benjamin Milde

milde@stud.tu-darmstadt.de

Supervised by: Prof. Dr. Chris Biemann

Kind of work: Seminar research

Abstract: This paper describes a case study for finding and transcribing slang and abbreviated words commonly used in twitter language. Using unusual words not found in a reference corpus, we create a candidate list for possible words worth exploring. We design a Human Intelligence Task (HIT) for the popular crowdsourcing platform crowdflower.com and pay non-experts to categorize and transcribe our candidate words. With the collected and labeled data we use machine learning to build a detector for slang and abbreviated words. We also build a dictionary of these words, which can be useful for the normalization of twitter texts in other NLP-tasks.

1 Introduction

Twitter, the most famous microblogging platform, is an abundant text corpus, as millions of tweets - microposts on twitter - are written daily and are easily accessible. However the informal nature of most tweets poses new challenges for natural language processing (NLP) algorithms. Slang words, abbreviations and (intended) misspellings of words are common examples of the extremely informal nature of the short texts from tweets, as users frequently abbreviate their posts to fit within the specified 140 character limit [FT07].

In this work, we leverage the power of crowdsourcing to automate human annotation by paid non-experts, to generate a list of labeled out of vocabulary (OOV) words from a 180 million word twitter corpus. This can then be used to train machine learning algorithms to automate the task of categorizing and detecting slang words. We also collect transcriptions and corrections of misspellings for our words, to create a OOV-dictionary which can possibly be useful in twitter language normalization.

2 Related work

Finin et. al. [FMK⁺10] annotated named entities from a large twitter corpus using crowdsourcing and came to the conclusion that this is a reliable and cheap way to delegate the tedious task of hand-labeling. Han et.al. [HB11] analysed and categorized OOV-words

from twitter, but labeled a smaller portion of the data per hand (450 words). There are also some efforts to normalize twitter mircotexts. Xue et. al. [XYD11] proposed a multi-channel model to normalize the text which was inspired by source channel theory. The performance of their model was measured in respect to 800 hand normalized sentences. Sood et. al. used a corpus of comments from a social news site and crowd sourcing to improve automated profanity detection using machine learning techniques [SAC12]. To our knowledge, there are currently no previous works on building a specific dictionary of common misspellings, slang words and other common OOV-words from twitter in significant size using crowd-sourcing.

3 Candidates for HIT-evaluation

To obtain a list of candidate words for slang words, abbreviations and common misspellings, we first built a word frequency dictionaries for a large twitter corpus collected in 2012 with 25 million tweets. The tweets are preprocessed, some obvious misspellings like repeating characters are automatically resolved to dictionary words (e.g. loveeeeeeeee → love) before labeling them as OOV-words.

All twitter words not contained in a reference corpus that appear frequently are then labeled as candidate words. We use the simple English Wikipedia as reference corpus ¹. Internet and twitter slang and twitter specific vocabulary is more prevalent in higher frequent OOVs (e.g. tweeting, retweet, followback, lmao, omfg, ...), while everyday slang words appear less frequent (e.g. booze, quid, bitchy, ...).

4 HIT design and results

We use the Crowdfower ² platform for our HITs. We designed a task in which workers had to identify a matching category for a OOV-word and additionally transcribe the word for us. Each task had five words to categorize and transcribe. Context is given in form of four sentences from twitter where the word occurred and the word in question is marked with an asterisk.

We are mainly interested in common misspellings and alternate spellings (gonna → going to, yr → your, ineed → i need) and slang words (slackling, bloke, belieber, dmed ... ³) which can sometimes be related to twitter itself, like ‘dmed’ for ‘direct messaged’, or are also used in every day language, like the British slang term ‘bloke’ for ‘guy’.

Crowdfower uses gold question automatically to track the performance of workers and rate their answers with a trust score. We provided 10 answers to selected tasks as gold data. If a worker misses to many gold questions, all his judgements become ‘untrusted’

¹<http://simple.wikipedia.org/>

²<http://crowdflower.com>

³For the curious, ‘slackling’ is a term for ‘to not work for long periods of time’ but can also be related to the trend sport slacklining. ‘Belieber’ is a slang word for the fans of Justin Bieber.

and are not included in the final results. We uploaded the 5000 most frequent candidates in random order from our word list. We paid \$150 in total over the course of one week and this resulted in 1900 finished items. We observed that roughly 12% of the candidate words are categorized as slang. It is worth noting, that Han et.al. [HB11] reports a similar rate for slang terms in OOV-words in twitter. Abbreviation / Alternate spelling + Slang words together make up 70% of our candidate list, which are the two groups of words we are interested in to train a classifier.

5 Automated categorization

We built a second word list, starting with all words that are contained in our twitter corpus. We filtered out words that did not appear in a smaller corpus of Gutenberg texts ⁴ and dictionary words. We also filtered words that did not appear in a similar frequency span in the twitter corpus as the candidate slang words so far. In a first experiment we trained several machine learning (ML) algorithms using a two class problem: decide if a word is ‘normal’ or ‘slang + misspelled or abbreviated’. We use the words from the method above as normal words, and all words categorized from workers as slang, misspelled or abbreviated as second category. We also excluded candidate words in cases where less than 66% of workers agreed on the category.

As features for the classifier we use TF-IDF vectors of character n-grams directly from the words. Already with the limited data of single words, a significant learning score above baseline (78%) can be achieved. We can also build a second classifier which uses the context of the words, i.e. the tweets they appear in, with character n-grams as features where $1 \leq n \leq 5$. Figure 1 plots the number of feature vectors used for learning against achieved F1-score. We used 10-fold crossvalidation for all data points.

6 Conclusion

Crowdsourcing is a cheap alternative to hand-labeling NLP datasets. We showed that the task of identifying and transcribing misspellings and slang words from twitter can be successfully mapped to a crowdsourcing task, which can be done by non-experts without training. The big advantage over hand-labeling is that it can be scaled to significant data sizes that are otherwise usually not possible. This is important for machine learning algorithms those performance usually also scales with the amount of training data available. This is also the case with our data, when using ML-methods to automate some of the categorisation. The collected data might also be useful to measure the performance of an attempt to automate transcription of OOV-words. Generally, twitter language normalization might profit from a dictionary of common twitter OOV-words.

⁴www.gutenberg.org

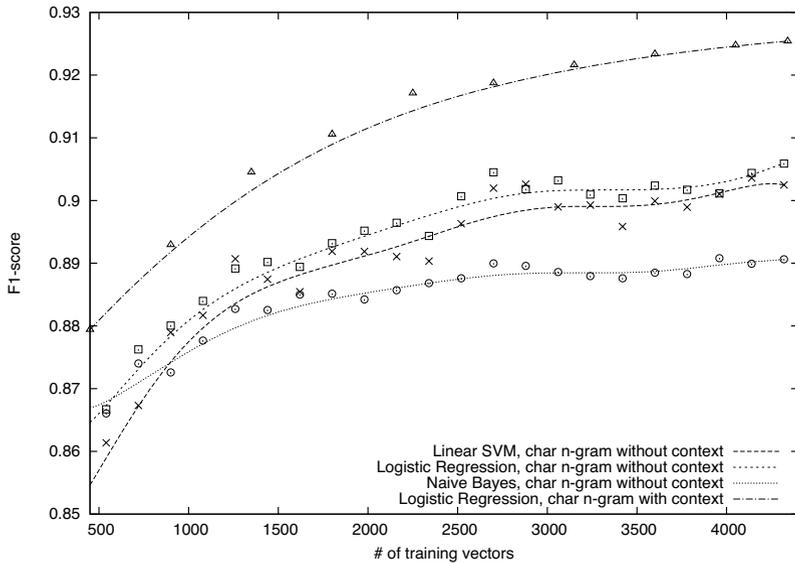


Figure 1: Number of feature vectors used for learning and achieved cross-validated F1-score for the slang word detection. All three tested learning algorithms perform similarly well. Naturally, providing context to the model in form of example tweets improves performance (Logistic regression, character n-gram with context). The baseline is roughly 78%.

References

- [FMK⁺10] Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. Annotating Named Entities in Twitter Data with Crowdsourcing. 2010(June):80–88, 2010.
- [FT07] Tim Finin and Belle Tseng. Why We Twitter : Understanding Microblogging. 2007.
- [HB11] Bo Han and Timothy Baldwin. Lexical Normalisation of Short Text Messages : Makn Sens a # twitter. pages 368–378, 2011.
- [SAC12] Sara Owsley Sood, Judd Antin, and Elizabeth F Churchill. Using Crowdsourcing to Improve Profanity Detection. Technical report, 2012.
- [XYD11] Zhenzhen Xue, Dawei Yin, and Brian D Davison. Normalizing Microtext. 2011.

Limits of *SRIOQ* in the Context of Reasoning Support for Multi-Agent Plan Modelling

Stephan Opfer

University of Kassel,
Distributed Systems Research Group,
Wilhelmshöher Allee 73, 34121 Kassel
opfer@vs.uni-kassel.de

Type of Work: Master's Thesis

Supervisors: Prof. Dr. Kurt Geihs, Dr. Hendrik Skubch

Abstract: Providing reasoning support during the modelling of Multi-Agent plans improves the quality of the results and the efficiency of the modelling process. Therefore, we investigated the applicability of a description logic reasoning support for the Multi-Agent modelling language ALICA. We choose *SRIOQ* as one of the most expressive description logics. For a proper judgement of its applicability, several reasoning tasks of different complexity are implemented. The encountered problems are highlighted and discussed. Finally it is concluded that only improved or alternative reasoning techniques are suitable for the complexity of Multi-Agent plans.

1 Introduction

In order to describe the behaviour and coordination of Multi-Agent Systems an appropriate modelling formalism is necessary to handle the related complexity. One of the most current formalisms and the context of our investigation is ALICA –A Language for Interactive Cooperative Agents [Sku12]. The plan structure of ALICA programs involves common concepts like finite state machines, tasks, plan hierarchies, sets of alternative plans, cardinalities, pre-, runtime-, and postconditions. Therefore, our results are also relevant for other formalisms, utilising description logics. The objective of providing a description logic reasoning support during the modelling process of Multi-Agent plans is to improve the plan quality and the efficiency of the modelling process. For example, the well-formedness of plans can be guaranteed by reasoning about the plan structure and concurrently considering the domain specific knowledge of annotated conditions. Furthermore, the description logic based approach facilitates exchanging or expanding of domain specific knowledge through OWL 2.0 technologies, which in turn improves the domain independence of the modelling formalism itself. Chapter 2 shortly introduces the general ALICA plan structure and highlights some problematic reasoning tasks. The details and possible solutions are given in Chapter 3 and 4, respectively.

2 Reasoning Support Tasks

In Figure 1 the abstract structure of an ALICA program is shown. The program includes four plans. One of them is the Root plan, which only includes the Default task, that is related to a finite state machine with only one state, namely Z_0 . If agents enter a state, they have to execute the included parts of the ALICA program. In case of Z_0 , they have to choose between three alternative plans, which are part of the Root Plantype. In general, plantypes create new levels in the plan tree hierarchy whereas behaviours, as in state Z_3 , are atomic elements. Although Plantype 4 is used in Alternative 2 and Alternative 3, the plan hierarchy is interpreted as tree at runtime.

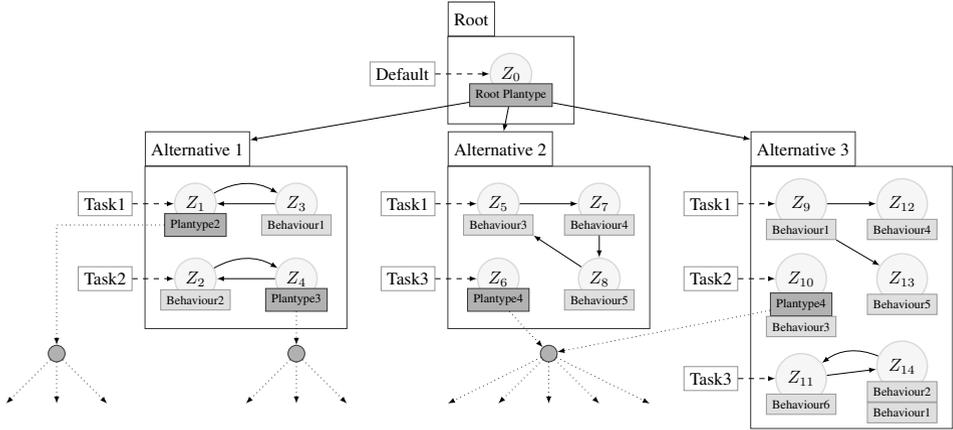


Figure 1: Abstract Part of an ALICA Program

The plan tree hierarchy has to be finite, which disallows cycles. If the plan modeller inserts the Alternative 2 in Plantype 4, Alternative 2 includes itself and the plan tree is infinite. Therefore, one of the implemented reasoning tasks is to check whether such cycles exist in the plan hierarchy. Other simple reasoning tasks ensure, that transitions connect states only within a plan and that the minimum and maximum cardinalities of a task are valid. A more complex reasoning task is to ensure the locality of conditions. The transitions between states, e.g., are guarded with conditions. These conditions are allowed to include domain specific parts and parts about the execution state of the ALICA program itself. Parts about the execution state are restricted to the scope of their corresponding plan. For example, it is not allowed to guard the transition between Z_5 and Z_7 with the condition that some agent is in state Z_9 , because Z_9 is not part of the plan Alternative 2.

In order to formulate the corresponding axioms for the given reasoning tasks, we choose the description logic *SR⁺OIQ* [HKS06], because it is one of the most expressive description logics and as it provides the semantic for the commonly known OWL 2.0, it has a great tool support. Despite of its expressiveness, the corresponding axioms are inexpressible within *SR⁺OIQ*. Only with the additional expressiveness of SWRL rules [HS04], it is possible to implement the aforementioned reasoning tasks. Furthermore, for some other reasoning tasks even the combination of *SR⁺OIQ* and SWRL rules is not expres-

sive enough. In order to capture the semantic of plantypes disjunctions about subconcept axioms would be necessary. Reconsidering Figure 1, all agents executing the Root plan have to execute the same alternative child plan. During the modelling process no decision is made and therefore it is necessary to axiomatise the fact that either the agents in Alternative 1, the agents in Alternative 2, or the agents in Alternative 3 are a subconcept of the agents in the Root plan. Axiom 1 captures this semantic, but it is not expressible with the combination of *SROIQ* and SWRL rules.

$$\begin{aligned}
 & (\text{Agent} \sqcap (\text{inPlan value Root}) \sqsubseteq \text{Agent} \sqcap (\text{inPlan value Alternative 1})) \\
 & \sqcup (\text{Agent} \sqcap (\text{inPlan value Root}) \sqsubseteq \text{Agent} \sqcap (\text{inPlan value Alternative 2})) \\
 & \sqcup (\text{Agent} \sqcap (\text{inPlan value Root}) \sqsubseteq \text{Agent} \sqcap (\text{inPlan value Alternative 3})) \quad (1)
 \end{aligned}$$

3 Discussion

Considering the stated reasoning tasks, the combination of *SROIQ* and SWRL rules seems to be acceptable, because most of the necessary axioms are expressible and the inexpressible parts can be tackled by complementing algorithms. The semantic of plantypes can be handled by adding one subconcept relation at a time and check whether this subconcept relation retains the knowledge base's consistency.

Two more general problems contradict the utilisation of *SROIQ*. The open world assumption, made by *SROIQ*, is inappropriate in the context of completely known plan structures and increase the risk of ontological overcommitment. A plan without states, e.g., would not create an inconsistent knowledge base, because the reasoner has no evidence to belief that the plan is different from other plans, which have states. In fact, the reasoner assumes to know, that no plan without states exists and therefore deduces the actual inconsistent plan to have at least one unknown state, which makes it consistent. In order to forbid the reasoner to make such open world based conclusions, a lot of additional axioms, which close the open world, are necessary. The list includes covering axioms, closure axioms, domain and target restrictions for roles, number restrictions, negative role assertions for individuals, and the definition of nominals. Especially the last two kinds of restrictions are coupled with a great deal of effort, since the assertions have to be stated for a lot of plan elements and cannot be part of the static background knowledge about ALICA programs. Furthermore, the risk of ontological overcommitment is increased due to the amount of additional axioms, which are necessary to close the open world. A knowledge representation is ontological overcommitted, if the point of view defined by the given axioms is more restrictive than necessary [Sch08]. If an ontology about people, e.g., requires that everyone has exactly one first name, middle name, and surname, it is probably overcommitted, because a lot of people have several additional middle names and some people do not have a middle name. As a consequence, it would be more appropriate to drop any number restrictions on the middle names. Nevertheless, it would be impossible to deduce, that two different people have the same name, because the reasoner has no evidence to belief that, given two people, all middle names are known. The needed evidence could be provided by introducing a nominal concept for middle names and negating

every connection from the two people to middle names, which does not belong to their names. In order to retain the necessary evidence, the middle name concept and the list of negated connections have to be updated every time a new middle name is introduced to the ontology. Similar effort is necessary to make the conclusion possible, that two states have the same number of incoming transitions. Therefore, it is very seductive to accept the overcommitment of the ontology and require number restrictions for middle names. As a consequence, the reasoner has evidence to deduce, that two people have the same name, because the maximum number of middle names for both people is reached and therefore all middle names are known. As described before, a lot of restrictions have to be made in order to close the open world in the context of completely known plan structures. The abundance of restrictions increases the risk of ontological overcommitment, because it is seductive to save labour by using more general and restrictive axioms. Furthermore, the large number of restrictions increases the ontologies complexity, which again increases the number of modelling failure like unintended overcommitment.

4 Conclusion

The expressiveness of *SRIQ* and SWRL rules is insufficient in the context of state of the art plan structures. Although the lack of expressiveness can be handled by some complementing algorithms, the effort to close the open world makes a description logic based reasoning support for the modelling process of Multi-Agent plans even more inappropriate and increases the risk of ontological overcommitment. Research tackling this problem is done by Motik et al. [MH09] and Tao et al. [TSBM10]. Alternatively, autoepistemic logics provide a good compromise between the open and closed world assumption [BL04].

References

- [BL04] R. Brachman and H. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [HKS06] I. Horrocks, O. Kutz, and U. Sattler. The Even More Irresistible *SRIQ*. In P. Doherty, J. Mylopoulos, and C. A. Welty, editors, *KR*, pages 57–67. AAAI Press, 2006.
- [HS04] I. Horrocks and U. Sattler. Decidability of *SRIQ* with complex role inclusion axioms. *Artif. Intell.*, 160(1&2):79 – 104, 2004.
- [MH09] B. Motik and U. Horrocks, I. and Sattler. Bridging the gap between OWL and relational databases. *J. Web Sem.*, 7(2):74 – 89, 2009.
- [Sch08] G. Schreiber. Knowledge Engineering. In F. v. Harmelen, V. Lifschitz, and B. Porter, editors, *Foundations of Artificial Intelligence*, chapter 25, pages 929–946. Elsevier, 2008.
- [Sku12] H. Skubch. *Modelling and Controlling Behaviour of Cooperative Autonomous Mobile Robots*. PhD thesis, University of Kassel, 05 2012.
- [TSBM10] J. Tao, E. Sirin, J. Bao, and D. L. McGuinness. Integrity Constraints in OWL. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. AAAI Press, 2010.

MVC-Pattern in einem verteilten System

Software Architektur Patterns

Hochschule Reutlingen
Alteburgstraße 150
72762 Reutlingen

Franziska Anders (Franziska.Anders@Student.Reutlingen-University.de),
Michael Däubler (Michael.Daeubler@Student.Reutlingen-University.de),
Kolja Hick (Andreas_Kolja.Hick@Student.Reutlingen-University.de),
Denis Krajcevic (Denis.Krajcevic@Student.Reutlingen-University.de)

Abstract: Dieser Aufsatz befasst sich mit dem Einsatz des Model-View-Controller-Patterns in einem verteilten System. Dabei wird der klassische MVC-Ansatz erläutert, Anforderungen und Probleme in verteilten Systemen aufgezeigt und ein Lösungskonzept aufgestellt.

1 Problemstellung

Wie kann eine interaktive Anwendung in einem verteilten System flexibel gestaltet werden, sodass unterschiedliche Views, Anpassungen an Benutzeroberflächen oder unterschiedliche Benutzertypen keine Auswirkungen auf den funktionalen Kern eines Systems haben? Aufgrund der unterschiedlichen Endgeräte und Interaktionsmöglichkeiten, die heutzutage bestehen, sind die oben genannten Anforderungen unvermeidbar. Daher bedarf es einer Softwarearchitektur, welche einen geeigneten Lösungsansatz liefert.

2 Lösungskonzept mit dem Model-View-Controller-Pattern

2.1 Der klassische MVC-Ansatz

Eine Entkopplung der Kernfunktionalität von den Benutzeroberflächen und der Interaktionskomponenten steigert die Flexibilität eines Systems. Das Model-View-Controller-Pattern verfolgt genau diesen Ansatz und unterteilt eine interaktive Anwendung in drei Hauptkomponenten.

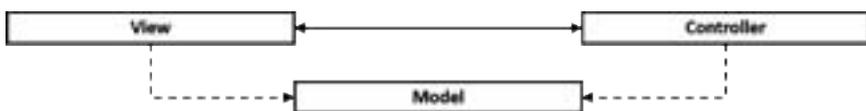


Abbildung 1: Model-View-Controller

Das Model stellt dabei die Kernfunktionalität dar und beinhaltet grundsätzliche Daten. Es ist vollkommen unabhängig von Benutzeroberflächen, Interaktionsmöglichkeiten und Inputdaten. Die einzige Verbindung zwischen dem Model und den Views und Controllern besteht darin, dass sich diese bei dem Model registrieren können. Bei Änderungen an den Daten des Models werden alle registrierten Views und Controller benachrichtigt. Daraufhin rufen die Views und Controller die aktuellen Daten des Models ab.

Eine View präsentiert dem Anwender Informationen und stellt damit die Benutzeroberfläche dar. Die anzuzeigenden Daten stellt das Model bereit. Um bei Änderungen des Models, die aktuellen Daten zu beziehen, beinhaltet jede View eine Update-Prozedur, welche nach einer Benachrichtigung vom Model automatisch initiiert wird. Zu jeder View gibt es einen assoziierten Controller.

Der Controller erhält vom User über die View Inputdaten, die normalerweise als Events realisiert werden und leitet diese als Service-Request an das Model oder die View weiter. Der Controller ist somit für die Benutzereingaben verantwortlich und definiert alle Ereignisbehandlungen, die eine View anbietet. Das Verhalten eines Controllers kann von einem Model abhängen. Ist dies der Fall, muss auch der Controller eine Update-Prozedur beinhalten, um, nach einer Benachrichtigung des Models, die aktuellen Daten beziehen zu können. Die View und der Controller bilden zusammen die Benutzerschnittstelle für den Anwender.

Bei einer objektorientierten Implementierung des MVC-Patterns würde für alle Komponenten mindestens eine Klasse erstellt werden, welche die jeweiligen Funktionalitäten realisiert. Dabei erben die View und der dazugehörige Controller von einer Klasse, die die Update-Funktionalität beinhaltet. Um die Benachrichtigungs-Prozedur und die Update-Prozedur zu realisieren, können unterschiedliche Design-Patterns eingesetzt werden. Ein Beispiel dafür ist das Observer-Pattern, auf welches in dieser Arbeit nicht weiter eingegangen wird.

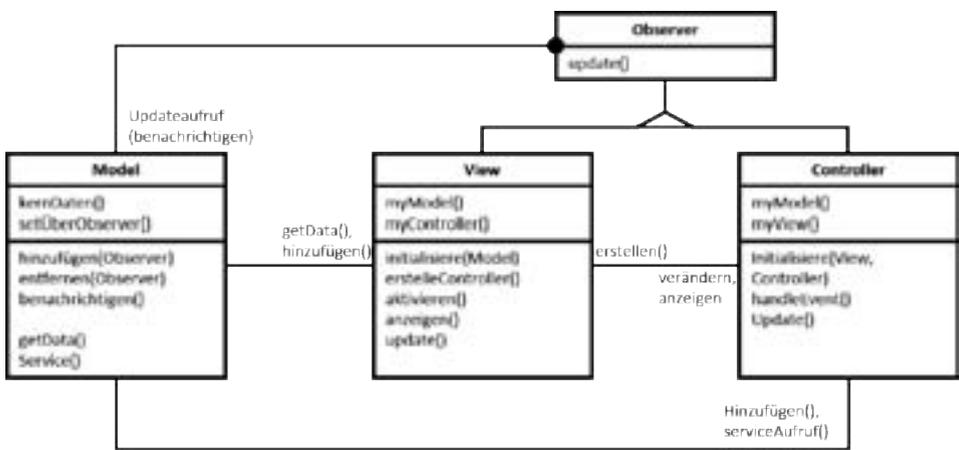


Abbildung 2: MVC Implementierung

Das MVC-Pattern verfolgt eine strenge Entkopplung des Modells von den Benutzerschnittstellen-Komponenten. Dadurch können mehrere Views implementiert und mit einem einzigen Modell verwendet werden. Eine weitere Konsequenz ist die dynamische Verwendung der Views. Während der Laufzeit eines Systems, können mehrere Views verwendet, geöffnet oder geschlossen werden. Es besteht kein Einfluss auf das Modell. Die Benachrichtigungs-Prozedur des MVC-Patterns, ermöglicht synchronisierte Benutzeroberflächen, da die Benachrichtigung ausgehend vom Modell an alle registrierten Views und Controller geschickt wird. Mit dieser Benachrichtigung ruft jede registrierte Komponente die aktuellen Daten ab. Somit sind alle Views synchronisiert. Das Entkopplungskonzept des MVC-Patterns ermöglicht den Austausch der Views und Controller eines Modells. Eine weitere positive Eigenschaft ist die Austauschbarkeit vom „Look-And-Feel“ einer Anwendung. Mit „Look-And-Feel“ ist die Interaktion zwischen dem User und der Anwendung gemeint. Dabei wird beispielsweise unterschieden, ob sich die Anwendung per Mausklick steuern lässt oder per Touchscreen mit Wischbewegungen. Unterschiedliche „Look-And-Feel“-Standards haben keine Auswirkung auf die Kernfunktionalität. Hierfür werden lediglich passende View- und Controller-Implementierungen für unterschiedliche Plattformen benötigt.

Bei einem falschen Einsatz des MVC-Patterns, können negative Eigenschaften auftreten, welche beispielsweise die Komplexität der Anwendung steigern und die eigentlichen Vorteile in den Hintergrund stellen. Zum Beispiel sollten simple Texte oder Menükomponenten nicht mit eigenen Views und Controllern realisiert werden. Dadurch würde die Komplexität enorm steigen und ein Overhead an Benachrichtigungen entstehen. Die Verwendung des MVC-Patterns kann eine unnötige Anzahl an Updates verursachen. Einzelne und simple Useraktionen sollten keine hohe Menge an Update-Prozeduren verursachen, außer diese sind nicht zu vermeiden. Als Beispiel kann hier die Veränderung einer Fenstergröße genannt werden. Diese sollte keine Update-Prozedur als Folge haben. Des Weiteren führt die enge Kopplung zwischen der View und dem dazugehörigen Controller zu einer schlechten Wiederverwendbarkeit der einzelnen Komponenten. Sie sind zwar separate Komponenten, jedoch besteht eine enge Korrelation und sie können dadurch nicht getrennt voneinander verwendet werden. Auch die starke Abhängigkeit von View und Controller zum Modell, macht diese nur schwer wiederverwendbar.

2.2 MVC in einem verteilten System

Der klassische MVC-Ansatz kann in verteilten Systemen nicht eins zu eins übernommen werden. Nehmen wir an, wir haben ein simples verteiltes System, das mit dem klassischen MVC-Pattern umgesetzt wurde und eine Benutzerschnittstelle auf einem Smartphone anbietet. View und Controller befinden sich somit auf dem Smartphone und das dazugehörige Modell befindet sich auf einem Server. Über das Smartphone lassen sich die Daten, die auf dem Server liegen, verändern. Der User startet eine Transaktion und ruft die Services des Modells auf dem Server auf. Da sich das Modell auf dem Server befindet, sind keinerlei Daten auf dem Smartphone gespeichert. Wenn die Transaktion aus irgendeinem Grund abbricht, hat das Smartphone keine Informationen über die aktuellen Daten oder den Status der Transaktion. Die Daten gehen leicht verloren und

können nicht rekonstruiert werden. Aus diesem Grund ist das MVC-Pattern minimal zu modifizieren. In einem verteilten System benötigt jede Benutzerschnittstelle ein eigenes Model, in dem der aktuelle Status und die aktuellen Daten gespeichert sind. Dieses (lokale) Model greift auf das (entfernte) Server-Model zu.

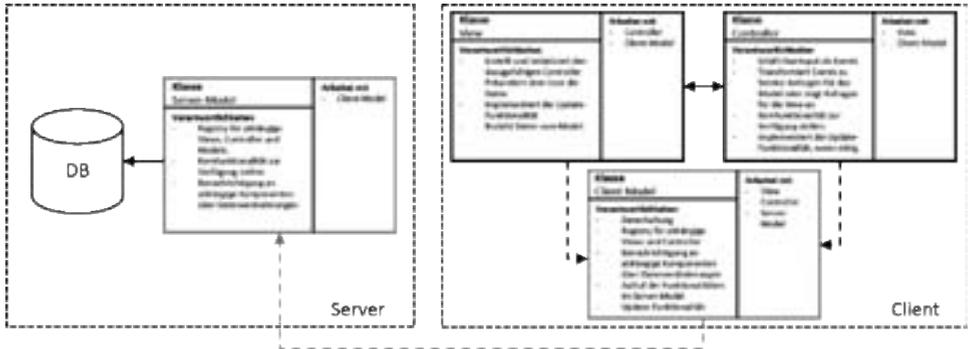


Abbildung 3: MVC in einem verteilten System

Das zusätzliche Client-Model hat die Aufgabe, die Daten zu halten und die eigentlichen Services des Modells auf dem Server aufzurufen. Um über Datenveränderungen des Server-Modells informiert zu werden, registriert es sich bei diesem. Das Server-Model beinhaltet, genau wie im klassischen Ansatz, eine Registry, in der alle am Model interessierten Komponenten registriert sind. Über die Registry benachrichtigt das Server-Model alle registrierten Komponenten über die eigene Veränderung. Daraufhin führt das Client-Model die Update-Prozedur aus. Nach dem erfolgreichen Update meldet auch das Client-Model, dass es sich verändert hat. Jeder Interessent vom Client-Model führt die Update-Prozedur aus. Durch die Erweiterung um ein Client-Model können nun die Daten auf Clientseite gespeichert werden. Sollte eine Transaktion abbrechen, besteht die Möglichkeit, die Transaktion ohne Datenverlust auf Clientseite zu wiederholen.

3 Einordnung und Abgrenzung ähnlicher Ansätze

Das MVC-Pattern lässt sich bei interaktiven Applikationen mit flexiblen Benutzeroberflächen einsetzen. Es gibt noch weitere Patterns, die in diesem Kontext verwendet werden. Das Presentation-Abstraction-Control-Pattern ist ein Beispiel dafür. Dieses Pattern teilt ein System in eine Baumhierarchie von PAC-Agents auf. Jeder dieser PAC-Agent beinhaltet drei Komponenten. Die Abstraction-Komponente ähnelt dem Model im MVC-Pattern. Die View und der Controller im MVC werden durch die Presentation-Komponente realisiert. Die Kommunikation und Steuerung zwischen diesen PAC-Agents übernimmt die Control-Komponente. Das PAC-Pattern organisiert die einzelnen Agents aufgrund ihrer Hierarchie optimal, sodass die gesamte Funktionalität des Systems gewährleistet wird.

Integration modellbasierten Testens in die Entwicklungsprozesse eines mittelständischen Unternehmens der Automatisierungsbranche

Kai Beckmann

Hochschule RheinMain, Labor für Verteilte Systeme,
Unter den Eichen 5, 65195 Wiesbaden
kai.beckmann@hs-rm.de

Art der Arbeit: Promotionsprojekt
Betreuer: Prof. Dr. Uwe Brinkschulte, Prof. Dr. Reinhold Kröger

1 Motivation

Software ist ein wesentlicher Faktor in der industriellen Wertschöpfung geworden. Gerade in Bereichen wie der Automatisierungsbranche können mangelnde Qualität oder Fehler in Produkten Schaden anrichten und unnötige Kosten verursachen. Das systematische Testen ist ein integraler Bestandteil des Software-Lebenszyklus und der Qualitätssicherung [Lig09]. Insbesondere das automatisierte Testen, beispielsweise für Regressionstests, kann zu hohen Effizienzgewinnen führen [RBGW10]. Die modellgetriebene Software-Entwicklung und das modellbasierte Testen versprechen, die wachsende Komplexität handhabbarer zu machen und die Effizienz zu erhöhen [SVEH07] [RBGW10]. Der Sicherheitsstandard IEC 61805-3 empfiehlt für kritische Anwendungen den Einsatz modellbasierter Techniken und Prozesse [IEC10]. Diesem Stand der Technik entsprechend gibt es eine Reihe akademischer Ansätze und realer am Markt eingeführter Produkte [MOSHL09] [Sch12].

Die Einführung dieser „modernen“ Technologien und Werkzeuge stellt aber in der Praxis gerade kleine und mittlere Unternehmen (KMU) vor große Herausforderungen. Dort wird häufig auf bestehenden Projekten aufgebaut und es existieren gewachsene Strukturen aus einem Gemenge an Modellen, Verfahren und Werkzeugen, die auch Eigenentwicklungen ohne strenge Orientierung an Standards sein können. Die Einführung vollständig neuer, dem Stand der Technik entsprechenden, Prozesse kann die Ressourcen eines KMUs übersteigen. Die Schulung der Entwickler und die Umstellung oder Portierung der existierenden Systeme und Projekte sind während des laufenden Tagesgeschäfts nahezu unmöglich.

Um eine praxistaugliche Einführung neuer Prozesse zu ermöglichen, ist es daher notwendig die existierenden Gegebenheiten zu berücksichtigen und zu integrieren. Häufig wird auch eine schrittweise Einführung gewünscht, die eine kontrollierte kontinuierliche Modernisierung der Software-Entwicklung ermöglicht.

2 Ansatz

Für die Systemmodellierung haben sich generell die „Unified Modeling Language“ (UML) der „Object Management Group“ (OMG) und entsprechende Erweiterungen über UML-Profile durchgesetzt [OMG11]. Für den Aspekt des modellbasierten Testes wurde bereits 2005 das „UML Testing Profile“ (UTP) veröffentlicht [OMG05]. Allerdings ist die Verwendung von UTP für die Testmodellierung und die Unterstützung von Testwerkzeugen bisher nicht sehr stark verbreitet [MOSHL09] [SL10]. Auf Grund der aktiven Weiterentwicklung durch die OMG soll UTP dennoch zusammen mit UML die Basis für den Lösungsansatz dieser Arbeit bilden.

Für den Ansatz dieser Arbeit sollen individuelle Domänenspezifische Sprachen („Domain-specific Language“, DSL) für die Modellierung von Systemen und Tests unterstützt, bzw. verwendet werden, um so die in den Unternehmen existierenden Modelle zu berücksichtigen. Über entsprechende Transformationen werden diese spezifischen Modelle in eine generische „Modellwelt“, basierend auf UML, überführt. Wie in Abbildung 1 dargestellt, soll die Modellierung vom System- und Testmodellen mit unternehmensspezifischen Werkzeugen in Form von DSLs geschehen können. Diese Werkzeuge können bereits existieren oder individuell für die Begebenheiten in einem Unternehmen entwickelt oder angepasst werden. Letzteres kann beispielsweise für die Einführung von modellbasierten Testen notwendig werden. Dafür müsste dann eine passende Test-DSL entworfen werden, die sich gut in die vorhandenen Strukturen einfügt und die notwendigen Testaspekte der Anwendungsdomäne berücksichtigt.

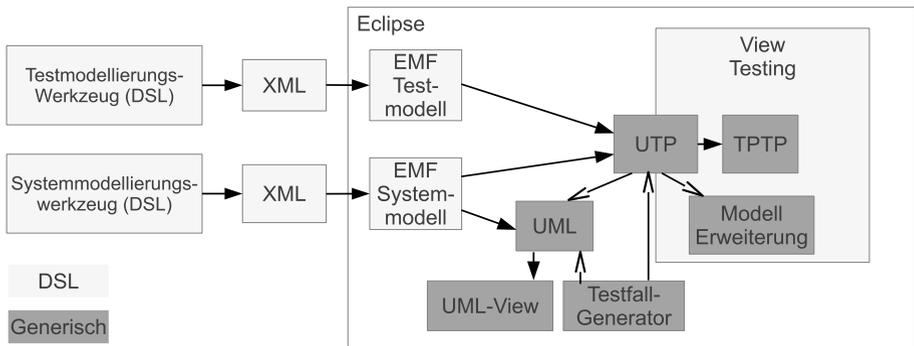


Abbildung 1: Einbindung existierender oder angepasster Modelle in eine Testumgebung.

Da die Modelle oder Speicherformate existierender Werkzeuge ggf. für die weitere Verarbeitung ungeeignet sind, wird eine Zwischentransformation in ein abstrakteres dem Anwendungszweck entsprechendes Modell vorgesehen. Von diesem ausgehend kann das Systemmodell dann in eine UML-Repräsentation transformiert werden, welche sich mit bekannten Verfahren und verfügbaren Werkzeugen für UML-basierte Modelle nutzen lässt, etwa um aus UML-State-Machine-Modellen Testfälle generieren zu lassen [SW10]. Generierte wie auch manuell modellierte Testfälle werden in ein UTP-konformes Modell

überführt und können damit durch generische Testwerkzeuge verarbeitet werden. Eine Forschungsfrage dieser Arbeit betrifft das Problem der Transformation von DSL-spezifischen Modelle in UML-Modelle. Es kann sein, dass die Semantik oder die modellierbaren Informationen einer DSL nicht vollständig auf UML und UTP abgebildet werden kann. Dies kann insbesondere dann vorkommen wenn auch nicht-funktionale Eigenschaften oder Anforderungen modelliert werden können. Hier soll untersucht werden, ob und wie eigenständig verwendbare UML-Repräsentierungen möglich sind und welche Erweiterung speziell des UTP-Testmodells notwendig sind, um den Informationsverlust zu minimieren.

3 Stand der Arbeiten

Diese Arbeit ist in ein Kooperationsprojekt (Hessen ModellProjekte, LOEWE 3) zwischen der Hochschule RheinMain und der Eckelmann AG in Wiesbaden eingebettet. Ziel des Projekts ist die Entwicklung eines Testframeworks mit dem modellbasiert Integrations- und Systemtests entworfen und automatisiert durchgeführt werden können. Die schon während der Projektlaufzeit vorgesehene Evaluierung der Ansätze und (iterative) Realisierung des Testframeworks durch ausgesuchte Entwicklungsgruppen des industriellen Kooperationspartners soll die Praxisrelevanz und Einsetzbarkeit sicherstellen.

Im Rahmen des Forschungsprojekts werden zwei Anwendungsfälle aus unterschiedlichen Bereichen betrachtet, deren Anforderungen an das Testframework sich unterscheiden. Der Fokus des ersten Anwendungsfalls liegt auf der Automatisierung von Tests für einzelne Steuergeräte, wohingegen der Fokus des zweiten Anwendungsfalls auf dem modellbasierten Testen auf Basis existierender System- und für das Unternehmen angepasster Testmodelle liegt. Für Ablaufsteuerungen wurden beim Kooperationspartner hausintern über die letzten 20 Jahre eine eigene, tabellenbasierte Notation für Endliche Automaten und darauf aufbauende Modellierungs- und Codegenerierungswerkzeuge entwickelt.

Bisher wurden im Rahmen des Projekts die Anforderungen an das Testsystem und die Testdefinitionen gesammelt und die bei Eckelmann existierenden Systemmodelle mit den hausintern entwickelten Modellierungswerkzeugen untersucht. Basis für die Testautomatisierung ist das Eclipse Projekt TPTP („Eclipse Test & Performance Tools Platform“) [TPT13], das auf den ersten Anwendungsfall adaptiert wurde.

Für die Integration der Eckelmann-eigenen Systemmodelle wurde eine Abbildung auf ein abstrakteres EMF-Metamodell erstellt und mittels QVT-Transformation in Eclipse umgesetzt. Aus diesem EMF-Metamodell kann bereits über eine weitere QVT-Transformation ein UML-State-Machine-Modell erzeugt werden. Diese ersten Schritte erlauben es, Erfahrungen mit den Werkzeugen und Möglichkeiten von Eclipse auf der einen und den Eigenschaften des Systemmodells von Eckelmann auf der anderen zu sammeln.

Im Weiteren sollen für den zweiten Anwendungsfall ein Testmodell und Testfallbeschreibungen in Form von angepassten DSLs entwickelt werden, die die existierenden Gegebenheiten bei Eckelmann vor allem für den Bereich der Ablaufsteuerungen gut widerspiegeln.

Literatur

- [IEC10] IEC. 61508-3 Functional safety of electrical/electronic/programmable electronic safety-related systems, 2010.
- [Lig09] Peter Liggesmeyer. *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. Spektrum Akademischer Verlag, Heidelberg, 2.. Auflage, 2009.
- [MOSHL09] Mohamed Mussa, Samir Ouchani, Waseem Al Sammane und Abdelwahab Hamou-Lhadj. A Survey of Model-Driven Testing Techniques. In *9th International Conference on Quality Software, QSIC '09.*, Seiten 167–172, Jeju, Korea (South), August 2009. IEEE.
- [OMG05] OMG. UML Testing Profile, 2005.
- [OMG11] OMG. OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1, 2011.
- [RBGW10] Thomas Roßner, Christian Brandes, Helmut Götz und Mario Winter. *Basiswissen Modellbasierter Test*. Dpunkt.Verlag GmbH, 2010.
- [Sch12] Ina Schieferdecker. Model-Based Testing. *IEEE Software*, 29(1):14–18, Januar 2012.
- [SL10] Muhammad Shafique und Yvan Labiche. A Systematic Review of Model Based Testing Tool Support. Technical report SCE-10-04, Carleton University, Mai 2010.
- [SVEH07] Thomas Stahl, Markus Völter, Sven Efftinge und Arno Haase. *Modellgetriebene Softwareentwicklung: Techniken, Engineering, Management*. Dpunkt.Verlag GmbH, 2.. Auflage, 2007.
- [SW10] Dehla Sokenou und Stephan Weißleder. ParTeG - Integrating Model-based Testing and Model Transformations. In Gregor Engels, Markus Luckey und Wilhelm Schäfer, Hrsg., *Software Engineering 2010 - Fachtagung des GI-Fachbereichs Softwaretechnik, 22.-26.2.2010 in Paderborn*, Jgg. 159 of *LNI*, Seiten 23–24. GI, 2010.
- [TPT13] Eclipse TPTP. Eclipse Test & Performance Tools Platform Project. <http://www.eclipse.org/tptp/>, Januar 2013.

Universelle Traversierung von NAT-Geräten*

Alexander Diener[†]

Thomas Zink, Oliver Haase, Jürgen Wäsch

Fakultät Informatik

HTWG Konstanz

Brauneggerstrasse 55

D-78462 Konstanz

diener.alexander@gmx.de

{tzink, haase, waesch}@htwg-konstanz.de

Abstract: Der Verbindungsaufbau in Peer-to-Peer Systemen über NAT-Geräte hinweg ist ohne den Einsatz spezieller Techniken nicht möglich. Der hier beschriebene Connectivity Manager führt automatisch eine Untersuchung der NAT-Situation, die Auswahl geeigneter NAT-Traversierungstechniken und die Herstellung einer Verbindung durch. In einer Evaluation konnte gezeigt werden, dass in jeder NAT-Situation erfolgreich eine Verbindung hergestellt werden kann.

1 Einleitung und Motivation

Durch den Einsatz von Network Address Translation (NAT) wird das Ende-zu-Ende Prinzip des Internets gebrochen und nicht jeder Rechner ist direkt adressierbar und somit öffentlich erreichbar. Besonders in Peer-to-Peer Systemen ist dies ein großes Problem, da sich hier die Peers meistens hinter einem NAT-Gerät befinden.

Um die Erreichbarkeit von Peers hinter einem NAT-Gerät trotzdem zu ermöglichen, wurden verschiedene NAT-Traversierungstechniken vorgeschlagen. Der Erfolg einer solchen Technik ist jedoch abhängig von den NAT-Geräten, die es zu durchdringen gilt [FSK05, MHWW11]. Um zu entscheiden welche Technik sinnvoll eingesetzt werden kann sind fundierte Kenntnisse über die NAT-Situation erforderlich, die meist nicht vorhanden sind. Aus diesem Grund ist ein Verfahren notwendig, das die NAT-Situation untersucht und geeignete NAT-Traversierungstechniken transparent für die Anwendung auswählt. Der im Rahmen einer Master Thesis [Die12] konzipierte und implementierte Connectivity Manager für das Universal Connection Establishment Verfahren kann in jeder NAT-Situation erfolgreich eine Verbindung zwischen Peers herstellen.

*Die Arbeit wurde im Kontext des FHprofUnt-Projektes "Transparente Integration von NAT-Traversierungstechniken in Java" durchgeführt, das vom BMBF und der Seitenbau GmbH in Konstanz gefördert wird. Projektpartner sind die HTWG Konstanz, Seitenbau GmbH und die Universität Konstanz.

[†]Bis 31.12.2012 Student im Master-Studiengang Informatik der HTWG Konstanz; seit 01.01.2013 tätig bei der HighCoordination GmbH in Überlingen am Bodensee.

Zunächst wird, falls verfügbar, mit einer Fallback-Technik eine Verbindung hergestellt. Sie funktioniert in jeder Situation und dadurch können die Peers ohne Verzögerung mit dem Senden von Daten beginnen. Parallel dazu wird die aktuelle NAT-Situation untersucht und mittels eines Entscheidungsbaumes die geeigneten NAT-Traversierungstechniken ausgewählt. Diese ungeordnete Menge wird sortiert und das Ergebnis sequentiell abgearbeitet, bis eine direkte Verbindung zu Stande gekommen ist. Ist dies der Fall, wird die initial hergestellte Fallback-Verbindung durch die neue günstigere Verbindung ausgetauscht.

2.3 Entscheidungsbaum

Ein Entscheidungsbaum ermöglicht es, für eine gegebene NAT-Situation geeignete NAT-Traversierungstechniken zu ermitteln. Abbildung 2 zeigt einen beispielhaften Entscheidungsbaum.

Basis für den Aufbau und die Generierung des Entscheidungsbaumes ist eine Menge von Trainingsdatensätzen, die von den NAT-Traversierungstechniken selbst geliefert werden. Diese beschreiben bei welchem NAT-Verhalten sie erfolgreich eine Verbindung aufbauen. Dadurch ist es möglich, nachträglich Techniken hinzuzufügen und den Baum entsprechend anzupassen.

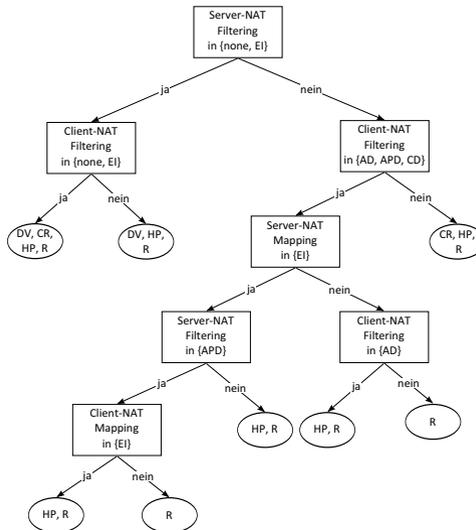


Abbildung 2: Beispielhafter Entscheidungsbaum für die NAT-Traversierungstechniken Connection Reversal (CR), Hole Punching (HP), Relaying (R) und einer Verbindung ohne spezielle Technik (DV) mit den in Abschnitt 2.1 beschriebenen Verhaltensweisen.

2.4 Bewertung und Sortierung

Die Sortierung der NAT-Traversierungstechniken erfolgt zunächst nach direkter und indirekter Verbindung, wobei Techniken mit direkter Verbindung bevorzugt werden. Anschließend erfolgt eine Sortierung aufsteigend nach der Dauer des Verbindungsaufbaus. Diese Liste wird sequentiell abgearbeitet, bis eine Verbindung zwischen den Peers zu Stande gekommen ist.

3 Implementierung und Evaluation

Die Implementierung des Connectivity Managers erfolgte in Java und steht unter GNU GPLv3 zur Verfügung (<https://github.com/htwg/UCE>). Details zur Softwarearchitektur und Implementierung sind in [Die12] zu finden.

Eine Evaluation hat gezeigt, wie sich der Connectivity Manager in einer möglichst realen Umgebung verhält. Dazu wurde eine Testumgebung entwickelt, in der zwei Peers Nachrichten austauschen. Die NAT-Geräte, hinter denen sich die Peers befinden, wurden mit `iptables` konfiguriert. Für den Verbindungsaufbau nutzten beide Peers den Connectivity Manager. In allen evaluierten NAT-Situationen konnte eine Verbindung hergestellt werden [Die12].

4 Fazit

Der Connectivity Manager macht es möglich, ohne Kenntnisse über NAT und deren Durchdringung Verbindungen zwischen Peers herzustellen. Die Implementierung ist so angelegt, dass das Verfahren erweiterbar und für eine Anwendung transparent ist.

Literatur

- [Die12] A. Diener. Konzeption und Implementierung eines Connectivity Manager für das Universal Connection Establishment Verfahren. Master Thesis, HTWG Konstanz, November 2012.
- [FSK05] B. Ford, P. Srisuresh und D. Kegel. Peer-to-peer communication across network address translators. In *Proc. of the USENIX Annual Technical Conference*, 2005.
- [MHWW11] D. Maier, O. Haase, J. Wäsch und M. Waldvogel. NAT hole punching revisited. In *Proc. of the 36th IEEE Conference on Local Computer Networks (LCN)*, 2011.
- [ML10] D. MacDonald und B. Lowekamp. NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN). IETF, RFC 5780 (Experimental), 2010.
- [SFK08] P. Srisuresh, B. Ford und D. Kegel. State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs). IETF, RFC 5128 (Informational), 2008.

Quelltextgenerierung für LEGO Mindstorms-Roboter

Johannes Geismann geismann@campus.uni-paderborn.de
Fraunhofer-Institut für Produktionstechnologie IPT
Projektgruppe Entwurfstechnik Mechatronik, Softwaretechnik
Zukunftsmeile 1, 33102 Paderborn

Abstract: Die Komplexität von mechatronischen Systemen wird stetig größer. MechatronicUML (MUML) ist eine Methode zur Entwicklung für Software von mechatronischen Systemen. Im Rahmen einer Bachelorarbeit wurde ein bestehender Quelltextgenerator für MUML-Modelle so erweitert, dass Quelltext für ein Echtzeitbetriebssystem generiert werden kann, welches auf einem LEGO Mindstorms-Roboter installiert ist.

Die Bedeutung von mechatronischen Systemen wird stetig größer. Auch im alltäglichen Gebrauch finden sie immer mehr Verwendung. Viele solcher Systeme müssen sicherheitskritische Aufgaben übernehmen, wie zum Beispiel die Steuerung eines automatischen Bremssystems eines Fahrzeugs. Neben einem Sensor, der eine bevorstehende Kollision melden kann, braucht es einen Mechanismus, der schnell genug entscheidet, welche Motoren wie reguliert werden müssen. Hierfür wird geeignete Software benötigt. Ein weiteres Beispiel für mechatronische Systeme sind autonome Roboter. Sie verfügen über Sensorik, Aktorik und Steuereinheiten, auf denen ihre Steuerungsintelligenz als Software läuft. Häufig besteht ein System nicht nur aus einer einzigen Steuereinheit (ECU) mit einer CPU, sondern aus vielen ECUs und Teilsystemen, die stark miteinander vernetzt sind. Daher wird es immer schwieriger, sicherheitskritische Anforderungen zu überprüfen.

Modellgetriebene Softwareentwicklung (MDSE) kann helfen, diesen Problemen entgegenzuwirken. Bei MDSE dient das Modell als formale Beschreibung und als Grundlage zum Generieren lauffähiger Programme [Bal05]. Dieses Konzept der Softwareentwicklung bietet Vorteile bei der Entwicklung großer Systeme: Durch MDSE können zum einen durch sogenanntes *Model Checking* [CGP00] schon früh im Entwicklungsprozess bestimmte Eigenschaften des Gesamtsystems oder eines Teilsystems bewiesen oder ausgeschlossen werden, sodass die gewünschte Sicherheit für das Verhalten gegeben ist. Zum anderen kann beim Entwicklungsprozess stark von Details abstrahiert und so bessere Übersicht gewährleistet werden. Außerdem bietet MDSE den Vorteil, dass einmal modellierte Softwaremodelle für andere Softwaresysteme wiederverwendet werden können [SV05].

Im Rahmen einer Bachelorarbeit im Bereich Softwaretechnik in der Projektgruppe Entwurfstechnik Mechatronik am Fraunhofer Institut IPT wurden Konzepte für eine Quelltextgenerierung für ein Echtzeitbetriebssystem (RTOS) eines LEGO Mindstorms-Roboters entwickelt. Die Arbeit fand im Rahmen des Verbundprojektes itsowl-IV „Intelligente Vernetzung“ des Spitzenclusters Intelligente Technische Systeme OstWestfalenLippe (it's OWL) statt. Eine Methode zur Entwicklung der Software für diese Systeme ist die Mechatronic-UML (MUML) [BBB⁺12]. Diese entstand im Sonderforschungsbereich 614 in Kooperation zwischen Regelungstechnikern, Ingenieuren und Informatikern unter Leitung der

Fachgruppe für Softwaretechnik der Universität Paderborn. MUML ermöglicht es, die verschiedenen Softwarekomponenten eines Systems, ihr Verhalten, ihre Kommunikation und ihre spätere Verteilung auf Hardware zu beschreiben. Im Entwicklungsprozess von MUML ist außerdem das Verifizieren von zeitbehafteten Verhaltenseigenschaften und das automatische Generieren von Quelltext aus den Modellen für die entsprechende Plattform vorgesehen. Damit die Vorteile der modellgetriebenen Softwareentwicklung in ausführbare Software umgesetzt werden können, wird eine spezifische Model-To-Text (M2T) Transformation benötigt. Zur Zeit existieren MUML-Transformationen für verschiedene Programmiersprachen und Simulationsumgebungen, welche allerdings noch nicht auf die Spezifika von (Echtzeit)-Betriebsystemen eingehen. Dies bringt neue Herausforderungen wie die Aufteilung in Anwendungsprozessen oder Scheduling-Verhalten mit sich.

Das Ziel der Arbeit war es, die bestehende Transformation von in MUML spezifizierten Softwaremodellen nach C für die Nutzung auf einem RTOS des Mindstorms-Roboters zu erweitern und zu evaluieren. Mindstorms dient als Beispielanwendung eines mechanischen Systems. Als RTOS wurde das zum Standard OSEK OS [OSE05] konforme `nxtOSEK`¹ verwendet, das es ermöglicht, Programme in C-Quelltext auszuführen. Für die Arbeit wurde die bestehende M2T-Transformation des `EmbeddedModeller` der `SOLUNAR GmbH`² erweitert. Mit dieser Entwicklungsumgebung kann ein Modell der Software mit MUML erstellt und dann aus diesem C-Quelltext generiert werden, der zum Testen der Software in einer Simulationsumgebung genutzt werden kann [PRR⁺12].

Das entwickelte Konzept beinhaltet die Übersetzung der Systemmodelle in lauffähigen Quelltext für `nxtOSEK`. Es benötigt für die Ausführung von Anwendungsprozessen sogenannte Tasks. Da OSEK OS ein Standard für *statische* Echtzeitbetriebssysteme ist, müssen alle Systemobjekte, wie Tasks, und deren Eigenschaften, wie Prioritäten, schon während der Implementierung festgelegt werden. Dazu wurde ein Konzept entwickelt, wie Systemkomponenten und deren Zustandsautomaten in eine Taskstruktur übersetzt werden können, das heißt, welche Modellelemente auf einen Task abgebildet werden sollen. Außerdem wurden ein Prioritätenschema für die Tasks und eine Ausführungsstrategie auf Basis dieser Prioritäten erarbeitet. Abbildung 1 zeigt schematisch das Überführen eines MUML-Komponentendiagramms in Tasks und deren Ausführungsstrategie auf `nxtOSEK`. Das Konzept sieht vor, für jede atomare Komponente [BBB⁺12] einen eigenen Task zu erzeugen. Zusätzlich werden Tasks eingeführt, die für die korrekte und regelmäßige Ausführung der generierten Tasks sowie für die asynchrone Kommunikation von Komponenten nötig sind. Durch höhere Task-Prioritäten wird sichergestellt, dass alle Komponententasks abgearbeitet sind, bevor sie durch einen Hilftask erneut angestoßen werden.

Ergebnis dieser Arbeit ist eine M2T-Transformation, die es ermöglicht, aus Softwaremodellen, die vorher durch ModelChecking verifiziert und in Simulationen getestet werden können, lauffähigen Quelltext für Mindstorms-Roboter zu generieren. Die entwickelte Quelltextgenerierung wurde aufbauend auf dem `EmbeddedModeller` ebenfalls als Eclipse-Plugin umgesetzt und anhand von modellierten Szenarien auf einem Mindstorms-Roboter evaluiert. Das Ziel des einen Szenarios war es, einen Mindstorms-Roboter, der autonom durch ein spezielles Labyrinth fährt, zu koordinieren [PRR⁺12].

¹<http://lejos-osek.sourceforge.net>

²<http://www.solunar.de>

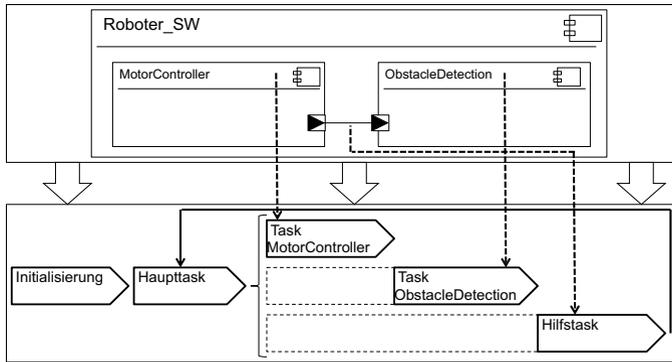


Abbildung 1: Übersetzung von Modellebene in Taskstruktur mit Ausführungsstrategie

Die Evaluierung am Roboter hat gezeigt, dass die Ausführungsgeschwindigkeit für die Evaluierungsszenarien sehr schnell war. Es wurden durchschnittlich 1,2 Überprüfungen der Echtzeitzustandsautomaten pro Millisekunde erreicht. Die begrenzten Speicherplatzanforderungen des Mindstorms wurden eingehalten. Für das Szenario betrug die Größe der ausführbaren Datei 54064 Byte. Der belegte RAM während der Ausführung betrug 5084 Byte. Es stellte sich allerdings heraus, dass die Modellierung nicht gänzlich unabhängig von der später verwendeten Hardware bleiben kann, da Faktoren wie Bauraum oder das Ansprechen der Sensorik Einfluss auf das Verhalten des Roboters hatten. Insbesondere die Sensorik zeigte im Vergleich zur Simulation stark abweichendes Verhalten.

Für diese Arbeit wurden verschiedene Einschränkungen der MUML aus Zeit- und Komplexitätsgründen getroffen. Zum Beispiel wurde kein Zeitverhalten in der Modellierung behandelt. Dies wäre zur Erfüllung von Zeitanforderungen aber notwendig. Weiterhin wurde nur der Fall betrachtet, dass nur für einen einzelnen Mindstorms-Roboter mit einer einzigen ECU Software modelliert und Quelltext generiert wird. Da die von nxtOSEK angebotenen Dienste nicht ausreichen, um dieses Problem zu lösen, wurde als zukünftiger Lösungsansatz eine Middleware vorgeschlagen. Diese vermittelt zwischen System- und Anwendungsschicht und kann somit zum Beispiel helfen, während der Modellierung von der späteren Verteilung der Tasks auf mehrere Steuerungseinheiten zu abstrahieren. Um den Rahmen der Arbeit nicht zu übersteigen, wurde diese Middleware nicht umgesetzt. Es wurden Anforderungen aufgestellt, die eine solche Middleware erfüllen muss, um die bisher nicht betrachteten Konzepte umzusetzen, wie zum Beispiel eine Erweiterung des bestehenden Taskmodells. In zukünftigen Arbeiten wird es interessant sein, die bisher nicht umgesetzten Konzepte zu realisieren und zu evaluieren.

Danksagung

Die Arbeit wurde im Rahmen des Spitzenclusters „Intelligente Systeme OstWestfalenLippe“ entwickelt. Der Spitzencluster wird gefördert vom Bundesministerium für Bildung und Forschung (BMBF). Ich danke der SOLUNAR GmbH für ihre Unterstützung und die Bereitstellung des EmbeddedModellers. Besonders danke ich Renate Löffler und Uwe Pohlmann für die konstruktiven Diskussionen und hilfreichen Ratschläge.

Literatur

- [Bal05] Helmut Balzert. *Lehrbuch der Softwaretechnik - Basiskonzepte und Requirements Engineering*. Spektrum, 2005.
- [BBB⁺12] S. Becker, C. Brenner, C. Brink, S. Dziwok, R. Löffler, C. Heinzemann, U. Pohlmann, W. Schäfer, J. Suck und O. Sudmann. The MechatronicUML Method - Process, Syntax, and Semantics. Bericht, Software Engineering Group, Heinz Nixdorf Institute, University of Paderborn, 2012.
- [CGP00] Edmund M. Clarke, Orna Grumberg und Doron A. Peled. *Model Checking*. MIT Press, 2000.
- [OSE05] OSEK. OSEK Operating System, February 2005.
- [PRR⁺12] Uwe Pohlmann, Hendrik Reddehase, Jens Röckemann, Wilhelm Schäfer und Robert Wagner. Generating Functional Mockup Units from Software Specifications. In *Modelica Conference*, 2012.
- [SV05] Thomas Stahl und Markus Völter. *Modellgetriebene Softwareentwicklung*. dpunkt, 2005.

FASL: Skriptsprache zur Programmierung mobiler Geräte

Manuel Grandeit, Christoph Romstück, Patrick Kreutzer
fasl@i2.cs.fau.de

Art der Arbeit: Projektarbeit („Mobile Application Development“)

Betreuer der Arbeit: A. Kumlehn, M.Sc., Dipl. Inf. D. Brinkers, Dipl. Inf. T. Werth

Abstract: Die Sprache *FASL* ermöglicht eine einfache Programmierung und Ausführung von Skripten auf mobilen Geräten wie Smartphones oder Tablets. Die Sprache ist speziell für den Einsatz auf diesen Plattformen konzipiert und bietet daher diverse Vorteile gegenüber anderen auf mobile Systeme portierte Sprachen. Ein wichtiger Teil des Projekts ist eine Entwicklungsumgebung, die eine komfortable Möglichkeit zur Code-Erzeugung auf mobilen Geräten bietet.

1 Zielsetzungen des Projekts

Tragbare Geräte sind heutzutage ständige Begleiter und wichtige Kommunikationsgeräte. Sie sind daher als Entwicklungsplattform besonders interessant, weil sie völlig neue Ideen für kleine Programme liefern. Bisher bestehen allerdings nur unzureichende Möglichkeiten „direkt“ auf diesen Geräten zu programmieren. Die „klassische“ Programmierung mittels eines Texteditors ist mit Touchscreen, kleinem Display und fehlender Hardware-Tastatur allerdings unkomfortabel.

Das Ziel des Projekts ist die Schaffung einer Möglichkeit zur Programmierung und Ausführung von Skripten auf mobilen Geräten wie Smartphones oder Tablet-Rechnern. Wichtig ist dafür insbesondere die Entwicklung einer Sprache, die eine komfortable Code-Eingabe trotz hardwarebedingter Einschränkungen ermöglicht. Die entwickelte Programmiersprache *FASL* bietet ein erweiterbares API, das den Zugriff auf die Dienste des mobilen Geräts (z.B. Kalender, Adressbuch, GPS) erlaubt. *FASL* erlaubt das mobile Schreiben von Skripten für kleinere Aufgaben (z.B. Lesen eines RSS-Feeds, Überprüfung aller Kontakte auf Geburtstag), unabhängig von Desktop-Rechnern und komplexen APIs.

Als Zielplattform wurde *Android* gewählt, da es weit verbreitet und Open-Source ist. Das Android-API bietet die notwendigen Schnittstellen für die Funktionen eines mobilen Geräts (z.B. Versenden einer SMS, Zugriff auf Kontaktdaten). Die *FASL*-App benötigt daher viele Berechtigungen (siehe dazu auch Abschnitt 2.2). Um potenziellem Misstrauen vorzubeugen wird das Projekt unter einer Open-Source-Lizenz veröffentlicht [KRG13]. Außerdem werden durch den gewährten freien Zugang zu den Quelltexten mögliche Erweiterungen und Verbesserungen vereinfacht. Zum Beispiel könnten durch eine Portierung des Projekts auf andere Plattformen die in der Programmiersprache *FASL* entwickelten Skripte dort ebenfalls genutzt werden, da sie grundsätzlich plattformunabhängig sind.

```

import "System" as sys; import "Contacts" as ctc; import "DateUtils" as du;
func main()
  cs = ctc:getContacts();
  foreach (c in cs)
    if (du:isAnniversaryToday(c.birthday))
      sys:showNotification(c.name + " hat Geburtstag", du:format(c.birthday));
    end
  end
end

```

Abbildung 1: Zeigt für alle Kontakte, die heute Geburtstag haben, eine Benachrichtigung an.

2 Die Programmiersprache FASL

2.1 Sprachbeschreibung

FASL bietet die aus imperativen Programmiersprachen bekannten Sprachkonstrukte wie Schleifen, Verzweigungen und Funktionen. Obwohl FASL stark typisiert ist, müssen Variablen nicht vor ihrer ersten Verwendung deklariert werden. Sie gelten ab der ersten Zuweisung als deklariert. Ihr Typ wird mittels Inferenz während der Übersetzung bestimmt [ALSU06]. FASL bietet die Datentypen `num` für numerische Fließkomma-Werte, `String` für Zeichenketten und `bool` für Wahrheitswerte. Zusätzlich gibt es assoziative Arrays, in denen jeder Wert eines „nativen“ Datentyps als Schlüssel verwendet werden kann. Das Sprachkonstrukt der `foreach`-Schleife erlaubt ein einfaches Iterieren über alle Elemente eines Arrays in definierter Reihenfolge. Des Weiteren bietet FASL die Möglichkeit zur Definition von Verbunddatentypen (*Records*). Die Abbildung 1 zeigt ein Beispielprogramm. Eine vollständige Sprachbeschreibung sowie weitere Materialien zu dem Projekt sind in [KRG13] zu finden.

2.2 Übersetzung und Ausführung

FASL-Skripte liegen in der entwickelten Android-App stets als Quellcode vor. Der integrierte Übersetzer transformiert das Programm daher zunächst in einen abstrakten Syntax-Baum (AST) [ALSU06]. Mehrere Analysephasen extrahieren anschließend die erforderlichen Typinformationen und stellen die semantische Korrektheit des Programms sicher. Die dabei gewonnenen Informationen werden auch von der grafischen Entwicklungsumgebung genutzt, um dem Programmierer genauere Vorschläge – beispielsweise zu Variablen- oder Funktionsnamen – anbieten zu können. Die Ausführung der Skripte übernimmt anschließend ein Interpreter, der direkt auf den Datenstrukturen des annotierten ASTs arbeitet.

2.3 Bibliotheken und Rechteverwaltung

FASL kann nicht direkt auf Geräte-Eigenschaften (z.B. SMS, Kontakte, GPS) zugreifen, damit FASL-Skripte plattformunabhängig sind. Der Zugriff erfolgt deswegen über integrierte Java-Bibliotheken, die eine einfache, kompakte und portable API bieten. Die Dokumentation der Funktionen für den Editor ist per Java-Annotationen geregelt.

Bei der Installation der App müssen alle Berechtigungen angefordert werden, die von einer der in den Bibliotheken verwendeten Methoden aus der Android-API benötigt werden [ANDR13]. Um dem Benutzer dennoch eine feingranulare Kontrolle über die Rechte eines Skripts zu bieten, implementiert die FASL-App ein eigenes Konzept für Berechtigungen, über welches jedes Skript eine Untermenge der Rechte anfordern kann. Ferner wird deren Vorhandensein bei jedem Aufruf einer Bibliotheksfunktion überprüft.

3 Die Entwicklungsumgebung

Statt wie in einem herkömmlichen Text-Editor zeichenweise zu editieren, baut der FASL-Editor auf ganze, vom Lexer generierte Token auf. Dies ermöglicht eine komfortable Positionierung der Eingabemarke auf (kleinen) Touchscreens. Der Code-Editor bietet kontextabhängig die einzufügenden Tokens in einer tabellarischen Darstellung an (anstatt einer herkömmlichen Tastatur). Dazu zählen neben den Schlüsselwörtern auch häufig benötigte Sonderzeichen, die auf Touchscreen-Tastaturen meist nur umständlich zu erreichen sind. Neben der komfortablen Eingabe von Strings und Zahlenwerten, beherrscht der Editor außerdem automatische oder manuelle Formatierungen sowie eine Undo/Redo-Funktionalität.

Des Weiteren bietet der Editor Dialoge zur Vereinfachung häufig auftretender Arbeitsschritte an (sog. *Wizards*), z.B. automatische Vervollständigung zum Import von Bibliotheken. Ein anderes Beispiel ist das Einfügen eines Funktionsaufrufs: der Wizard zeigt eine Liste aller aufrufbaren Funktionen an (samt Parameternamen und -typen sowie Dokumentation).

4 Verwandte Arbeiten

TouchDevelop von Microsoft ist für viele Plattformen verfügbar und verwendet ebenfalls eine auf mobile Geräte angepasste Sprache und Entwicklungsumgebung. Allerdings bietet es nur auf Geräten mit Windows Phone Zugriff auf alle Eigenschaften des Geräts¹ und ist nicht von Drittpersonen erweiterbar [TMHF11]. Das Projekt *android-scripting* für Android versteht sich als Infrastruktur, um verschiedene bereits existierende Interpreter auf dem mobilen Gerät zu verwenden, bietet allerdings keine auf Touchscreens oder kleine Displays angepasste Oberfläche [JG11]. Das für Android verfügbare *DroidDevelop* ermöglicht das Entwickeln kompletter Apps, ist aber auf die klassische Bedienung mittels Tastatur ausgelegt und bietet keine Abstraktion der Gerätefunktionalitäten [DD13]. Es gibt eine Reihe von integrierten Entwicklungsumgebungen, die es erlauben, direkt auf Android-Geräten Apps zu entwickeln [TIDE13, AIDE13, JIDE13]. Dabei wird statt einer speziell angepassten Sprache das Android-SDK verwendet, das für einfache Skripte übertrieben kompliziert ist. Außerdem ist die Oberfläche meist nur auf größeren Geräten, z.B. Tablets, komfortabel.

¹Die Umsetzung für andere Plattformen erfolgt im Web-Browser.

5 Zusammenfassung und Ausblick

Die prototypisch umgesetzte Android-App bietet eine intuitiv bedienbare Entwicklungs- und Ausführungsumgebung zur Programmierung von FASL. Dabei können vor allem kleine Skripte für Alltagsaufgaben schnell und einfach dank der einzigartigen Token-basierten Eingabe direkt auf dem mobilen Gerät umgesetzt werden. FASL ist damit verwandten Projekten, die ebenfalls die Programmierung mobiler Geräte als Ziel haben, überlegen.

FASL soll (mit Hilfe anderer Open-Source-Entwickler) auf weitere mobile Systeme portiert werden. Des Weiteren ist das Einrichten eines öffentlichen Verzeichnisses für Skripte angedacht. Zudem werden wir die Ausführung von Skripten komfortabler gestalten, indem z.B. Skripte regelmäßig oder ereignisbasiert gestartet werden. Um die Programmierung auf Touchscreens noch weiter zu erleichtern, ist eine kontextabhängige Hilfe zur Auswahl von Variablen und Funktionen (z.B. durch das Ausnutzen von Typinferenz) geplant.

Literatur

[ALSU06] A. Aho, M. Lam, R. Sethi, J. Ullman, *Compilers: Principles, Techniques, and Tools* Addison-Wesley, 2006

[ANDR13] Android API Guides, *Permissions*, 10.01.2013,
<http://developer.android.com/guide/topics/security/permissions.html>

[KRG13] P. Kreutzer, C. Romstöck, M. Grandeit, *FASL Sprachbeschreibung*, 24.01.2013,
<http://mad.cs.fau.de/fasl>

[TMHF11] N. Tillmann, M. Moskal, J. de Halleux, M. Fahndrich, *TouchDevelop - Programming Cloud-Connected Mobile Devices via Touchscreen*, no. MSR-TR-2011-49, 2011

[JG11] L. Jordan, P. Greyling, *Practical Android Projects*, Apress, 2011

[DD13] AsSoft, *DroidDevelop*, 24.01.2013,
<https://play.google.com/store/apps/details?id=com.assoft.DroidDevelop>

[TIDE13] Spartacus Rex, *Terminal IDE*, 25.01.2013,
<https://play.google.com/store/apps/details?id=com.spartacusrex.spartacuside>

[AIDE13] appfour GmbH, *AIDE - Android Java IDE*, 25.01.2013,
<https://play.google.com/store/apps/details?id=com.aide.ui>

[JIDE13] Tanapro GmbH, T. Arn, *JavaIDEdroid*, 25.01.2013,
https://play.google.com/store/apps/details?id=com.t_arn.JavaIDEdroidPRO

Ein Pattern Katalog für Qualitätsbewertungsmethoden für Software Architekturen

Perihan Bulut, Meryem Cetinkaya, Yasemin Guruhan, Thomas Gutenberger, Philipp
Maier, Buket Oezmen, Mehtap Yilmaz

University of Applied Science Reutlingen
Master - Wirtschaftsinformatik
Alteburgstraße 150
72762 Reutlingen

Vorname.Nachname@Student.Reutlingen-University.de

Art der Arbeit: Projektarbeit

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

Abstract: Für die Qualitätsbewertung von Software Architekturen wurden am Software Engineering Institute verschiedene Verfahren entwickelt. Auf der Basis dieser Verfahren wurde ein Pattern Katalog entwickelt, der alle relevanten Phasen als Pattern beinhaltet.

1 Einführung

Bei der Entwicklung von Software spielen verschiedene Faktoren eine Rolle, wie z.B. die Kosten, die bei der Entwicklung entstehen. Jedoch hat sich in der Praxis erwiesen, dass eine zu enge Betrachtungsweise wie auf Kosten oder Entwicklung von schnellen Lösungen semi-befriedigend sein können, da hier meistens der Aspekt der Qualität zu wenig Beachtung findet. Aus diesem Grund ist es auch wichtig, die Qualitätsmerkmale einer zu entwickelnden oder bestehenden Software zu berücksichtigen. Um die Qualität einer Software Architektur hinsichtlich funktionaler und nicht-funktionaler Anforderungen adäquat zu validieren, bieten sich verschiedene Evaluierungsverfahren an.

Doch warum ist die Evaluation einer Architektur so essentiell? Laut [CKK11] können Fehler bzw. Probleme, die früh in der Entwicklungsphase gefunden werden, mit einem erheblich geringeren Kostenaufwand behoben werden, als dieselben Fehler, welche erst in der Testphase entdeckt werden. Die Architekturevaluation bietet also einen kostengünstigen Weg, um zu einem frühen Zeitpunkt Fehler zu erkennen, zu verbessern oder zu eliminieren und dadurch die richtige Architekturentscheidung zu treffen.

Die Qualitätsbewertungsmethoden, die im Rahmen dieses Projektes untersucht wurden, wurden alle am Software Engineering Institute (SEI) entwickelt:

- SAAM – Software Architecture Analysis Method
- ATAM – Architecture Tradeoff Analysis Method
- ARID – Active Reviews for Intermediate Designs
- QUASAR – A Method for the QUality Assessment of Software-Intensive System Architectures

Das Ziel dieser Arbeit ist es, ein Pattern Katalog für die Evaluation von Software Architekturen zu erstellen auf Basis der Qualitätsbewertungsmethoden.

2 Qualitätsbewertung - Pattern Katalog

Im Abbildung 1 ist ein Auszug des Pattern Katalogs als Strukturbaum zu sehen. Die Bewertungsmethoden bestehen alle aus verschiedenen Schritten, die zur Durchführung nötig sind. Aus diesen Schritten wurden Patterns abgeleitet.

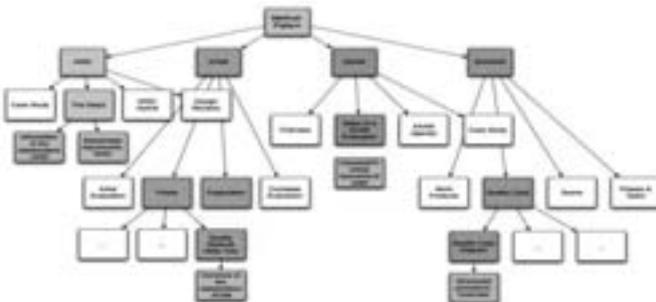


Abbildung 1: Qualitätsbewertungsmethoden – Strukturbaum

In den folgenden Unterabschnitten werden einige ausgewählte Patterns vorgestellt. Das „Quality Attribute Utility Tree Pattern“ bezieht sich auf „ATAM“. Das „QUASAR Pattern“ bezieht sich auf „Structured Procedure“.

2.1 Quality Attribute Utility Tree Pattern

Problem: Wie können abstrakte Qualitätsmerkmale feingranularer dargestellt werden?

Lösung: Die Erstellung eines Nutzenbaums (“utility tree”). Hier werden die Qualitätsattribute und ihre Abhängigkeiten festgelegt, dadurch entsteht eine Baumstruktur. Hierbei werden Szenarien zur Bewertung der Attribute definiert und

anschließend priorisiert bzw. nach Wichtigkeit gruppiert. Der Nutzenbaum besteht aus folgenden Ebenen:

- 1. Ebene: „Utility“ als Wurzelknoten
- 2. Ebene: Beinhaltet alle relevanten Qualitätsmerkmale, wie z.B. Leistung, Modifizierbarkeit, Verfügbarkeit, Sicherheit, etc.
- 3. Ebene: Die Qualitätsmerkmale werden in Kategorien aufgeteilt, d.h. sie werden auf feingranulare Anforderungen heruntergebrochen
- 4. Ebene: Entwicklung von konkreten Szenarien

Output: Priorisierte Anforderungen (Qualitätsmerkmale), die in Form von Szenarien vorliegen.

Vorteil: Hilft bei der Priorisierung und bei der Definierung von konkreten Zielen.

Teilnehmer: Architekturteam, Manager, Kundervertreter.

Erklärung

Warum: Bevor die Architektur beurteilt werden kann, müssen die Systemziele spezifischer und konkreter sein.

→ Der Nutzenbaum dient dazu, die Qualitätsziele zu konkretisieren. Unter jedem Qualitätsmerkmal ist eine spezifische Verfeinerung.

→ Unter jedem Qualitätsmerkmal gibt es spezifische Verfeinerungen, das führt zum Schritt in Richtung verfeinern der Merkmalsziele in Qualitätsmerkmalsszenarien

Wie:

1. Ermittlung der entscheidenden Qualitätsmerkmale
2. Definition der Merkmale über Szenarien
3. Priorisierung der Szenarien: den Szenarien werden Prioritäten (Hoch, Mittel oder Niedrig) und durch den Architekten der notwendige Aufwand zugeordnet.

Output: Aufschlüsselung für die Messbarkeit , Priorisierung für die Einschätzbarkeit

2.2 QUASAR Pattern

Problem: Wie ist es möglich, durch eine Qualitätsbewertung zu gehen, ohne Unterbrechungen?

Lösung: Eine strukturierte Verbindung sind QUASAR Teilnehmer und beinhalten Arbeitsprozesse.

Techniques: Eine unabhängige Sicht auf geteilte Aufgaben geht auf den nächsten Schritt über

Outputs: Qualitätsberichte, Präsentationsunterlagen, Checklisten, Scorecards.

Teilnehmer: Anforderungen-Team, Architekturteam, Assessment Team, Stakeholder

Vorteil: Klare Verfahren, Planung durchsetzen (Budget, Zeitplan, Persona), redundanzfrei, Messen während der Entwicklungsphase

Nachteile: Bürokratisch, hierarchisch, Aufwand von Mitarbeiterarbeit

Erklärung:

- Das Anforderung-Team ermittelt Anforderungsstrategien.
- Das Architekturteam ermittelt Anforderungsstrategien.
- Qualitätsfälle werden strukturell entwickelt und an das Assessment Team weitergegeben.
- Das Assessment Team bewertet die Qualitätsfälle und ermöglicht somit eine klare Berichterstattung.

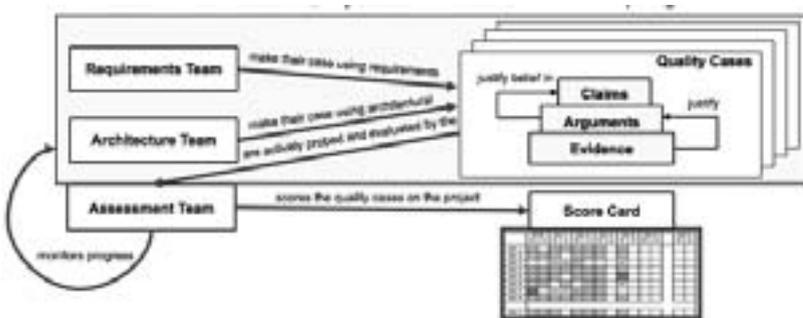


Abbildung 2: Strukturiertes QUASAR Verfahren

3 Fazit und Ausblick

Durch den Pattern Katalog der noch alle Verfahren beinhaltet können im weiteren Verlauf die Gemeinsamkeiten und Unterschiede zwischen den Verfahren besser ermittelt werden. Dadurch ist es möglich ein einheitliches Verfahren zu entwickeln auf Basis eines einzigen Pattern Katalogs.

Literaturverzeichnis

- [CKK11] Clements, P.; Kazman, R.; Klein, M.: Evaluating Software Architectures – Methods and Case Studies. Addison-Wesley Longman, Amsterdam, 2011
- [F06] Firesmith, D.: QUASAR: A Method for the QUality Assessment of Software-Intensive System Architectures: <http://www.sei.cmu.edu/reports/06hb001.pdf>, 2006, abgerufen am 22.12.2012

Reverse Engineering des Windows Phone Process Execution Models

Norman Hansen, Franke Dominik

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

Abstract: Das Lebenszyklusmodell (Process Execution Model) einer mobilen Anwendung beschreibt die Zustände und Transitionen, in der sich eine Anwendung befinden kann. Die korrekte Implementierung hat direkten Einfluss auf die Qualität der Software. Inhalt der in diesem Dokument beschriebenen Bachelorarbeit ist die Rekonstruktion des Lebenszyklusmodells für Windows Phone 7.5 Anwendungen unter Verwendung eines bewährten Reverse Engineering-Ansatzes für Lebenszyklen mobiler Applikationen. Das Ergebnis ist die Anwendbarkeit des Ansatzes auf Windows Phone und die rekonstruierten Lebenszyklusmodelle.

1 Einleitung

Das Lebenszyklusmodell einer mobilen Anwendung, für Windows Phone *Process Execution Model* genannt, beschreibt Zustände und Transitionen, in denen sich eine Anwendung befinden kann. Während der Ausführung einer mobilen Anwendung durchläuft diese meist mehrere dieser Zustände. Die zugrunde liegende Plattform benachrichtigt die Anwendung über die Zustandsänderungen um entsprechend reagieren zu können. Ein Beispiel wäre das Speichern von eingegebenen Daten beim Pausieren einer Anwendung, damit diese auch bei einer Fortführung zur Verfügung stehen. Somit spielt die korrekte Implementierung der Reaktion auf Zustandsänderungen eine wesentliche Rolle für die Qualität mobiler Anwendungen. Meist sind die Lebenszyklusmodelle, die dem Entwickler beschreiben sollen, wie eine Anwendung auf Zustandsänderungen reagieren kann und welche Zustände es gibt, durch Grafiken und beschreibenden Text gegeben. Diese Grafiken und Texte können wegen der meist fehlenden formalen Grundlage verschiedene Interpretationen zulassen oder unvollständig sein. Lebenszyklusmodelle, die diese Problematik beheben, wurden bereits für Android und iOS rekonstruiert (vgl. [FEWK11] und [FEK12]) und werden in diesem Dokument für Windows Phone präsentiert.

Das Dokument ist wie folgt gegliedert: Kapitel 2 beschreibt das von Microsoft dokumentierte Lebenszyklusmodell für die zwei Frameworks Silverlight und XNA, die in Windows Phone verwendet werden. Zudem wird beschrieben, welches Verfahren zum *Reverse Engineering* des Lebenszyklusmodells verwendet wird. Kapitel 3 besteht aus zwei Teilen. Der erste behandelt das durch Reverse Engineering gewonnene Lebenszyklusmodell für Silverlight Anwendungen, während der zweite Teil das entsprechende rekonstruierte XNA Lebenszyklusmodell beschreibt. Kapitel 4 beendet die Arbeit mit einem Fazit.

2 Grundlagen

Dieses Kapitel be-
Phone Anwendun-
satz zum verwend

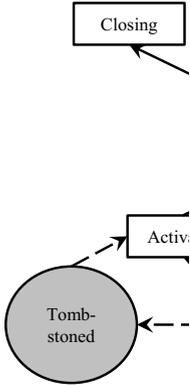


Abbildung 1: Offizi-
Silverlight Anwend

Anwendungen für Windows Phone verwenden entweder das Silverlight oder XNA Framework, oder eine Kombination aus beiden. Das Silverlight Framework wurde ursprünglich zur Entwicklung von Web-Anwendungen und das XNA Framework zur Entwicklung von Anwendungen für bspw. die XBOX 360 verwendet. Abbildung 1 zeigt das offizielle Silverlight Lebenszyklusmodell, bestehend aus 3 Zuständen (*Running*, *Dormant* und *Tombstoned*) und 4 Ereignisse (*Launching*, *Closing*, *Activated*, *Deactivated*), die ausgelöst werden, wenn sich der Zustand einer Anwendung ändert¹. Eine Zustandsänderung von *Dormant* nach *Tombstoned* löst kein Ereignis aus und gibt einer Anwendung somit keine Reaktionsmöglichkeit. Abbildung 2 zeigt eine der offiziellen Darstellungen des XNA Lebenszyklusmodell². Im Gegensatz zu Abbildung 1 sind hier keine Zustände, sondern auftretende Methodenaufrufe angegeben.

Zur Rekonstruktion des Lebenszyklusmodells für Silverlight und XNA Anwendungen wurde der in [FEWK11] vorgestellte Ansatz zum Reverse Engineering von Lebenszyklen mobiler Anwendungen verwendet. Dieser setzt sich aus 4 Schritten zusammen:

- *Implementierung des gesamten Lebenszyklusses*: In diesem Schritt wird eine Anwendung erstellt, die alle Callback-Methoden implementiert, die bei Zustandsänderungen einer Anwendung aufgerufen werden können.

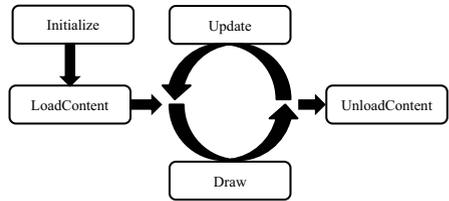


Abbildung 2: Offizielles Lebenszyklusmodell für
XNA Anwendungen

¹Siehe [http://msdn.microsoft.com/en-us/library/ff817008\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/ff817008(v=vs.92).aspx)

²Siehe <http://msdn.microsoft.com/en-us/library/gg490774.aspx>

- *Log Injektion*: In einem zweiten Schritt wird ein Log-Mechanismus implementiert, der den Aufruf jeder Callback-Methode mit einer eindeutigen ID, der Zuordnung zur Anwendung und einem Zeitstempel protokolliert.
- *Identifizierung der Auslöser für Zustandsübergänge*: Der dritte Schritt ist die Bestimmung eines Testkatalogs, der möglichst alle Auslöser für Zustandsänderungen von Anwendungen umfasst. Für Windows Phone ist bspw. das Betätigen der Zurück-Taste nachdem die Anwendung gestartet wurde ein solcher Auslöser. Die Vollständigkeit des Testkatalogs kann nicht garantiert werden, da Testfälle nur auf Basis der Dokumentation und Erfahrung im Umgang mit dem Gerät ermittelt werden können. Nachdem alle Einträge des Testkatalogs ausgeführt und die Ergebnisse gespeichert sind, wird der nächste Schritt umgesetzt.
- *Rekonstruktion des Lebenszyklusmodells*: In Abhängigkeit der protokollierten möglichen Zustandsübergänge wird ein Modell auf Basis einer formalen Notation, bspw. als Mealy-Automat, konstruiert.

3 Reverse Engineering des Windows Phone Execution Models

Dieses Kapitel präsentiert die rekonstruierten Lebenszyklusmodelle für Silverlight und XNA Anwendungen für Windows Phone.

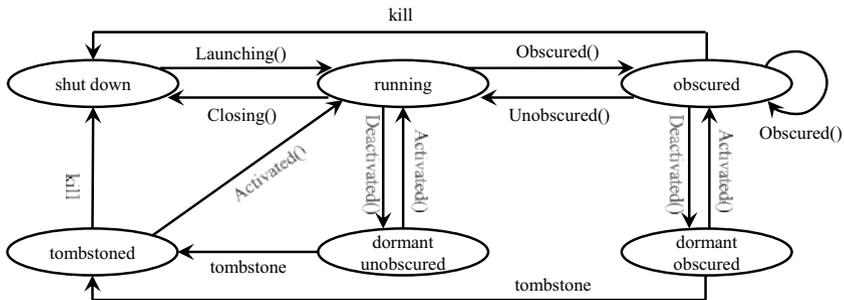


Abbildung 3: Rekonstruiertes Silverlight Lebenszyklusmodell

Das durch Verwendung, des in Kapitel 2 beschriebenen Reverse Engineering, rekonstruierte Silverlight Lebenszyklusmodell ist auf Abbildung 3 zu sehen. Dieses Modell erweitert das Modell von Abbildung 1 um einen *shut down* und einen *obscured*-Zustand, in dem die Anwendung teils verdeckt ist, und entsprechenden Ereignisse (*Obscured* und *Unobscured*), so wie um einen *dormant obscured*-Zustand. Die Zustandsübergänge entlang der mit *kill* und *tombstone* beschrifteten Kanten finden ohne Aufruf von Methoden statt. Dennoch konnten diese Übergänge durch Verwendung des Debuggers aufgrund von Benachrichtigungen über laufende und beendete Prozesse beobachtet werden.

Das mit dem beschriebenen Ansatz rekonstruierte Lebenszyklusmodell für XNA Anwendungen ist auf Abbildung 4 dargestellt. Die Zustände des Modells entsprechen den Zu-

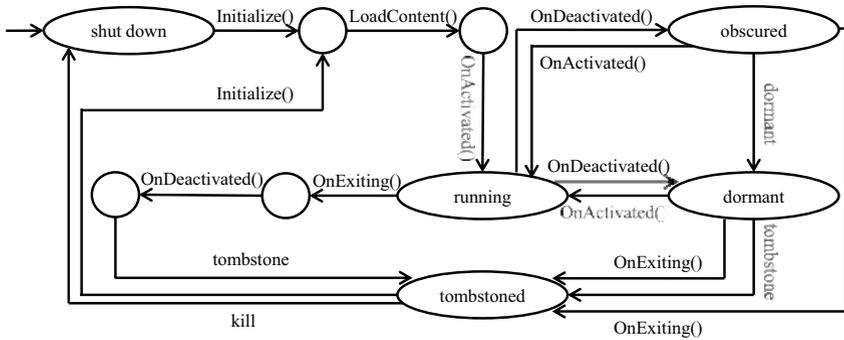


Abbildung 4: Rekonstruiertes XNA Lebenszyklusmodell

ständen im Silverlight-Modell, mit dem Unterschied, dass es für XNA nur einen *dormant*-Zustand gibt. Verglichen mit dem offiziellen XNA-Modell, beinhaltet das rekonstruierte Modell eine Reihe zusätzlicher Callback-Methoden (*OnActivated*, *OnDeactivated*, *OnExiting*), die je nach Zustandsänderung in unterschiedlichen Reihenfolgen aufgerufen werden können. Die Methoden *Update* und *Draw* sind nicht mehr enthalten, da diese nicht in direktem Zusammenhang mit dem Anwendungslebenszyklus stehen. Desweiteren ist es nicht möglich mit dem verwendeten Test-Katalog einen Aufruf der in Abbildung 2 erwähnten *UnloadContent*-Methode zu bewirken.

4 Fazit

Der in [FEWK11] vorgestellte Reverse Engineering Ansatz kann auch zur Rekonstruktion von Windows Phone Lebenszyklusmodellen verwendet werden. Keines dieser rekonstruierten Modelle stimmt mit der offiziellen Dokumentation überein. Dies liegt sowohl daran, dass zusätzliche Zustände und Callback-Methoden identifiziert wurden, als auch, dass in den rekonstruierten Modellen grafisch veranschaulicht wird aus welchen Zuständen eine Anwendung ohne Rückmeldung beendet werden kann.

Literatur

- [FEK12] Dominik Franke, Corinna Elsemann und Stefan Kowalewski. Reverse Engineering and Testing Service Life Cycles of Mobile Platforms. In *2nd DEXA Workshop on Information Systems for Situation Awareness and Situation Management (ISSASiM)*, Seiten 16–20. IEEE Computer Society, 2012.
- [FEWK11] Dominik Franke, Corinna Elsemann, Carsten Weise und Stefan Kowalewski. Reverse Engineering of Mobile Application Lifecycles. In *18th Working Conference on Reverse Engineering (WCRE 2011)*, Seiten 283–292. IEEE Computer Society, 2011.

Systematisches Reengineering eines Softwaresystems am Beispiel einer Sprachtherapie-Software

Daniel Krämer

Hochschule Bonn-Rhein-Sieg
Fachbereich Informatik
daniel.kraemer@smail.inf.h-brs.de

Art der Arbeit: Bachelor-Thesis, eingereicht am 12.09.2012

Betreuer/in der Arbeit: Prof. Dr. Simone Bürsner (simone.buersner@h-brs.de)

Abstract: Als letzte, der eigentlichen Entwicklung nachgelagerte Phase beansprucht die Wartung eines Softwaresystems zumeist einen nicht unerheblichen Teil des gesamten Software-Lebenszyklus. Sind die Grenzen konventioneller Wartungsmaßnahmen erreicht, ermöglicht ein Reengineering durch tiefgreifendere Modifikationen eine weitere Evolution der Software. Dieses Papier geht zunächst der Frage nach, wie ein Reengineering-Projekt durchgeführt werden kann. Anhand eines in der Praxis umgesetzten Projektes wird daran anschließend auf einige Besonderheiten eingegangen, die sich im Kontext eines Reengineerings gewachsener Altsysteme hin zu einer Web-Applikation und der Entwicklung einer gemeinsamen technischen und fachlichen Basis für mehrere, ähnlich gelagerte Projekte ergeben.

1 Disziplin des Reengineerings

Das weite Gebiet des Software Engineerings erstreckt sich über viele Einzeldisziplinen, welche zahlreiche erprobte Methoden für die Phasen der Anforderungserhebung, Analyse, Konzeption, Implementierung und Integration eines Softwaresystems bereitstellen. Softwareprojekte enden in den meisten Fällen aber nicht bereits mit der Auslieferung des Produktes, sondern umfassen nachgelagerte Wartungsarbeiten, die sich über einen langen Zeitraum erstrecken können. In der Praxis zeigt sich immer wieder, dass solche Tätigkeiten langfristig mit einem Strukturverlust des betreffenden Softwaresystems verbunden sind, welcher die Wartungsarbeiten wiederum zunehmend erschwert. Wird schließlich ein Punkt erreicht, an dem eine einfache Wartung nicht mehr möglich ist oder die allgemeine Beschaffenheit des Softwaresystems eine weitere Evolution verhindert, bedarf es zumeist tief greifender Modifikationen. Die Disziplin des Reengineerings nimmt sich diesem Problem an, indem sie für solch fundamentale Veränderungen geeignete Methoden und Vorgehensweisen diskutiert und den beteiligten Entwicklern auf diese Weise einen Leitfaden an die Hand gibt (vgl. [Balz11], S. 540; [DDN08], S. 10ff.; [Mas06], S. 110).

Generell lassen sich Reengineering-Projekte in mindestens zwei abgeschlossene Phasen unterteilen. Die erste Phase wird als Reverse Engineering bezeichnet und widmet sich der (Wieder-)Herstellung für das Projekt relevanter Dokumentationsartefakte des Altsystems. Das übergeordnete Ziel ist dabei die Entwicklung eines tiefgreifenden Verständnisses für das zu überarbeitende System. Daran schließt sich die Phase des Forward Engineerings an, während der auf Basis der Ergebnisse des Reverse Engineerings das Neusystem entwickelt wird (vgl. [Balz11], S. 538; [DDN08], S. 8; [Mas06], S. 110f.).

Es lässt sich festhalten, dass die konzeptionelle Richtung der Entwicklung mit dem Wechsel von der einen zur anderen Phase ins Gegenteil verkehrt wird. In der Phase des Reverse Engineerings wird u.a. versucht, aus dem Quellcode des Altsystems die Softwarearchitektur und das Softwaredesign zu rekonstruieren, um damit gleichermaßen potentielle Probleme aufzufindig machen und die der Software zugrunde liegenden Anforderungen erkennen und nachvollziehen zu können. Diese können zu Beginn der Phase des Forward Engineerings bei Bedarf ergänzt und angepasst werden. Nachfolgend geschieht die zielgerichtete Entwicklung des Neusystems, welche zunächst eine Modellierung der neuen Architektur und des neuen Designs umfasst. Der an dieser Stelle erfolgende Übergang von der Architektur des Altsystems zu der des Neusystems gestaltet sich u.U. sehr komplex, weshalb er je nach Umfang des Projektes eine zusätzliche, dritte Phase darstellen kann. Anschließend erfolgt die Implementierung des Neusystems, welches die im Altsystem zuvor identifizierten Probleme beseitigen soll. Werden die beiden Phasen zusammen betrachtet, so wird schnell deutlich, dass während eines Reengineering-Projektes ein vorhandenes System (in Form seines Programmcodes) unter Zuhilfenahme einiger Zwischenschritte in ein anderes transformiert wird (vgl. [Balz11], S. 538-541; [DDN08], S. 8-12; [Mas06], S. 110f.; S. 113).

2 Reengineering-Projekt im Umfeld einer Sprachtherapie-Software

Im Rahmen der Bachelor-Thesis hat der Autor ein Reengineering-Projekt für eine Software durchgeführt, die seit nunmehr 20 Jahren in der Sprachtherapie Verwendung findet. Mit dem verhältnismäßig hohen Alter des Softwaresystems ging vor allem eine Fülle von Dokumentationsartefakten einher, deren Verlässlichkeit und Relevanz für das Reengineering-Projekt in beschränkter Zeit ermittelt werden mussten. Bedingt durch eine Abwesenheit der ursprünglichen Entwickler stand als ergänzende, verlässliche Informationsquelle neben der ausführbaren Applikation selbst nur der Quelltext zur Verfügung.

Im praktischen Teil der Abschlussarbeit wurde in Form einer Java-Web-Applikation ein evolutionärer Prototyp entwickelt, welcher einige grundlegende Limitierungen der aktuellen Implementierung aufhebt und mit einer überarbeiteten Software-Architektur sowie einem vollständig neu gestalteten Klassendesign die im Laufe der Jahre entstandenen Strukturverluste beseitigt. Eine besondere Herausforderung bestand somit auch in der Abbildung des alten fachlichen Klassen-Modells auf ein neues sowie einer Desktop- auf eine Web-Applikation.

Eine wichtige Besonderheit stellte ferner die Anforderung dar, dass das zu entwickelnde Konzept nicht nur für die Sprachtherapie-Software selbst eingesetzt werden können sollte, sondern auch für einige verwandte Web-Projekte des Unternehmens, welche eine große Menge gemeinsamer funktionaler Anforderungen besitzen. Gemeinsam war allen Projekten vor allem das Ziel, die Benutzer im Kontext eines kognitiven Trainings bei der Bearbeitung zuvor definierter Übungsreihen (bestehend aus einzelnen Übungen verschiedener Form) zu unterstützen. Differenzen gab es hingegen bei der jeweils anvisierten Zielgruppe sowie der damit verbundenen Ausgestaltung einzelner Bestandteile (z.B. Benutzeroberfläche). Auf technischer und funktionaler Ebene wurden somit weitestgehend gleiche Funktionalitäten, wie etwa eine Benutzerverwaltung oder ein Modul zum Durchführen von Übungsreihen, benötigt. Gewünscht war daher eine geeignete Basis für eine Menge gleichartiger Softwareprojekte. Hiermit verbunden war die zusätzliche Erfordernis, bei Konzeption und Implementierung (bis zu einem gewissen Grad) Rücksicht auf die Anforderungen und Rahmenbedingungen dieser Projekte zu nehmen, sodass die entwickelten Resultate sich in alle Systeme gleichermaßen einfügten und die Ausgestaltung weiterer projektspezifischer Anforderungen nicht blockierten.

3 Ergebnisse

Die hinsichtlich eines kognitiven Trainings an den Prototypen gestellten Anforderungen werden von diesem vollständig erfüllt. Einige wesentliche Nachteile der alten Implementierung, wie beispielsweise die hohe Kopplung der Schichten untereinander, weist der neu konzipierte Prototyp nicht auf. Somit eignet er sich als solide, einheitliche Basis für die verwandten Web-Projekte des Unternehmens, welche von der Wiederverwendbarkeit einzelner Komponenten und der damit einhergehenden Arbeitersparnis in der Entwicklung profitieren können. Implementiert wurden desweiteren einige neue Funktionen, die aufgrund der architektonischen Limitierungen im Altsystem nicht umsetzbar waren.

Um den Benutzern des Altsystems den Umstieg zu erleichtern, wurde die grundlegende Interaktion, wie etwa die Durchführung einer Übungsreihe, beibehalten. Auf der anderen Seite erfolgte jedoch eine Anpassung an den Kontext einer Web-Applikation, wie beispielsweise die Realisierung der Navigation. Als besonders kritisch kann die Portierung der Daten eingestuft werden, deren Aufwand sich als enorm herausgestellt hat. Sind das alte und neue Modell nicht deckungsgleich, müssen mitunter komplexe Algorithmen überlegt und implementiert werden, welche die Abbildung bewerkstelligen. Im durchgeführten Projekt wurde für diesen Zweck eine eigene Applikation entwickelt, welche das Therapiematerial des alten Systems in das neue migriert.

Bezüglich des mit dem hohen Alter der Software einhergehenden Umfangs der Dokumentation hat sich eine gestufte Analyse bewährt. In einem ersten Schritt erfolgt zunächst eine Nummerierung und Sichtung der Dokumentationsquellen (im durchgeführten Projekt hauptsächlich Aktenordner).

Hierbei wird vermerkt, welche Dokumente aus welcher Quelle bezogen werden können und welches Alter diese aufweisen. Mehrere ähnliche Dokumente können dabei zu Themenblöcken (z.B. Architektur, Tests) gruppiert werden, deren gemeinsames Alter sich als Zeitspanne angeben lässt. Basierend auf den Zielen des Reengineering-Projektes und dem ermittelten Alter der Dokumente lassen sich diese in einem zweiten Schritt priorisieren. Ausgehend hiervon können nachfolgend punktuell detailliertere und damit zeitaufwendigere Analysen durchgeführt werden. Stehen die ursprünglichen Entwickler nicht mehr für Rückfragen zur Verfügung, so empfiehlt es sich, stattdessen mit den Personen zu sprechen, welche bislang für die Wartung der Software verantwortlich waren. Von ihnen kann man u.U. weitere Informationen erhalten, die aus Quelltext und Dokumentation nicht unmittelbar ersichtlich sind.

Um mit dem zu entwickelnden Neusystem ein gemeinsames technisches und funktionales Fundament für mehrere gleichartige Softwaresysteme zu schaffen, hat sich die frühe Durchführung eines Meetings, in welchem die in den verschiedenen Software-Projekten gleichermaßen verlangten Anforderungen gesammelt werden, als nützlich erwiesen. Auf Grundlage der zusammengetragenen Informationen lässt sich eine minimale Basis entwerfen, welche zu einem späteren Zeitpunkt auf die verschiedenen Projekte angepasst werden kann. Eine lose Kopplung vereinfacht in diesem Zusammenhang eine Austauschbarkeit einzelner Komponenten, wie etwa der Benutzeroberfläche. Darüber hinaus können die gemeinsam genutzten Bestandteile ggf. zentral gestaltet und gewartet werden. Der im Zuge der Bachelor-Thesis vom Autor entwickelte Prototyp dient nun als technische und funktionale Basis für eine Menge überwiegend gleichartiger, im Detail aber verschiedener Softwaresysteme. Eine erste Anpassung für ein konkretes Projekt hat gezeigt, dass zwar durchaus noch gewisse Überarbeitungen einzelner Bestandteile (z.B. Modell-Klassen) von Nöten sind, der Fokus der weiteren Entwicklung aber überwiegend auf den projektspezifischen Anforderungen liegen kann.

Insgesamt hat das Projekt als solches die Erkenntnisse anderer Entwickler (vgl. Reengineering Patterns in [DDN08]) deutlich bestätigt. Bekräftigt werden kann auch die Relevanz der Disziplin des Reengineerings für die kommerzielle Softwareentwicklung, weil innerhalb des betrachteten Unternehmens weitere Systeme existieren, für die langfristig ein Reengineering angedacht ist. Ohne methodische Grundlagen lassen sich derart umfangreiche Vorhaben allerdings nur mit hohen Risiken durchführen.

Literaturverzeichnis

- [Balz11] Balzert, H.: Lehrbuch der Softwaretechnik, 3. Auflage, Spektrum Akademischer Verlag, Heidelberg 2011.
- [DDN08] Demeyer, S.; Ducasse, S.; Nierstrasz, O.: Object-Oriented Reengineering Patterns. First Open Source Edition, Square Bracket Associates, Schweiz 2008.
- [Mas06] Masak, D.: Legacysoftware, Springer-Verlag Berlin Heidelberg 2006.

Anwendung der Nutzwertanalyse zur Auswahl von Frameworks zur Unterstützung testgetriebener Softwareentwicklung

Sebastian Kunze

Hochschule Darmstadt
Fachbereich Informatik
sebastian.kunze@stud.h-da.de

Art der Arbeit: Bachelorarbeit

Betreuer der Arbeit: Dr. Matthias O. Will

Abstract: Agile Prozessmodelle in Kombination mit testgetriebener Entwicklung eignen sich besonders für die Entwicklung von mobilen Applikationen. Zur Unterstützung werden dabei häufig Testing Frameworks genutzt. Die Nutzwertanalyse ermöglicht dabei a priori die Auswahl des passenden Frameworks. Um die Teilmenge der im jeweiligen Entwicklungskontext passenden Frameworks zu identifizieren, werden zunächst Auswahlkriterien entwickelt. Für die Anwendung der Nutzwertanalyse unabhängig vom aktuellen technischen Kontext werden sodann qualitative Bewertungskriterien entwickelt und zur Bewertung herangezogen. Damit kann das für den jeweiligen Entwicklungskontext am besten geeignete Framework ausgewählt werden.

1 Motivation

Bei der Entwicklung mobiler Applikationen werden häufig agile Prozessmodelle verwendet, um mit Hilfe von Prototypen auf Änderungswünsche des Kunden eingehen zu können (vgl. [AB07], [AB04]). Die testgetriebene Entwicklung kann die Stabilität der Prototypen wesentlich verbessern (vgl. [LI05, S.33ff]). Diese ist für die Bewertung des Softwareprodukts durch den Kunden im Entwicklungszyklus und dessen Zufriedenheit besonders wichtig.

Bei der Auswahl eines adäquaten Frameworks stehen dem Entwickler zahlreiche Alternativen zur Verfügung. Daher ist ein Verfahren gesucht, mit dem qualitative Bewertungskriterien unabhängig vom mobilen Betriebssystem transparent verglichen werden können, um das am besten geeignete Framework für den Entwickler zu identifizieren.

2 Die Testgetriebene Entwicklung in der Softwareentwicklung

Mit agilen Entwicklungsmodellen wird versucht, den Kunden eng in den Softwareentwicklungsprozess einzubinden, um auf Änderungswünsche besser eingehen zu können (vgl. [HA10, S.1ff]). Dabei stellt das Scrum-Modell den bekanntesten Vertreter der agilen Entwicklungsmodelle dar. Hier wird in Entwicklungszyklen von wenigen Wochen eine Auswahl der Anforderungen umgesetzt, um sie anschließend dem Kunden vorzustellen und eine gemeinsame Bewertung durchzuführen (vgl. [RO11]).

Innerhalb dieser zyklischen Entwicklung kann die testgetriebene Entwicklung eingesetzt werden. Hierbei werden die eigentlichen Modultests vor den Softwaremodulen erstellt und erst im Anschluss mit der Implementierung begonnen (vgl. [MA07, S.32ff]). Abschließend wird der Modultest und das Softwaremodul refaktoriert, indem die Struktur¹ unter Beibehaltung des implementierten Verhaltens verbessert wird (vgl. [LI05, S.11ff]).

3 Nutzwertanalyse

Die Nutzwertanalyse, die zur Analyse der Frameworks herangezogen wurde, erlaubt die Bewertung von Entscheidungsalternativen auf Basis von qualitativen Bewertungskriterien (vgl. HO08). Gegenüber anderen Verfahren ist es hier möglich, die Kriterien nach Priorität individuell zu gewichten. Für die Berechnung des Nutzwertes wird dabei die Punktzahl, die für die Zielerreichung eines Bewertungskriteriums verteilt wird, mit der Gewichtung multipliziert.

Die Nutzwertanalyse unterstützt die Entscheidungsfindung, indem sie das Framework mit dem größten Nutzen ermittelt. Dabei lässt sich der Individualisierungsprozess in vier aufeinanderfolgende Phasen gliedern (siehe Abbildung 1). Zunächst werden aus den verfügbaren Frameworks über anwendungsspezifische Auswahlkriterien die für den Entwicklungskontext passenden ausgewählt, bevor in den nächsten drei Schritten die eigentliche Nutzwertanalyse über qualitative Bewertungskriterien angewendet wird. Dies ermöglicht die transparente Auswahl des passenden Testing Frameworks.



Abbildung 1: Phasen und Tätigkeiten der Nutzwertanalyse

¹ Struktur im Sinne von Lesbarkeit, Übersichtlichkeit, Verständlichkeit, Erweiterbarkeit, Redundanzfreiheit und Testbarkeit (vgl. [RO04, S.13ff]).

3.2 Auswahlkriterien

Zunächst mussten die Entscheidungsalternativen domänenspezifische Auswahlkriterien erfüllen, um in der Nutzwertanalyse näher betrachtet zu werden. Dabei mussten diese

- das Erstellen von Modultests auf der Android-Plattform ermöglichen und
- die automatisierte Testausführung unterstützen.

Mit Hilfe dieser Auswahlkriterien konnte die Entscheidungsalternativen *Android Testing Framework*, *Robolectric*, *Robotium* und *MonkeyRunner* identifiziert werden.

3.3 Bewertungskriterien

Im zweiten Schritt wurden Bewertungskriterien für die Anwendung von Testing Frameworks zur testgetriebenen Entwicklung ermittelt und gewichtet. Das Ergebnis ist in Abbildung 2 als Zielbaum dargestellt. Die *Dauer der Testausführung* und die *Testabdeckung* erhielten dabei eine vergleichsweise hohe Priorität, da sich diese direkt auf die Produktivität und Effektivität der testgetriebenen Entwicklung auswirken. Die *Entwicklung* bewertet Aspekte im Entwicklungsprozess, Möglichkeiten der Erweiterung und den fachlich-technischen Support. Weiterhin bewertet die *Verständlichkeit* wie einfach sich Akzeptanztest in Modultests umwandeln lassen.



Abbildung 2: Zielbaum mit ermittelten Bewertungskriterien und relativer (l.) und absoluter (r.) Gewichtung

4 Ergebnis

Abbildung 3 zeigt das Ergebnis der durchgeführten Nutzwertanalyse. Dabei stellt sich *Robolectric* als Framework mit dem größten Nutzwert heraus.

Für die Bewertung der Kriterien wurde ein fünfstufiges Bewertungsschema entwickelt, welches den Zielerreichungsgrad der Bewertungskriterien abstuft und eine Punktzahl vergibt. Durch die Multiplikation von Gewichtung und Punktzahl konnten anschließend die Teilnutzwerte der Bewertungskriterien berechnet werden. Die Summe der Teilnutzwerte diente als Vergleichsgegenstand, sodass das am besten geeignete Framework für den Entwickler innerhalb der testgetriebenen Entwicklung identifiziert werden konnte.

Dabei wurden die Bewertungskriterien unabhängig von der Android-Plattform ermittelt, um die Nutzwertanalyse ebenfalls für die Analyse von Frameworks anderer Betriebssysteme einsetzen zu können.

Bewertungskriterien	Gewichtung (in Prozent)	Android Testing Framework		Subotectik		Robotium		MonkeyRunner	
		Bewertung	Nutzwert	Bewertung	Nutzwert	Bewertung	Nutzwert	Bewertung	Nutzwert
Dauer der Testausführung	30.0%	③	150	●●	270	③	150	③	150
Entwicklungsprozess	10.0%	●●	84	●●	90	●●	84	③	42
Erweiterbarkeit	6.0%	●●	54	●●	54	③	30	○○	6
Support	4.0%	●●	36	③	20	③	22	○○	6
Integrationstests	15.0%	③	75	③	75	●●	135	●	105
Isolationstests	9.0%	●	63	●	63	●	63	○○	9
Perfomanztests	6.0%	●	42	●	42	●	42	●	42
Interpretieren von Modultests	10.0%	③	50	③	50	●	70	○	30
Verfassen von Modultests	10.0%	③	50	③	50	●	70	○	30
Gesamt	100.0%	●	604	●	714	●	666	○	430

○○ = mangelhaft (1 Pkt.), ○ = ausreichend (3 Pkt.), ③ = befriedigend (5 Pkt.), ● = gut (7 Pkt.), ●● = sehr gut (9 Pkt.)

Abbildung 3: Ergebnis der Nutzwertanalyse

5 Ausblick

Mit Hilfe der Bewertungskriterien kann die Nutzwertanalyse für die Analyse des am besten geeigneten Frameworks zur Unterstützung der testgetriebenen Entwicklung herangezogen werden. Dieses unterstützt den Entwickler, indem es die Stabilität der entwickelten Softwaremodule wesentlich verbessern kann. Allerdings gewährleistet das ermittelte Framework nicht, dass die gewünschten Anforderungen des Kunden umgesetzt werden. Daher sollte zusätzlich eine gemeinsame testgetriebene Entwicklung erfolgen, in welcher die Anforderungen des Kunden als Verhaltenstests spezifizieren werden.

Literaturverzeichnis

- [AB07] Abrahamson, P.: Agile software development of mobile information systems. In Advanced information system engineering (S. 1-4). Springer Verlag 2007.
- [AB04] Abrahamson, P.: An agile approach for mobile application development. Conference on Object Oriented Programming System Language and Applications; Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages and applications (S. 174-175). Vancouver: ACM.
- [HA10] Hanser, E.: Agile Prozesse, U: Von XP über Scrum bis MAP. Springer Verlag 2010.
- [HO08] Hoffmeister, W.: Investitionsrechnung und Nutzwertanalyse. Berliner Wissenschafts-Verlag GmbH 2008.
- [LI05] Link, J.: Softwaretests mit JUnit, U: Techniken der testsgetriebenen Entwicklung. dpunkt.verlag 2005.
- [MA07] Martin, R.: Agile Principles, Pattern and Practises in C#. Pearson Education 2007.
- [RO11] Rook, A.: Agile Softwareentwicklung, U: Ein Überblick. dpunkt.verlag 2011.
- [RO04] Rook, S.: Refactoring in großen Softwareprojekten, U: Komplexe Restrukturierung erfolgreich durchführen. dpunkt.verlag 2004.
- [SO07] Sommerville, I.: Software Engineering. Pearson Studium 2007.
- [SP10] Spillner, A.: Basiswissen Softwaretest. dpunkt.verlag 2010.

Bootstrapping of component based software systems

Johannes Schauer

j.schauer@jacobs-university.de

Abstract:

With Canonicals Ubuntu, Samsungs Tizen and Jollas Sailfish OS there exists an increasing trend to use Free and Open Source (FOSS) based software distributions for emerging mobile platforms or even as unified platforms for mobile and desktop computing. As new CPU architectures and coprocessor extensions are being brought to market on a regular basis, those FOSS distributions have to be bootstrapped regularly as well. Up to now, bootstrapping such systems involved up to years of manual labor and human intuition due to the presence of millions of dependency cycles between tens of thousands of software components. This paper uses the Debian GNU/Linux distribution to demonstrate how the cyclic dependency situation of even one of the biggest component based distributions can be solved fast and efficiently for cross- and native-compilation by dependency graph analysis.

1 Introduction

Free and open source software distributions manage software components as packages [DCTZ08]. Apart from embedded distributions, most distributions make the assumption that during compile-time of any software, all packages of the distribution are available. Furthermore, they assume that source packages are natively compiled. While these assumptions do not usually create any problem throughout the native life cycle of a distribution, they do prove to become problematic when bootstrapping the distribution for a new architecture. During the bootstrapping process, a minimal build system must be cross compiled. The set of cross compiled packages should be kept as small as possible. Furthermore, when native compilation starts, only the minimal system is available on the new architecture, thereby creating build dependency cycles.

The Debian GNU/Linux distribution was not only chosen for this work because of the popularity of Debian based distributions, but also because existing research on component based software focuses on it. Furthermore, it is unique amongst other distributions as it encodes cross build dependencies using the Multi-Arch system [Lan09]. It allows to install binary packages of different architectures alongside each other or allows binary packages of one architecture to provide functionality for binary packages of other architectures. Another reason for picking Debian is its availability for a wide range of architectures where other popular distributions are only available for x86 and ARM processors. The work described in this paper is used for the ongoing arm64 port by Linaro of the Debian based distribution Ubuntu.

2 Dependency relationships

Debian organizes packages in source packages and binary packages. Source packages carry the source code of a software project together with Debian related meta data information like build dependency and build conflict relationships. Source packages are compiled to produce binary packages. Those are usually specific for each architecture. Source as well as binary packages depend on other binary packages and may also conflict with them. Source packages need binary packages to satisfy their build dependencies while binary packages need others to satisfy their runtime dependencies. Dependency relationships are formulated in conjunctive normal form [MBD⁺06]. This means that for many packages, there exist different choices of other packages that can be installed to satisfy their dependencies. Any such choice of a set of packages without any conflicts between them is called installation set and checking for its existence is an NP-complete problem [CDL⁺06, MBD⁺06].

3 The Bootstrapping process

Bootstrapping can be divided into two parts: the cross phase and the native phase. During the cross phase, a minimal set of binary packages is cross compiled. Those binary packages must be enough to allow native compilation. They must therefore include the base system and a compiler. Once the minimal build system is created, the native phase starts and all other packages are compiled natively. Build dependency cycles have to be solved during both phases.

From a dependency perspective, cross compilation is easier than native compilation. This is because many build dependency cycles are solved by dependencies being able to be satisfied by existing build architecture binary packages. During native compilation, host and build architecture are the same and therefore only (possibly not yet existing) packages of the new architecture can be used to satisfy build dependencies, therefore creating more dependency cycles.

Several tools have been developed to facilitate the bootstrapping process. The tools can be connected together using unix pipes and there exists a specific pipeline for the cross as well as for the native phase respectively. The library for dependency resolution behind those tools is `dose3`.

4 Dependency graph

For each phase, a dependency graph is built. A vertex in this dependency graph represents either a source package or an installation set of a binary package. An edge will either be a build-depends edge connecting a source package node to an installation set or a build-from edge connecting an installation set to the source packages it builds from. Because

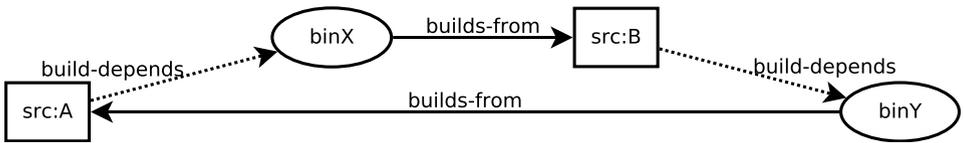


Figure 1: A simple dependency cycle of length 4

these two types of edges exist, all dependency cycles will be even numbered length. An example can be seen in figure 1.

Because source and binary packages possess multiple installation sets in the normal case, the generated dependency graph is not unique. By choosing different installation sets for source and binary packages, a dependency graph of different size can be calculated. Since binary packages within the graph can conflict with each other, the problem of calculating a dependency graph of minimal size is still unsolved.

On the other hand, a so called strong dependency graph can be generated which is unique and at the same time a lower bound of the size of the full dependency graph. Instead of generating installation sets for source and binary packages, their strong dependencies [ABDZ09] are computed. Strong dependencies of a package are the set of packages which are part of every possible installation set. This is what makes the resulting graph a lower bound.

If there were no conflicts, a minimal dependency graph can be found by using an off-the-shelf constraint satisfaction solver like `aspcud` [GKS11].

Dependency cycles can only be broken by removing build-depends edges. Since a machine cannot know without which build dependencies a source package can be built, a human has to supply the information of which edges are potentially droppable beforehand. The developed analysis tools make it easy for a human to find build dependencies that make sense to be investigated whether they can be dropped. Once enough build-depends edges have been chosen so that the graph can be made acyclic a build order is produced by removing as few build-depends edges as possible, thereby making the graph acyclic. This removal is a solution of the feedback arc set problem [LC66]. Source packages which have been decided to be built with reduced build dependencies must be scheduled for a rebuild later so that possible errors from building them without all build dependencies available are avoided.

5 Results

Based upon a selection of only essential and build-essential packages for the minimal build system, the native phase of bootstrapping Debian for arm64 creates a dependency graph consisting of 39468 vertices. It contains one big (926 vertices) strongly connected component encompassing all core packages and eight other small (2-7 vertices) SCC which

belong to languages that need themselves to be built.

The analysis focused on the main strongly connected component of 926 vertices and 7354 edges because once it is solved, it would mean that 99% of packages in Debian can be natively compiled from there.

The performance of the feedback arc set algorithm can be adapted for better runtime. Within 3 minutes, a feedback arc set of 97 edges can be calculated. With 7 minutes, a feedback arc set of 90 edges is found and given 8 hours, the number of edges that is removed drops to 87.

6 Acknowledgments

The author wants to especially thank Pietro Abate of the Mancoosi team at University Paris Diderot for his continuous help and support, Wookey from linaro.org and the Debian community.

References

- [ABDZ09] Pietro Abate, Jaap Boender, Roberto Di Cosmo, and Stefano Zacchiroli. Strong Dependencies between Software Components. In *ESEM 2009: International Symposium on Empirical Software Engineering and Measurement*, pages 89–99. IEEE, 2009.
- [CDL⁺06] Roberto Di Cosmo, Berke Durak, Xavier Leroy, Fabio Mancinelli, Jérôme Vouillon, et al. Maintaining large software distributions: new challenges from the FOSS era. In *Proceedings of the FRCSS 2006 workshop.*, pages 7–20, 2006.
- [DCTZ08] Roberto Di Cosmo, Paulo Trenzentes, and Stefano Zacchiroli. Package Upgrades in FOSS Distributions: Details and Challenges. In *HotSWUp’08: International Workshop on Hot Topics in Software Upgrades*, pages 7:1–7:5. ACM, 2008.
- [GKS11] Martin Gebser, Roland Kaminski, and Torsten Schaub. aspcud: A linux package configuration tool based on answer set programming. *arXiv preprint arXiv:1109.0113*, 2011.
- [Lan09] Steve Langasek. Multiarch Specification. <https://wiki.ubuntu.com/MultiarchSpec>, 2009.
- [LC66] Abraham Lempel and Israel Cederbaum. Minimum feedback arc and vertex sets of a directed graph. *Circuit Theory, IEEE Transactions on*, 13(4):399–403, 1966.
- [MBD⁺06] Fabio Mancinelli, Jaap Boender, Roberto Di Cosmo, Jérôme Vouillon, Berke Durak, Xavier Leroy, and Ralf Treinen. Managing the Complexity of Large Free and Open Source Package-Based Software Distributions. In *ASE 2006: Automated Software Engineering*, pages 199–208. IEEE, 2006.

Aufbau und Auswertung wiederverwendbarer Strukturen aus Ticketsystem- und Versionskontrollsystem-Daten für multiple Analysen

Jochen Wierum

Hochschule Bonn-Rhein-Sieg
Fachbereich Informatik

jochen.wierum.2007w@informatik.fh-bonn-rhein-sieg.de

Art der Arbeit: Master-Thesis, eingereicht am 31.10.2012

Betreuerin der Arbeit: Prof. Dr. Simone Bürsner (simone.buersner@h-brs.de)

Abstract: Dieses Papier zeigt eine Lösung, beliebige Zusammenhänge zwischen Tickets und Commits zu analysieren. Es wird eine auf einem Graphen basierende Datenstruktur vorgestellt, die Inhalte aus Versionskontrollsystemen und Ticketsystemen speichert. Der Graph ist optimiert für Durchsuchbarkeit und stellt die Zusammenhänge zwischen den einzelnen Artefakten beider Systeme in den Vordergrund. Dies ermöglicht es, verschiedene Algorithmen zu definieren, welche Muster im Graphen suchen und die Ergebnisse grafisch aufbereiten. Der Ansatz wurde praktisch an dem Open Source Projekt Redmine evaluiert.

1 Motivation zum Mining von Repositories

Im Laufe ihres Lebens erfährt Software häufig Änderungen. Repositories helfen dabei, das Verwalten dieser Änderungen zu vereinfachen und zu verwalten. Ticketsysteme (Issue Tracking Systems, ITS) werden genutzt, um neue Anforderungen und Bugs zu verwalten, während Versionskontrollsysteme (Version Control Systems, VCS) Änderungen am Quelltext nachhalten [Spi05]. Die Inhalte von ITS und VCS können auch nach der eigentlichen Umsetzung von Änderungen noch verwendet werden, um das Projekt zu analysieren.

Zahlreiche Untersuchungen konzentrieren sich auf einzelne Aspekte eines Projekts und analysieren dazu isoliert ITS oder VCS-Daten. Um die Aussagen kombinieren zu können ist es relevant, die Daten einheitlich zu repräsentieren. Bestehende Ansätze, die mehrere Repositories verbinden, konzentrieren sich meist wenig auf die konkrete Modellierung der Verlinkung verschiedener Artefakte untereinander [ADG04, BSM10, OYS⁺04].

Die diesem Papier zugrundeliegende Master-Thesis liefert einen Ansatz, die Daten aus ITS und VCS als einen zusammenhängenden Graphen zu repräsentieren. Der Ansatz überführt die Daten aller Repositories in eine gemeinsame Datenstruktur, um auch übergreifende Zusammenhänge sichtbar und durchsuchbar zu machen. Auf dieser, von den Quellsystemen

unabhängigen Datenstruktur, werden dann exemplarisch praxisrelevante Algorithmen zur Anreicherung und Auswertung der Struktur definiert. Zuletzt wird der Ansatz praktisch in einem Framework implementiert und beide werden kritisch bewertet.

2 Repräsentation der Daten

Die Inhalte von ITS und VCS werden als ein Graph modelliert. Graphen lassen sich sehr effizient auf komplexe Muster durchsuchen und sind gut geeignet, um die Beziehungen zwischen einzelnen Artefakten und ihren Bestandteilen abzubilden. Der Graph wird so definiert, dass eine logische Strukturierung aller importierten Daten entsteht [AG08]. Daten werden dedupliziert abgelegt, sodass verwandte Knoten über gemeinsam referenzierte Knoten schnell identifiziert werden können. Dies ermöglicht eine gute Durchsuchbarkeit.

Knoten und Kanten werden mit Attributen und die Kanten zusätzlich mit einem eindeutigen Typen versehen [RDW12]. Ausgehend von einem Einstiegsknoten werden zunächst Repository-Knoten definiert. Diese verweisen auf ITS- und VCS-Repositories, Commits, Tickets, Personen, Tags, Keywords, etc. Über sie können die jeweiligen Instanzen schnell erreicht werden.

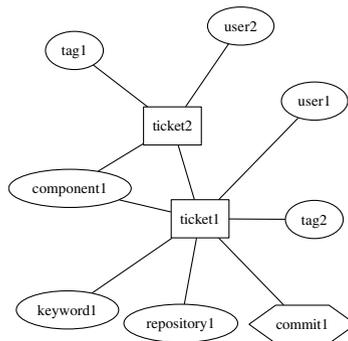


Abbildung 1: Ausschnitt eines Graphen

Pro Ticketversion wird ein neuer Ticket-Knoten angelegt. Eine neue Ticketversion entsteht, sobald ein Ticket verändert wird. Ebenso wird pro importiertem Commit ein Knoten angelegt. Attribute, welche an mehreren Stellen verwendet werden (wie Personen, Tags, Komponenten, usw.) werden dabei durch gesonderte, gemeinsame Knoten repräsentiert. Einmalige Werte wie Texte oder Timestamps, werden nicht ausgelagert. Der resultierende Graph ist zusammenhängend und Gemeinsamkeiten zwischen Tickets und Commits sind in der Struktur des Graphen abgebildet. Abbildung 1 zeigt beispielhaft einen Ausschnitt aus einem möglichen Graphen.

In einem weiteren Schritt wird der Graph mit weiteren Knoten und Kanten angereichert, die indirekt aus dem Inhalt der Tickets und Commits gewonnen werden können. Dabei werden Schlüsselwörter und Verweise auf andere Tickets und Commits aus den importier-

ten Texten extrahiert und ebenfalls im Graph abgelegt.

3 Algorithmen und deren Evaluation

Die Suchalgorithmen definieren Muster, welche ausgehend von einem gegebenen Knoten im Graphen gesucht werden. Die einzelnen Funde werden anschließend zusammengefasst und visuell aufbereitet. Zum Durchsuchen der Struktur werden exemplarische Algorithmen wie in Tabelle 1 definiert. Für die gewählten Algorithmen werden verschiedene Aufbereitungen wie bei [LDGG10, BINF12, SL09, TLC⁺09] benötigt.

Projektrollen	Ticketworkflow	Wissensverteilung
Ermittelt alle Rollen der Projektbeteiligten durch Analyse der Status, mit denen ihnen Tickets zugewiesen werden. Generiert wird eine Tabelle.	Ermittelt den typischen Workflow von Tickets, indem analysiert wird, welche Statusübergänge in den Tickets vorkommen. Generiert wird ein Graph.	Ermittelt Schlüsselwörter im Text der Tickets, die oft, aber nur von wenigen Personen verwendet werden. Generiert wird eine Tabelle.

Tabelle 1: Übersicht über exemplarische Implementierungen von Algorithmen und Suchmustern

Aufbauend auf einer Graphdatenbank wird in der Arbeit unter Verwendung von funktionalen und objektorientierten Ansätzen ein Framework entwickelt, das den theoretischen Ansatz implementiert. Mit Hilfe des Frameworks wird das Ruby-on-Rails-Projekt Redmine (<http://www.redmine.org/>, Stand Oktober 2012) untersucht. Eine Anwendung der in Tabelle 1 angeführten Algorithmen ergibt die in Tabelle 2 erläuterten Ergebnisse.

Projektrollen	Ticketworkflow	Wissensverteilung
Das Projekt verfügt über keine dedizierten Tester. Auch Reviews finden nicht statt. Die meisten Personen beteiligen sich als Reporter für Bugs, nur einige wenige Beteiligte sind Entwickler.	Es konnte identifiziert werden, dass das Projekt keinen fest definierten Arbeitsablauf für Tickets besitzt. Es konnte keine wiederkehrende Struktur im Statusverlauf der Tickets erkannt werden. Der größte Anteil Tickets wird direkt geschlossen, und nicht erkennbar verarbeitet.	Es wurden alte Versionen von Redmine, Rails und Ruby identifiziert, die für die aktuelle Entwicklung nicht mehr relevant sind. Außerdem wurden Sprachen in den Übersetzungen erkannt. Ggf. können Änderungen in der Oberfläche nicht in diese Sprachen übertragen werden.

Tabelle 2: Übersicht der Ergebnisse der in Tabelle 1 definierten Algorithmen

Weitere Algorithmen und deren Evaluation werden in der Master-Thesis betrachtet.

4 Zusammenfassung und Ausblick

Die Arbeit zeigt, wie Daten aus unterschiedlichen Repositories so kombiniert werden können, dass artefaktübergreifende Fragestellungen mithilfe einer einheitlichen Datenrepräsentation beantwortet werden können. Das ermöglicht es, auch komplexere Zusammenhänge bei der Analyse von Projekten zu beantworten.

Die Genauigkeit der Ergebnisse ließen sich durch eine feingranularere Speicherung der importierten Daten weiter verbessern. So könnten etwa Suchen definiert werden, die ausschließlich Bugs durchsuchen, anstatt sich auf alle Tickets des ITS zu beziehen. Auch eine Erweiterung der Datenstruktur um neue Knoten ist eine Möglichkeit, noch vielfältigere Analysen definieren zu können. Hier wären Ticketanhänge oder comittete Dateien denkbare Kandidaten, die etwa auch die Bewertung von Code-Qualität ermöglichen würden.

Literatur

- [ADG04] Omar Alonso, P Devanbu und Michael Gertz. Database techniques for the analysis and exploration of software repositories. In *MSR*, Jgg. 4, Seiten 37–41, 2004.
- [AG08] R. Angles und C. Gutierrez. Survey of graph database models. *Computing Surveys*, 40(1):1, 2008.
- [BINF12] P. Bhattacharya, M. Iliofotou, I. Neamtiu und M. Faloutsos. Graph-based analysis and prediction for software evolution. In *Proceedings of the 2012 International Conference on Software Engineering*, Seiten 419–429. IEEE Press, 2012.
- [BSM10] Stefan Biffl, Wikan Danar Sunindyo und Thomas Moser. Semantic integration of heterogeneous data sources for monitoring frequent-release software projects. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on*, Seiten 360–367. IEEE, 2010.
- [LDGG10] A. Lamkanfi, S. Demeyer, E. Giger und B. Goethals. Predicting the severity of a reported bug. In *Mining Software Repositories (MSR), 2010 7th IEEE Working Conference on*, Seiten 1–10. Ieee, 2010.
- [OYS⁺04] Masao Ohira, Reishi Yokomori, Makoto Sakai, Ken-ichi Matsumoto, Katsuro Inoue und Koji Torii. Empirical project monitor: A tool for mining multiple project data. In *International Workshop on Mining Software Repositories (MSR2004)*, Seiten 42–46. Citeseer, 2004.
- [RDW12] Marko A. Rodriguez und Pierre De Wilde. Gremlin, 6 2012.
- [SL09] Holger Schackmann und Horst Lichter. Evaluating process quality in GNOME based on change request data. In *Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories, MSR '09*, Seiten 95–98, Washington, DC, USA, 2009. IEEE Computer Society.
- [Spi05] D. Spinellis. Version control systems. *Software, IEEE*, 22(5):108–109, 2005.
- [TLC⁺09] H.M. Tran, C. Lange, G. Chulkov, J. Schönwälder und M. Kohlhase. Applying Semantic Techniques to Search and Analyze Bug Tracking Data. *Journal of Network and Systems Management*, 17(3):285–308, 2009.

Dokumentation von Software Architekturen

Andrew Page, Markus Riegger, Mehtap Yilmaz

University of Applied Science Reutlingen
Master - Wirtschaftsinformatik
Alteburgstraße 150
72762 Reutlingen
Andrew.Page@ Student.Reutlingen-University.de
Markus.Riegger@ Student.Reutlingen-University.de
Mehtap.Yilmaz@ Student.Reutlingen-University.de

Art der Arbeit: Projektarbeit

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

Abstract: Diese wissenschaftliche Arbeit thematisiert die Vorgehensweise um Software Architekturen präzise dokumentieren zu können. Dabei wird der Hauptfokus auf die relevanten Architekturkonzepte Views, die Dokumentation von Informationen mit dessen Beziehungen zu den einzelnen Elementen und daraus resultierend die Entwicklung eines Pattern Katalogs gelegt.

1 Einführung

Software Architekturen haben sich als fundamentale Konzepte etabliert, um große, komplexe Software Systeme erfolgreich modellieren zu können. Da Architekturen langlebige Konstrukte darstellen, die sich über die Zeit den Anforderungen des Marktes und den neuen Technologien stellen müssen, sind diese ständigen Veränderungen ausgesetzt. Daher ist die Kommunikation und die Dokumentation von Architekturen so wichtig, wie das Modellieren von Architekturen. Sollte es nämlich keine Möglichkeit geben eine Architektur zu verstehen oder diese analysieren zu können so ist die Erweiterbarkeit, die Wartung und das Verständnis über ein System nicht gegeben und daher die Modellierung auch bedeutungslos. In den folgenden Kapiteln werden Elemente und die Vorgehensweisen aufgeführt, die es einem Architekten ermöglichen, Architekturen detailliert zu erklären und dokumentieren zu können[CBB10].

2 Views

Eines der wichtigsten Konzepte in Bezug zur Dokumentation von Software Architekturen ist eine View. Software Architekturen sind sehr komplex und lassen sich nicht durch eine eindimensionale Sicht erklären. Denn Komponenten haben

Subkomponenten und interne als auch externe Beziehungen. Eine View betrachtet eine Menge von Systemelementen und deren Beziehungen zueinander. Diese Views vertreten dabei auch die verschiedensten Systemstrukturen die gleichzeitig in Software Systemen vorkommen. Views werden verwendet um einen Teilaspekt eines Systems darzustellen. Dadurch werden komplexe Architekturen übersichtlicher und verständlicher. Aus vorgegebenen Zielen lassen sich die relevanten Views der Architektur entwickeln, diese dienen letztendlich zur Spezifikation von Richtlinien für Entwickler oder als Vorlage der Analyse des Systems oder sogar zur Generierung von automatischen erzeugtem Quellcode. Wichtig ist, dass verschiedene Views auch verschiedenen Qualitätseigenschaften beinhalten. Als Beispiel betrachten wir die Systemschichten als View so erhalten wir eine Aussage über die Software Portabilität. Wenn wir die Einsatzsicht eines Systems betrachten, erhalten wir eine Aussage über die Leistung und die Zuverlässigkeit des Systems. Die Stärke der Views liegt in der Betonung auf einen bestimmten Aspekt des Systems, während dessen andere Aspekte außen vor gelassen werden, damit ein entstandenes Problem oder Herausforderung bewältigt werden kann.

Die Dokumentation der Views beinhaltet eine graphische Darstellung der primären Elemente und deren Beziehungen in einer Sicht. Dabei werden die Elemente kategorisiert. Bei einer Kategorisierung werden die in einer Sicht darzustellenden Elemente erläutert und die Elementeigenschaften definiert. Eine Spezifikation der Elementschnittstellen und deren Verhalten. Eine Anleitung über die Flexibilität des Systems, in der integrierte Mechanismen vorkommen, die eine Architektur spezifizieren. Beschränkungen, Grundprinzipien und Designerklärung der Architektur. Eine Einführung in die Paketstruktur mit einem Handbuch, damit eine Interessengruppe zügig auf ein gewünschtes Element oder Information zugreifen kann. Informationen über den Zusammenhang der verwandten Views und über den Zusammenhang zur Gesamtarchitektur. Managementinformationen damit die gesamte Paketstruktur effektiv genutzt und gewartet werden kann [CBB10].

3 Ausschnitt Pattern Katalog

Im Rahmen dieses Projektes wurde ein Pattern Katalog für die Dokumentation von Software Architekturen entwickelt. In den folgenden Unterabschnitten werden einige ausgewählte Patterns aus diesem vorgestellt.

- Das „Generalization Style Pattern“ bezieht sich auf „Module View“.
- Das „Top-Down-Refinement-Pattern“ bezieht sich auf „Component & Connector View“.

3.1 Generalization Style Pattern

Problem:

Wie kann eine Spezialisierungsbeziehung zwischen Modulen angezeigt?

Lösung:

- Generalization Style wird durch eine *ist-ein* Beziehung dargestellt
- Modules in diesem Style fangen Gemeinsamkeiten und Unterschiede ein
- Erweiterungen können durch Hinzufügen, Entfernen oder Ändern von Kindklassen gemacht werden; eine Änderung in der Elternklasse führt automatisch zu einer Veränderung in allen Kindklassen, da diese von ihm erben
- Generalization ermöglicht sowohl eine Schnittstellenvererbung als auch eine Implementierungsvererbung

Erklärung:

Dieses Style Pattern ist nützlich

- um Vererbung in objektorientierten Designs auszudrücken
- für die Erfassung von Gemeinsamkeiten mit Variationen als Kinder
- für die Unterstützung der Wiederverwendung

3.2 Top-Down-Refinement-Pattern

Problem:

Wie können komplexe Architekturen modelliert werden, damit diese transparent für bestimmte Interessengruppen bleiben, aber dennoch detailliert sind?

Lösung:

Drill-Down Funktionen fügen Informationen zu vorhandenen views hinzu, durch Offenlegung der Elemente und ihrer Beziehungen. Rollup hat die Funktionalität Informationen zu verbergen, die in bestimmten Designabschnitten nicht mehr für bestimmte Interessengruppen notwendig sind.

Erklärung:

Diese Herangehensweise ermöglicht es, Informationen zu dokumentieren, da das Design über die Zeit reift. Decomposition Refinements halten die views Konsistenz und konzentrieren sich auf ein einzelnes Element:

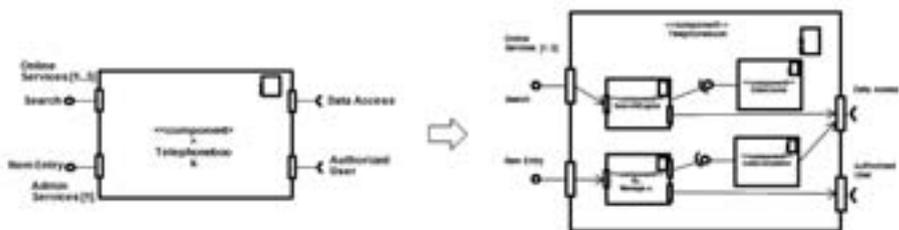


Abbildung 2: Top-Down-Refinement-Pattern

4 Dokumentation einer View

Die Einheit für die Dokumentation von Sichten (Views) ist das Sichtenpaket (view packet). Das Sichtenpaket ist die kleinste Informationseinheit, welche einem Stakeholder

jemals übergeben wird. Sichtenpakete erlauben es Sichten in handhabbare Stücke zu teilen, weil eine Darstellung als Gesamtdokumentation zu komplex wäre. Ein solches Packet kann Informationen über einen kleinen Bereich eines Systems oder über einen Detaillierungslevel modellieren. Untergeordnete Sichtenpakete können Elemente detaillierter darstellen. Folgendes Template kann zur Dokumentation von Sichten oder Sichtenpaketen verwendet werden.

“Primary presentation” veranschaulicht die Elemente und deren Beziehungen. Enthalten sind die Informationen des Systems, welche übermittelt werden sollen. “Element catalog” enthält wichtige Details der Elemente eines Diagramms. “Context diagram” zeigt die Abgrenzung zwischen Informationen eines Sichtenpakets und welchen Informationen ausserhalb des “Scopes” eines Paketes. “Variability guide” zeigt das Anwenden aller Variationen, welche Teil einer Architektur eines Sichtenpakets ist. “Architecture background” erklärt warum das verwendete Design ausgewählt wurde. In “Other information” sind die Informationen abhängig von den Standards des jeweiligen Unternehmens. Informationen an dieser Stelle sind nicht architekturell. “Related view packets” enthält, falls vorhanden, einen Zeiger auf verwandte Pakete [CBB10].

5 Viewübergreifende Dokumentation von Informationen, welche sich auf mehrere Views beziehen

Die Dokumentation von Informationen (beyond-view), welche mehr als ein View und das ganze Packet betreffen. Diese Art der Dokumentation verbindet die Sichten. Die drei wichtigsten Aspekte sind Wie, Was und Warum [CBB10].

1. How the documentation is organized to serve a stakeholder
 - Documentation roadmap
 - Description of the parts
 - How stakeholders might use the package
 - View template
2. What the architecture is
 - System overview
 - Mapping between views
 - Directory
 - Project glossary and acronym list
3. Why the architecture is the way it is
 - Background, design constraints, and rationale

How the documentation is organized to serve a stakeholder erklärt wie eine Dokumentation aufgebaut ist (documentation roadmap). Es erlaubt einem Stakeholder einen schnellen Zugriff auf die für den Stakeholder relevanten Informationen.

What the architecture is besteht aus einem Systemüberblick, dem Mapping zwischen Views, dem Directory (eine Art Inhaltsverzeichnis über Indexe), dem Projektglossar und der Acronymliste.

Abbildung 1 Template viewübergreifende Dokumentation

Why the architecture is the way it is beinhaltet z. B. externe Abhängigkeiten.

Literaturverzeichnis

[CBB10] Clements, P.; Bachmann, F.; Bass, L.: Documenting Software Architectures: Views and Beyond. Addison-Wesley Longman, Amsterdam, 2010.

Konzeption und Implementierung eines Risiko- und Chancen-Management-Moduls für die Project + Business Software *admileo*

Dennis Ziegenhagen

Hochschule Osnabrück
Fakultät Ingenieurwissenschaften und Informatik
dennis.ziegenhagen@hs-osnabrueck.de

Art der Arbeit: Bachelorarbeit
Betreuer der Arbeit: Prof. Dr. Frank M. Thiesing, Hochschule Osnabrück (GI-Vertrauensdozent)
Dipl.-Inform. Ralf Brand, Archimedes Software und Consulting GmbH & Co. KG

Abstract: Es wird ein Konzept erstellt, welches allgemeine Methoden, Praktiken und Hilfsmittel des Risikomanagements (RM) in Form von Funktionalitäten eines Software-Moduls umsetzt. Ziel ist es, durch zusammenhängende Teilfunktionen einen umfassenden Risikomanagement-Prozess abzubilden. Die Teilfunktionen sind auch separat nutzbar, um eine Unterstützung für bereits existierende RM-Prozesse zu bieten oder diese zu ergänzen. Das entstandene Konzept wird als neues Modul für die Project + Business-Software *admileo* umgesetzt und in diese integriert.

1 Problem und Motivation

Nicht erkannte oder nicht behandelte Risiken können Projekte verzögern, ihre Qualität mindern oder gar komplett scheitern lassen [Wan07; Sei11]. Die Liste entsprechender Beispiele wird stetig ergänzt. Aktuell sei das Projekt „Flughafen Berlin Brandenburg“ genannt, dessen Verzögerung unter anderem auf mangelndes Risikomanagement zurückgeführt wird [Neu12]. Auch der Zusammenbruch ganzer Unternehmen kann in Teilen auf nicht erkannte, ignorierte oder falsch behandelte Risiken zurückgeführt werden [Gle11].

Zur Lösung dieses Problems ist es mit der Integration von Risikomanagement-Praktiken in die Projekt- oder Unternehmensstruktur nicht getan, da der Erfolg des Risikomanagements trotzdem ausbleiben kann [Wan07]. Effektives Management von Risiken ist ein systematischer und kontinuierlicher Prozess, der in allen Projektphasen aktiv betrieben werden muss [Har04].

Im Rahmen der Bachelorarbeit wurden allgemeine Risikomanagement-Praktiken recherchiert. Die hier gewonnenen Ergebnisse werden zu einem allgemeinen Konzept erweitert, welches gängige Risikomanagement-Methoden aus der Praxis in Form von Software-Funktionalitäten umsetzt. Dabei bilden der praktische Nutzen und die zielgerichtete Unterstützung im Risikomanagement den Schwerpunkt des Konzepts.

Des Weiteren beinhaltet die Bachelorarbeit die Planung und die Implementierung eines Software-Moduls zur konkreten Umsetzung des Konzeptes. Das Modul trägt den Namen „Risiko- und Chancen-Manager“ und wird in die Software *admileo* der Firma Archimедon¹ integriert. Dabei sind Schnittstellen und Interaktionen mit existierenden Modulen ebenso zu berücksichtigen, wie die reibungslose Anknüpfung an die Benutzerführung. Für die Implementierung wird die Programmiersprache Java in der Eclipse-Entwicklungsumgebung verwendet. Zur Datenhaltung wird eine PostgreSQL-Datenbank in Verbindung mit SQL2JAVA eingesetzt.

2 Umsetzung des Risikomanagement-Prozesses

Unabhängig vom speziellen Bereich, in dem Risikomanagement angewendet wird, lässt sich der (idealisierte) RM-Prozess allgemein und übergreifend beschreiben. Dies geht auch aus der Norm ISO 31000 hervor, welche den Prozess vereinheitlicht und dabei einige hilfreiche Grundsätze zur Verfügung stellt [Brü08]. Die einzelnen Schritte, aus denen sich der Prozess zusammensetzt, finden sich in dieser und ähnlicher Form häufig in Literatur und RM-Theorien wieder [Har04; Möl12; Wan07]. Sie dienen für das Konzept als Leitfaden und Grundgerüst, um sicher zu stellen, dass der Prozess auch tatsächlich vollständig abgebildet wird. Dabei werden die einzelnen Schritte mit Hilfe der recherchierten Praxismethoden umgesetzt und erweitert:

- Best Practices
- Vermeidung häufig gemachter Fehler
- Einbindung von Hilfsmitteln und -methoden

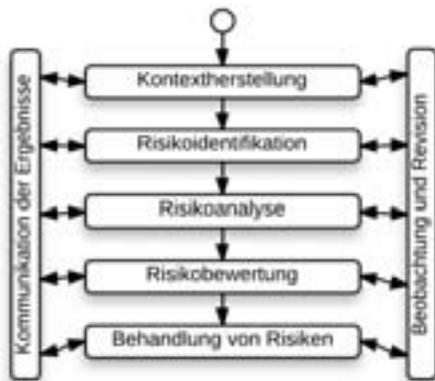


Abbildung 1 Risikomanagement-Prozess nach ISO 31000

¹ siehe www.archimедon.de

Da ein wesentlicher Schwerpunkt von admileo im Projektmanagement liegt, wurde besonders das Projekt-Risikomanagement berücksichtigt. Die einzelnen Schritte des RM-Prozesses wurden deshalb den existierenden Workflows des Projektmanagements angepasst.

Für jeden Schritt des Risikomanagement-Prozesses wird die Umsetzung im Folgenden knapp zusammengefasst:

Kontextherstellung

Ein Teil des Risikokontextes wird durch die Verknüpfung zum Projektmanagement hergestellt:

- Das Projekt, für welches das Risiko angelegt wird
- Die gewählte Risiko-Kategorie
- Ggf. Verbindung zur Herkunft aus dem Risiko-Katalog (s.u.)

Risikoidentifikation

Als Hilfsmittel zur Risikoidentifikation dient eine Erweiterung der häufig verwendeten Risiko-Checklisten durch die interne Risiko-Datenbank: Es entsteht ein projektübergreifender, unternehmensweiter Risiko-Katalog, der neben Standard-Risiken auch in der Lage ist, *Lessons Learned*² abzubilden.

Risikoanalyse

Aus den Feldern zur Erfassung der Risikodaten wird ein Formular zur Analyse des Risikos gebildet. Die Verwendung von Pflichtfeldern gewährleistet dabei die Einhaltung eines Mindestniveaus bei der Analyse.

Risikobewertung

Zur Bewertung muss eine (geschätzte) Eintrittswahrscheinlichkeit angegeben werden. Zur Auswirkung des Risikos kann der Anwender unterschiedliche Angaben machen: Kosten, Dauer und Qualität.

Behandlung von Risiken

Für die Behandlung von Risiken, die nicht toleriert werden, lassen sich Maßnahmen anlegen. Der Benutzer kann definieren, welcher Art die Beeinflussung des Risikos durch die Maßnahme sein soll:

- Veränderung der Eintrittswahrscheinlichkeit
- Veränderung der Auswirkung(en)
- Transferierung des Risikos („abwälzen“)

² Sammeln von Erfahrungen, die für spätere Projekte relevant sein können

Beobachtung und Revision

Neben einer zentralen, hierarchischen Risiko- und Maßnahmenübersicht werden zusätzliche Risiko-Zusammenfassungen im jeweiligen Projekt angezeigt. Risiken können mit Hilfe der Monte-Carlo-Simulation aggregiert werden (Zusammenfassen der Einzelrisiken durch ein numerisches Analyse- bzw. Simulationsverfahren auf Basis von Zufallswerten. Ergebnis ist eine Wahrscheinlichkeitsverteilung für die zu erwartenden Auswirkungen. [Gle01]).

Für die Auswirkung der Risiken werden Bruttodaten (ohne Einberechnung der Maßnahmen) und Nettodaten (inklusive Maßnahmen-Einberechnung) angezeigt.

Kommunikation der Ergebnisse

Risiken sind prinzipiell für alle Projektbeteiligten sichtbar und können von allen Projektbeteiligten angelegt werden. Über ein internes Nachrichtensystem werden relevante Ereignisse (z.B. „neues Risiko angelegt“) automatisch kommuniziert. Es besteht die Möglichkeit, manuell angelegte Risiken in den globalen Katalog aufzunehmen.

3 Ausblick

Die Entwicklung des Moduls endet nicht mit dem Abschluss der Bachelorarbeit. Das entwickelte Konzept legt den Schwerpunkt auf das Management von Projektrisiken. Erweiterungen bezüglich des Managements von Unternehmensrisiken sind denkbar. Für letztere könnte gerade aus finanzwirtschaftlicher Sicht eine Einbeziehung von Kennzahlen wie z.B. *Value at Risk* [Bro12] eine wertvolle Ergänzung darstellen.

Literaturverzeichnis

- [Bro12] Broll, Udo und Wahl, Jack E. 2012. *Risikomanagement im Unternehmen*. Wiesbaden : Springer Gabler, 2012.
- [Brü08] Brühwiler, Bruno. 2008. Neue Standards im Risikomanagement. *Management und Qualität*. 2008, 5.
- [Gle01] Gleißner, Werner. 2001. Auf nach Monte Carlo - Simulationsverfahren zur Risiko-Aggregation. *RISKNEWS*. 2001, 1.
- [Gle11] Gleißner, Werner. 2011. Die größte anzunehmende Dummheit im Risikomanagement. *Risk, Compliance & Audit*. 2011, 1.
- [Har04] Harrant, Horst und Hemmrich, Angela. 2004. *Risikomanagement in Projekten*. München : Carl Hanser Verlag, 2004.
- [Möll12] Möller, Thor, et al. 2012. *Projekte erfolgreich managen*. Köln : TÜV Media, 2012.
- [Neu12] Neuhaus, Carla. 2012. Fragen der Größenordnung. *Der Tagesspiegel*. [Online] Verlag Der Tagesspiegel GmbH, 13. Mai 2012. [Zitat vom: 04. Januar 2013.] <http://www.tagesspiegel.de>.
- [Sei11] Seidl, Jörg. 2011. *Multiprojektmanagement*. Heidelberg : Springer-Verlag, 2011.
- [Wan07] Wanner, Roland. 2007. *Projekt-Risikomanagement*. Norderstedt : Books on Demand GmbH, 2007.

eTaddy - Ein integratives Framework für die Erstellung, Durchführung und Analyse von Eyetracking-Daten

Tanja Blascheck, Michael Raschke, Thomas Ertl

Abstract: Eyetracking-Experimente werden vermehrt dazu eingesetzt, das Blickverhalten von Probanden während der Durchführung verschiedener Aufgaben zu untersuchen. Ein Großteil des Arbeitsaufwandes bei der Auswertung eines Eyetracking-Experiments wird für manuelle Aufgaben wie die Eingabe von Fragebogenantworten, das Sortieren der aufgezeichneten Daten sowie in das Durchführen von statistischen Tests und die Erstellung von Visualisierungen aufgewendet. Dieser Artikel präsentiert ein integratives Framework, das den Versuchsleiter bei der Erstellung, Durchführung und Auswertung eines Eyetracking-Experiments unterstützt.

1 Einführung

Eyetracking ist in den letzten Jahren zu einer Standardtechnik für Benutzerexperimente geworden. In einem Eyetracking-Experiment wird die Augenbewegung von Probanden beim Lösen einer Aufgabe aufgenommen und anschließend analysiert. Bei der Analyse der Eyetracking-Daten werden verschiedene Visualisierungstechniken und statistische Berechnungen eingesetzt. Da in einem Benutzerexperiment meist eine größere Anzahl von Probanden untersucht und verschiedene Stimuli eingesetzt werden, muss eine große Menge an Daten miteinander verglichen und ausgewertet werden. Aus diesem Grund ist die Auswertung von Eyetracking-Experimenten sehr aufwändig.

Um die Durchführung der Auswertung eines Eyetracking-Experiments zu vereinfachen, haben wir ein integratives Framework entwickelt, das den Versuchsleiter bei der Erstellung, Durchführung und Analyse eines Eyetracking-Experiments unterstützen soll. Das Framework besteht aus drei Komponenten, welche die drei Phasen der Studiererstellung, -durchführung und -analyse abbilden. Alle Daten der Studie werden in einer Datenbank gespeichert und stehen zu jedem Zeitpunkt zur Verfügung, ohne weitere Konvertierungen durchführen zu müssen. Außerdem können über ein Plug-In-System Statistiksoftwarepakete und selbst entwickelte Visualisierungen oder statistische Auswertalgorithmen eingebunden werden.

2 Related Work

Eine gute Einführung in das Thema Eyetracking wird von Nielsen und Pernice gegeben [2]. Goldberg und Kotval zeigen anhand eines illustrativen Eyetracking-Experiments deren wichtigste Aspekte [5]. In den meisten Eyetracking-Experimenten werden die Daten mit

Visualisierungen in Form von Heatmaps und Scanpaths ausgewertet. Eine gute Übersicht über Heatmaps wird durch Duchowski et al. gegeben [7]. Weitere Analysetechniken auf Basis der Verwendung von Areas of Interest (AOIs) sind beispielsweise Transitionsmatrizen [1], [3] oder die Parallel Scan-Path Visualisierungen [8].

Neue Eyetracking-Visualisierungen für die Analyse werden meist als eine unabhängige Software entwickelt. Beispiele sind eSeeTrack [6] oder eyePatterns [9]. Kommerzielle Eyetracking-Software wie Tobii Studio oder BeeGaze von SMI bieten es dem Benutzer nicht an, eigene Visualisierungen zu integrieren. Diese Arbeit folgt im Gegensatz zu existierenden Lösungen einem integrativen Konzept, das alle einzelnen Analysekomponenten in einer Software vereint. Das Konzept wurde in Form eines Prototypen implementiert und wird im folgenden Abschnitt näher beschrieben.

3 eTaddy - Eyetracking Analysis, Conduction, and Designtool for User Studys

Bei der Studiererstellung können neben der Erstellung des Studiendesigns auch Fragebögen erstellt werden, die im späteren Benutzerexperiment dem Probanden zusammen mit Anleitungen oder Sehtests in der gewünschten Reihenfolge in einem separaten Programmfenster präsentiert werden. Der Proband kann dadurch den Fragebogen digital ausfüllen. Die Software hat für diesen Schritt des Experiments einen Split-Screen implementiert, so dass der Versuchsleiter den Ablauf der Studie in einem zweiten Fenster kontrollieren kann. Er kann in diesem zweiten Fenster steuern, welcher Fragebogen oder welche Informationen dem Probanden angezeigt werden und erhält Informationen über den Startzeitpunkt der Studie.

Nachdem das Benutzerexperiment durchgeführt wurde, erfolgt die Analyse in vier Schritten (vgl. Abbildung 1): Zuerst werden die erhobenen und aufgenommenen Daten des Eyetracking-Experiments in die Datenbank geladen (1). Anschließend stehen dem Versuchsleiter alle Stimuli sowie Antworten der Fragebögen, Durchführungszeiten, Genauigkeitswerte und Eyetracking-Metriken aller Probanden zur Verfügung. Der Benutzer kann im nächsten Schritt mehrere Stimuli und Probanden auswählen, welche für die Analyse von Interesse sind (2). Aus einer Liste können Metriken, Statistiken und Visualisierungen ausgewählt werden (3). Die Ergebnisse der statistischen Berechnungen oder die Visualisierung der Eyetracking-Daten werden in der Mitte des Programmfensters angezeigt (4). Durch die Speicherung der einzelnen Schritte des Analyseprozesses in der Datenbank kann der Benutzer beim nächsten Programmstart mit seinen bereits durchgeführten Analysen direkt weiterarbeiten.

Die Berechnungen und Visualisierungen werden in sogenannten „History-Nodes“ dargestellt. Eine History-Node ist eine grafische Repräsentation einer einzelnen Berechnung oder Visualisierung. Sie enthält sowohl die Informationen zur Studienaufgabe, zum Stimulus, zur durchgeführten Berechnung oder angezeigten Visualisierung, zu den Probanden sowie das Datum der Berechnung als auch die Berechnungsergebnisse oder die Visualisierung selbst (dargestellt in (4) in Abbildung 1). Alle Ergebnisse können in einem Studienbericht automatisch zusammengefasst werden.

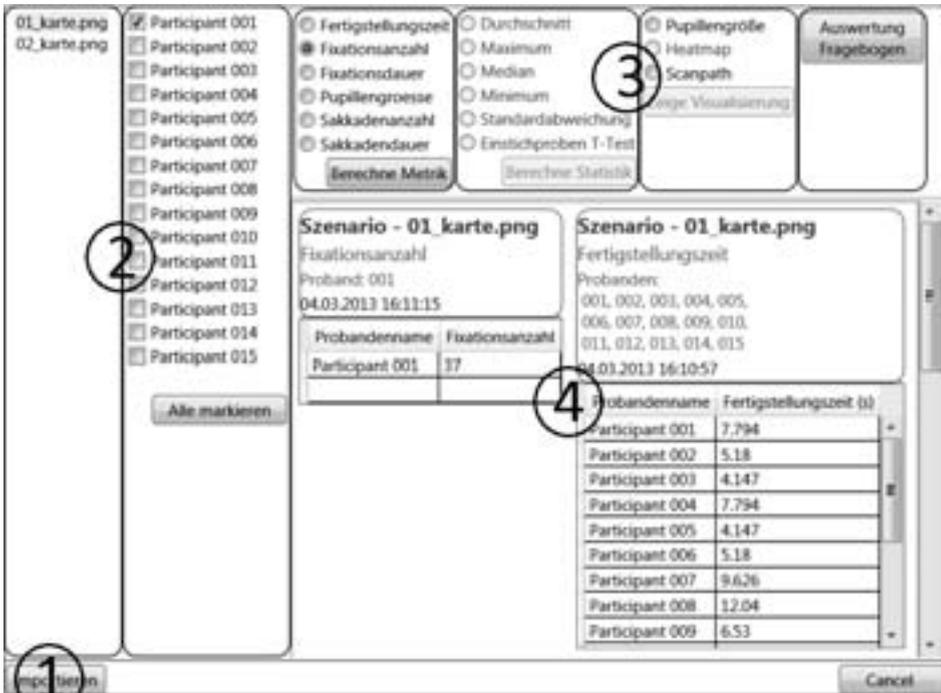


Abbildung 1: Das entwickelte Eyetracking-Framework eTaddy bietet dem Benutzer die Möglichkeit Eyetracking-Daten in eine Datenbank zu importieren (1), Stimuli und Probanden für die Analyse auszuwählen (2) und verschiedene Metriken, Statistiken oder Visualisierungen zu verwenden (3). Alle Ergebnisse der Analyse werden durch „History-Nodes“ dargestellt (4).

eTaddy¹ ist ein in C# programmiertes Framework und kann auf jedem Standard-Windows-PC mit einem .NET-Framework 3.5 oder höher verwendet werden.

4 Zusammenfassung und Ausblick

In diesem Artikel wurde ein integratives Eyetracking-Framework für die Erstellung, Durchführung und Analyse von Eyetracking-Experimenten vorgestellt. Das entwickelte Eyetracking-Framework unterstützt den Versuchsleiter bei der Analyse der Ergebnisse mit Hilfe von statistischen Analyseverfahren und Visualisierungen. Zusätzliche Metriken, Statistiken und Visualisierungen können durch ein Plug-In-System hinzugefügt werden.

Das Framework wurde in bereits durchgeführten Eyetracking-Experimenten erfolgreich eingesetzt. Derzeit wird es für die Auswertung von Benutzerstudien im Umfeld der Chronobiologie erweitert. Geplant ist zusätzlich der Einsatz im Rahmen von Fahrsimulationsexperimenten sowie auf dem Gebiet der Kognitionswissenschaften.

¹Für weitere Informationen über die Software kontaktieren Sie bitte die Autoren.

Literatur

- [1] Ponsoda, V., Scott, D., Findlay, J. M.: A probability vector and transition matrix analysis of eye movements during visual search. *Acta Psychologica* 88(2), 167–185 (1995)
- [2] Nielsen, J., Pernice, K.: *Eyetracking Web Usability*. New Riders, USA (2010)
- [3] Goldberg, J. H., Kotval, X. P.: Computer interface evaluation using eye movements: methods and constructs. *International Journal of Industrial Ergonomics* 24(6), 631–645 (1999)
- [4] Privitera, C. M., Stark, L. W.: Algorithms for Defining Visual Regions-of-Interest: Comparison with Eye Fixations. *IEEE Trans. Pattern Anal. Mach. Intell.* 22(9), 970–982 (2000)
- [5] Goldberg, J. H., Helfman, J. I.: Comparing Information Graphics: A Critical Look at Eye Tracking. *Proceedings of BELIV'10: BEyond time and errors: novel evaLuation methods for Information Visualization* (2010)
- [6] Tsang, H. Y., Tory, M., Swindells, C.: eSeeTrack—visualizing sequential fixation patterns. *IEEE Transactions on Visualization and Computer Graphics* 16(6), 953–62 (2010)
- [7] Duchowski, A. T., Price, M. M., Meyer, M., Orero, P.: Aggregate gaze visualization with real-time heatmaps *Proceedings of the Symposium on Eye Tracking Research and Applications*, 13–20 (2012)
- [8] Raschke, M., Xuemei, C., Ertl, T.: Parallel Scan-Path Visualization. *Proceedings of the 2012 Symposium on Eye-Tracking Research and Applications*, 165–168 (2012)
- [9] West, J.M., Haake, A.R., Rozanski, E.P., Karn, K.S.: eyePatterns—Software for Identifying Patterns and Similarities Accross Fixation Sequences *Proceedings of the 2006 Symposium on Eye-Tracking Research and Applications*, 149–154 (2006)

Fog of Triage: Usage of the Fog of War concept for the Triage in a Mass Casualty Incident (MCI)

Sebastian Klingenbeck, Tayfur Coskun, Teodora Velikova, Johannes Roith,
Gudrun Klinker

Fakultät für Informatik
Technische Universität München
Boltzmannstraße 3
85748 Garching bei München

{klingenb, coskun, velikova, roith, klinker}@in.tum.de

Abstract: This paper describes the adaption of the Fog of War concept, known from real-time strategy games, to the triage in a mass casualty incident (MCI). During the triage the patients are categorized according to the severity of their injuries. The main goal of this study is to investigate if the Fog of War concept improves the performance of the triage process. The Fog of Triage (FoT) concept has been implemented and evaluated in an user-centered-study with the Arbeiter-Samariter-Bund (ASB) München¹. Two applications have been implemented: Digital Triage application and Fog of Triage application. The Fog of Triage applications extend the Digital Triage application using the Fog of War concept. In an evaluation with six subjects the Digital Triage application was compared to the Fog of Triage application. The measured parameters were: time per patient, System Usability Scale (SUS) and NASA Task-Load-Index (NASA-TLX). In addition a qualitative feedback was gathered through interviews. There were no statistically significant differences between these two applications considering the measured parameters. In the interviews the majority of the subjects rated the Fog of Triage concept as useful. Therefore, the FoT provides an additional benefit without reducing the performance of the digital triage. This research is part of the SpeedUp project².

1 Motivation

The main goal of the SpeedUp project is to speed up the triage in a mass casualty incident (MCI). In an MCI not all injured can be treated immediately. The patients are categorized according to the severity of their injuries to maximize the chance of survival. This categorization is called triage. Seriously injured patients are treated immediately, whereas

¹The Worker's Samaritan Federation (Arbeiter-Samariter-Bund Deutschland, ASB) is an aid and welfare organization devoted to rescue services.

²The project SpeedUp is funded by the German Federal Ministry of Education and Research (BMBF) within the program „Research for Civil Security“ (May 1st, 2009 - April 30th, 2012, FKZ: 13N10175). Website: <http://www.speedup-projekt.de>

the treatment of slightly injured patients is delayed. The probability of mistakes and inaccuracies increases because of the stressful and chaotic situation during the triage process. For instance, it can happen that a paramedic has to search for an injured person before first aid care can be provided and transport to hospital can be arranged. This additional effort requires a lot of time and reduces the chance of survival of all injured, not only of one individual. With a digital triage system inaccuracies could be avoided. In addition, it is helpful to know in which sections the triage has been completed to avoid unneeded screenings. The Fog of War concept from real-time strategy games is a possibility to keep track of screened areas.

2 Related Work

Other projects in the context of digital aid for triage in an MCI are, for example, e-Triage and WIISARD: The e-Triage project is a digital triage system using tablet PCs for data entry and distributed databases for data synchronization between paramedics and emergency service operation controls [DEA⁺11]. The digital triage system developed in the WIISARD-project uses mobile WiFi hotspots with GPS, which are deployed to build up an independent communication infrastructure. Patients that have been screened during the triage are labeled with intelligent patient tags. These tags provide patient location and identification. They store local triage data and send them to the operation controls [KCB⁺06][LPCR05].

The Fog of War concept, appearing first in the real-time strategy game Warcraft II³, hides areas that are not observed by the player. This means that enemy units are hidden from the player, while the terrain is still visible beneath a fog. The player not only has to explore the map once, but also has to observe the map continuously with his units.

To the best of our knowledge no project has yet investigated the usage of the Fog of War concept for the triage of an MCI. In our study we present the first approach to apply Fog of War to the triage process.

3 Method

In a user-centered development four Android applications were developed: Fog of Triage (FoT) application, used by the paramedics for the triage of the patients, FoT overview application, used by the ambulance incident commander (AIC) to get an overview of the position of the patients, Digital Triage application, used by the paramedics and Digital Triage overview application, used by the AIC. Our approach in the evaluation was to conduct a comparison between the digital applications and the FoT applications. The applications were developed with Android 4.0.3 using the Google Maps API.

³See: <http://arstechnica.com/old/content/2005/10/gamingevolution.ars/12> (visited 2012-12-04)

3.1 Apparatus

The FoT application and the Digital Triage application were installed on an ASUS EeePad Transformer TF101, equipped with an anti-reflection display foil from Diphos. The two overview applications were installed on an ASUS EeePad Transformer Prime TF201. The LG Smartphone P920 Optimus 3D was used as an access point for mobile internet.

3.2 Evaluation

During the evaluation the Digital Triage application was compared to the FoT application. Six participants from the ASB took part. The participants were between 25 and 59 years old, therefore the average age was 41.5 years. The hypothesis of the evaluation was that the FoT concept improves the speed of the triage without increasing the stress for the paramedics.

4 Results

The following sections present the most important qualitative and quantitative results of the evaluation.

4.1 Quantitative Results

To test the hypothesis of the evaluation Time per patient, System Usability Scale (SUS) value and NASA Task Load Index (NASA-TLX) value were measured. The SUS indicates how usable an application seems to be [Bro96], whereas the NASA-TLX measures how demanding a task appears to be for the user [HS88].

Application	Time per patient	SUS	NASA-TLX
Digital Triage	29.67s (6.38s)	76.25 (13.94)	7.08 (1.51)
Fog of Triage	29.83s (18.12s)	77.92 (13.39)	7.64 (1.02)

Table 1: Results of time per patient, SUS and NASA-TLX and their standard deviation in brackets

Paired t-tests failed to prove that these differences are statistically significant.

4.2 Qualitative Results

Qualitative feedback was gathered in interviews. The most important results were: Transmitting the triage data directly to the AIC was considered to be useful; all applications were easy to use; the Fog of Triage concept was reasonable; a scale could be useful to get an estimation of the revealed area.

5 Discussion and Future Work

The quantitative results are not statistically significant, but the qualitative results are promising. It can be concluded that the Fog of Triage concept yields a benefit for the triage and does not have a negative influence on the usability. The significance could be revisited in another evaluation with more subjects. We recommend to implement these features in the next prototype:

Define the section on the map; move and remove the patient markers (i.e. when a patient is transported to the hospital); save the intermediate steps for each patient in the triage algorithm in order to obtain a more accurate first diagnosis; manually specify the size of the revealed area; place comment markers on the map.

The following features were required by the participants in the interviews:

Manually specify the size of the revealed area; the use of a bigger font; adapt the buttons for bigger hands; the AIC should have a scenery overview and the total number of patients in the area; display a scale to estimate the dimension of the revealed area; add additional information to the patients (i.e. patient has a craniocerebral injury). These features should be implemented in the next iteration of the prototype.

References

- [Bro96] J. Brooke. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189:194, 1996.
- [DEA⁺11] Anton Donner, Stefan Erl, Christine Adler, Anton Metz, and Marion Krüsmann. Projekt e-Triage: Datenmanagement für die elektronische Betroffenenfassung und Akzeptanz bei Rettungskräften. 2011.
- [HS88] S.G Hart and L.E Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Human mental workload*, 1:139–183, 1988.
- [KCB⁺06] James P. Killeen, Theodore C. Chan, Colleen Buono, William G. Griswold, and Leslie A. Lenert. A wireless first responder handheld device for rapid triage, patient assessment and documentation during mass casualty incidents. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, pages 429–433, 2006.
- [LPCR05] Leslie A. Lenert, Douglas A. Palmer, Theodore C. Chan, and Ramesh Rao. An Intelligent 802.11 Triage Tag for medical response to disasters. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, pages 440–444, 2005.

Koaktivitäts- und Interaktivitätsbewertung potentieller sozialer Lernplattformen

Rüdiger Marwein

Hochschule Reutlingen
Medien- und Kommunikationsinformatik
Ruediger.Marwein@Student.Reutlingen-University.DE

Art der Arbeit: Projektarbeit “Interaktive Systeme”

Betreuer/in der Arbeit: Dr. rer. nat. Kai Holzweißig, Prof. Dr. rer. nat. Gabriela Tullius

Abstract: Digitale Werkzeuge im Lehr- und Lernkontext werden unter dem Begriff “eLearning” zusammengefasst. Sie sind in der Regel relativ starre Sender-Empfänger-Modelle. Einzelpersonen lesen sich vorgegebene Inhalte an, üben diese ein und weisen schlussendlich das angeeignete Wissen mit einem Multiple-Choice-Test nach. Diese Werkzeuge sind mit modernen Lehrmethoden, die stark auf Gruppenarbeit setzen, nur schwer vereinbar. Es wird ein Konzept untersucht, das solche Lernumgebungen beschreibt, und in Korrelation zu bereits existierenden Lösungen gesetzt.

1 Motivation

Moderne Lehrmethoden im konstruktivistischen Sinne setzen stark auf Gruppenarbeit und soziale Interaktion. Um diese Methode auch für räumlich verteilte Gruppen zu nutzen scheinen netzwerkgestützte und insb. soziale Anwendungen prädestiniert zu sein. Die Qualität der Gruppenarbeit lässt sich jedoch nicht ohne Weiteres sinnstiftend in den digitalen Raum übertragen.

Keil entwickelt mit dem Konzept der „Medi@rena“ ein Rahmenwerk für eine kollaborative Lern- und Arbeitsumgebung und leitet dessen Qualitäten her [KE10a] [KE10b]. Obwohl die Medi@rena als Rahmenwerk für neue Software entstand, wird die Hypothese aufgestellt, dass sie als Bewertungsgrundlage zur Auswahl geeigneter Softwareprodukte zweckdienlich ist.

2 Konzept und Qualitäten der Medi@rena

Um die in [KE10a] hergeleiteten Qualitäten einem breiteren Publikum zugänglich zu machen, wurde die Essenz der Beschreibungen in acht Kernaussagen – je vier zu “Interaktivität” und “Koaktivität” - zusammengefasst.

Interaktivität	Responsivität	Die Anwendung reagiert nahezu sofort auf Manipulation.
	Auswertung	Es ist dem Anwender möglich, verfügbare Daten erkenntnisstiftend anders anzuzeigen, ohne sie hierfür neu erfassen zu müssen.
	Objektorientierung	Kreierbare Objekte lassen sich zu größeren Gebilden zusammensetzen. Die Eigenschaften der Objekte sind änderbar.
	Manipulation	Die Anwendung bietet mehrere Zugänge zur Manipulation von Objekten.
Koaktivität	Verteilte Persistenz	Die Daten der Anwendung werden in einem Netzwerk gespeichert; ggf. redundant. Der Nutzer nimmt es als einen einzigen Speicherort wahr.
	Berechtigung	Die Anwendung gestattet, Zugriffsrechte auf Objektebene zu vergeben. Informationen können privat und öffentlich sein.
	Ereignisbehandlung	Die Nutzer werden gegenseitig über Aktivitäten anderer Nutzer und über Veränderungen am System echtzeitnah informiert.
	Koordination	Die Informationen über aktuelle Vorgänge im System genügen, um gemeinsame Arbeitsschritte besser durchzuführen.

3 Untersuchung ausgewählter Umgebungen

Zur Prüfung der durch die Medi@rena beschriebenen Kriterien wurden vier Anwendungen gewählt, die insbesondere unter Studenten als Lern- und Arbeitswerkzeug intensiv genutzt werden, aber nicht in den klassischen eLearning-Bereich (wie bspw. das OpenSource Produkt „moodle“) fallen. Zur Einstufung der Ausprägung je Kriterium wurde eine 4-stufige Likert-Skala (0-3) eingesetzt.

Die extrahierten Kernaussagen wurden bei der Untersuchung der einzelnen Qualitäten auf Funktionen in den Produkten abgebildet. Dabei wurde versucht, die Funktionen der Anwendungen anschlussfähig auszulegen, sie also auch entgegen der Absichten der Entwickler zu nutzen.

Bei den analysierten Anwendungen handelt es sich um die Gruppen-Funktion in „Facebook“, den Kurzmitteilungsdienst „Twitter“, die kollaborativen Cloud-Anwendung „Google Drive“ und das kollaborative Whiteboard „Conceptboard“.

Es ist rasch ersichtlich, welche Anwendungen eher als Lernplattform geeignet sind.

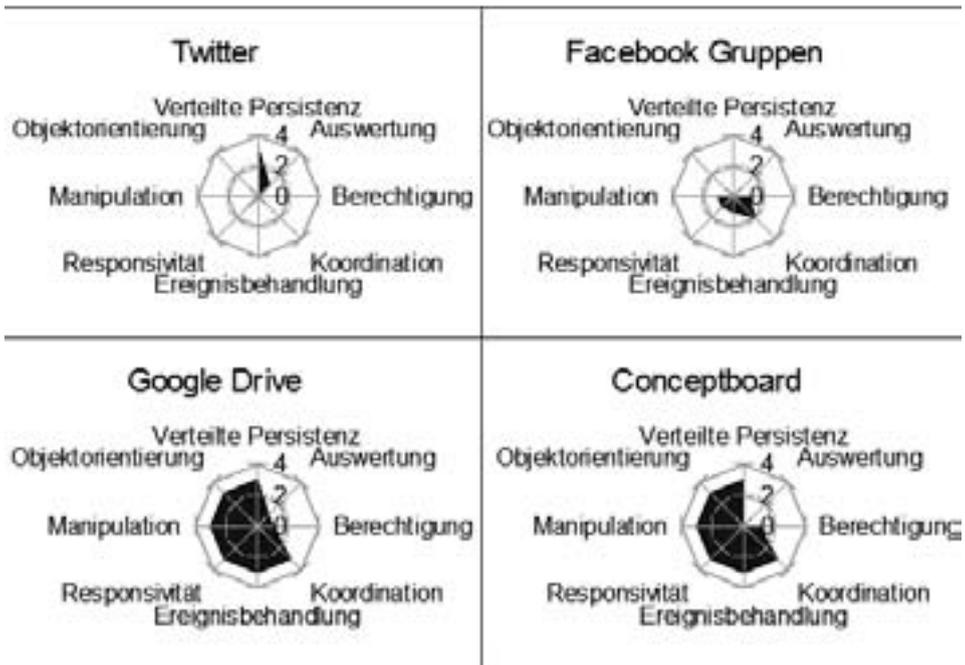


Abbildung 1: Ausprägung der Medi@rena-Kriterien in vier exemplarischen Anwendungen.

„Google Drive“ und „Conceptboard“ decken die Kriterien einer Medi@rena stärker ab, da sie darauf ausgelegt sind, gemeinsam und möglichst ohne Medienbrüche an einem Artefakt zu arbeiten, wengleich Google Drive sogar mehrere Dokumentarten anbietet. Die Vielfalt der erarbeitbaren Inhalte ist jedoch kein direktes Kriterium der Medi@rena.

Auffällig ist auch, dass die Kriterien „Berechtigung“ und „Auswertung“ durchweg entweder gar nicht oder nur sehr schwach ausgeprägt sind. Ein möglicher Grund hierfür kann über das Technology Acceptance Model 2 (TAM2) [VE00] (siehe auch 4.) hergeleitet werden, da beide Kriterien Einfluss auf die Komplexität der Anwendung haben. Trübt das zu erlernende Regelwerk bspw. die wahrgenommene Einfachheit, mindert das die wahrgenommene Nützlichkeit nach TAM2 und führt somit zu Akzeptanzproblemen [VE00].

4 Infrastrukturelle Implementierung

Das Technology Acceptance Model (TAM) [DA89] und TAM2 [VE00] wurde auf Merkmale einer sinnvollen Nutzungs-Infrastruktur hin untersucht.

[VE00] kommt zu einer Reihe empirisch ermittelten Ergebnissen bezüglich subjektiver Norm und empfundener Nützlichkeit. Konnte eine Anwendung durch den Medi@rena-Abgleich ermittelt werden, kann die Infrastruktur rund um die Anwendung mit Hilfe des TAM2 gestaltet werden. Um eine optimale Arbeitsumgebung unter Rücksichtnahme dieser Erkenntnisse zu implementieren, scheinen folgende Faktoren zielführend zu sein:

- In der Lernumgebung muss es eine oder mehrere Personen (Tutor/Coach) geben, die sich schon gut mit der Software auskennen, den Einsatz befürworten und Hilfestellung bei Problemen geben können. So wird die „subjektive Norm“ positiv vorgebelegt.
- Da sich verpflichtender Einsatz in Kombination mit subjektiver Norm (Tutor/Coach) positiv auf die Nutzungsabsicht auswirkt, sollte die Nutzung verpflichtend sein.
- Um der Software eine Relevanz bei der Erarbeitung von Wissen zu geben, sollten die Ergebnisse der Arbeit auch mit bzw. innerhalb der Software präsentiert werden.

5 Fazit und Ausblick

Auch wenn die untersuchten Anwendungen noch nicht vollständig der Medi@rena entsprechen, so kommen sie den Anforderungen doch schon sehr nahe. Es wurde gezeigt, dass die Kriterien gut auf bestehende Software anwendbar sind und sich tägliche Anwendungen dadurch identifizieren lassen.

Die Bewertung ist trotz der Definition stets subjektiv. Eine gemeinschaftliche Plattform, die eine fallspezifische Anwendungs-Bewertung anhand der Kriterien ermöglicht, wäre für Lehrkräfte sinnvoll.

Literaturverzeichnis

- [KE10a] Keil, R.: Das Differenztheater – koaktive Wissensarbeit als soziale Selbstorganisation, In (Hannelore Bublitz u.a., Hrsg.) Automatismen, Wilhelm Fink Verlag, München, 2010, S.205-229.
- [KE10b] Keil, R.: eLearning 2.0 vom Kopf auf die Füße gestellt, In: Jahrbuch Medienpädagogik 8, VS Verlag für Sozialwissenschaften, Wiesbaden, 2010, S. 121-146
- [DA89] Davis, F. D.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology., MIS Quarterly 13 (3), 1989 , S. 319–340.
- [VE00] Davis, F. D., Venkatesh, Viswanath: A Theoretical Extension of the technology Acceptance Model: Four Longitudinal Studies. In: Management Science, vol 46, No.2., 2000, INFORMIS, Hanover, MD 21076 USA, 2000; S. 186-204

Markerloses, modellbasiertes Echtzeit-Tracking für AR-Applikationen

Jessica Millberg
Institut für Visual Computing
Hochschule Bonn-Rhein-Sieg
Jessica.Millberg@h-brs.de

Art der Arbeit: Masterarbeit

Betreuer der Arbeit: Prof. Dr.-Ing. Rainer Herpers, Prof. Dr. André Hinkenjann

GI-Ansprechpartner: Prof. Dr. Sascha Alda

Abstract: Augmented Reality (AR) findet heutzutage sehr viele Anwendungsbereiche. Durch die Überlagerung von virtuellen Informationen mit der realen Umgebung eignet sich diese Technologie besonders für die Unterstützung der Benutzer bei technischen Wartungs- oder Reparaturvorgängen. Damit die virtuellen Daten korrekt mit der realen Welt überlagert werden, müssen Position und Orientierung der Kamera durch ein Trackingverfahren ermittelt werden. In dieser Arbeit wurde für diesen Zweck ein markerloses, modellbasiertes Trackingsystem implementiert. Während einer Initialisierungs-Phase wird die Kamerapose mithilfe von kalibrierten Referenzbildern, sogenannten Keyframes, bestimmt. In einer darauffolgenden Tracking-Phase wird das zu trackende Objekt weiterverfolgt. Evaluiert wurde das System an dem 1:1 Trainingsmodell des biologischen Forschungslabors Biolab, welches von der Europäischen Weltraumorganisation ESA zur Verfügung gestellt wurde.

1 Einleitung

In dieser Arbeit wird ein markerloses, modellbasiertes Trackingsystem vorgestellt. Modellbasierte Trackingansätze lassen sich prinzipiell in zwei Kategorien einteilen: Tracking durch Detektion und rekursives Tracking [LSFK10]. Tracking durch Detektion ermittelt die Kamerapose durch das Matching von Merkmalen zwischen dem aktuellen Kameriframe und einem Referenzbild. Rekursives Tracking, auch Frame-To-Frame Tracking genannt, ermittelt die Kamerapose durch einen Minimierungsprozess. Dazu wird allerdings eine ungefähre Schätzung der Pose benötigt, wozu in der Regel die bekannte Kamerapose des vorherigen Frames verwendet wird.

Da für Tracking durch Detektion kein Wissen über eine frühere Kamerapose bekannt sein muss, wird dieses Verfahren in den meisten modellbasierten Trackingansätzen [LVTF03, PHMG06, PLW11] für die Initialisierung benutzt. Die Referenzbilder werden dabei auch als Keyframes bezeichnet. Da das Matching mit einer Keyframedatenbasis unter Umständen sehr zeitintensiv sein kann, benutzen die erwähnten Verfahren einen rekursiven Ansatz zur Fortführung des Trackings.

2 Markerloses, modellbasiertes Tracking am Biolab

2.1 Erstellung der Keyframes

Für die Erstellung der Keyframes wurde ein Softwaretool eingesetzt, welches durch eine Kette von semiautomatischen Schritten Keyframes erzeugt. Der Benutzer lädt zunächst ein aufgenommenes Keyframebild, intrinsische Kameraparameter sowie ein 3D-Modell des zu trackenden Objektes. Im Anschluss daran wird das Bild mit dem 3D-Modell überlagert, welches sich in allen Freiheitsgraden durch die Maussteuerung bewegen lässt. Sobald die Überlagerung für den Benutzer optisch zufriedenstellend ist, lässt sich die dazugehörige Kamerapose abspeichern. In einem letzten Schritt werden die Keyframeinformationen in ihrer Gesamtheit erfasst. Dabei werden auf dem Keyframebild Merkmale mit dem SURF Detektor [BTG06] gesucht. Zu den gefundenen Merkmalen werden mit der bekannten Kamerapose und dem 3D-Modell Schnittpunkttests durchgeführt. Die 3D-Koordinaten der getroffenen Geometrie und die dazugehörige 2D-Information aus dem Bild ergeben 2D/3D-Korrespondenzen, die in einer XML-Datei zusammengefasst gespeichert werden.

2.2 Initialisierung

Während der Initialisierung wird die Kamerapose ermittelt, ohne dass bereits Zusatzinformationen zur vorherigen Pose bestehen. Zunächst werden mit dem SURF Algorithmus markante Merkmale im Bild gesucht und in Deskriptoren zwecks Wiedererkennung beschrieben. Zur Optimierung der Laufzeit wurde die SURF Implementierung „Speeded Up SURF“ [FTK10] eingesetzt, welche die rechenaufwendigen Operationen auf die GPU auslagert. Das Matching der Merkmale wurde mit der FLANN-Bibliothek [ML09] realisiert. Die 3D-Koordinaten der Keyframe Merkmale können somit an die Merkmale im Kameriframe weitergereicht werden. Aus diesen gewonnenen 2D/3D-Korrespondenzen wird die Kamerapose mit dem EPnP-Algorithmus [LMNF09] berechnet, wobei RANSAC [FB81] zur robusten Eliminierung von Ausreißern eingesetzt wird. Zur Verfeinerung der Pose wird schließlich eine Levenberg-Marquardt Minimierung eingesetzt in Kombination mit dem Huber M-Schätzer. Ein beispielhafter Initialisierungsablauf ist in Abb. 1 dargestellt.

2.3 Tracking

Nach einer erfolgreichen Initialisierung wird die Tracking-Phase eingeleitet. In diesem Verarbeitungsschritt wird der, auf dem optischen Fluss basierende, Kanade-Lucas-Tomasi (KLT) [ST94] Algorithmus verwendet. Zunächst findet einmalig ein Vorverarbeitungsschritt statt. In diesem werden Merkmale mit dem Shi-Tomasi Detektor gesucht. Zu jedem dieser Merkmale wird ein Schnittpunkttest, anhand der aus der Initialisierung bekannten Pose, durchgeführt, der bei Erfolg die 3D-Koordinate des Schnittpunkts zurückgibt. Die 2D-Merkmale der so ermittelten 2D/3D-Korrespondenzen werden schließlich im weiteren



Abbildung 1: Ablauf der Initialisierung verdeutlicht am 1:1 Biolab-Trainingsmodell.

Trackingverlauf verfolgt. Anschließend wird erneut mit einem Minimierungsverfahren die alte Kamerapose zu einer aktuellen verfeinert. Diese Vorgehensweise wird solange iteriert bis das Tracking ein Abbruchkriterium erreicht (zu wenig Merkmale verfolgt, Reprojektionsfehler der Merkmale zu groß), woraufhin erneut die Initialisierung aufgerufen wird.

3 Evaluation & Ausblick

Das entwickelte Trackingsystem wurde an dem 1:1 Biolab-Trainingsmodell (siehe Abb. 1) in verschiedenen Szenarien, bei denen eine Translation bzw. Rotation der Kamera oder eine Verdeckung des Modells durchgeführt wurde, evaluiert. Durch die Verwendung eines markerbasierten Trackingsystems wurden Ground Truth Daten ermittelt. Getestet wurde auf einem Intel Xeon E5630 (2,53 GHz) Desktop PC mit einer NVIDIA Quadro FX 580 unter Verwendung einer Logitech Quickcam Pro mit einer Auflösung von 640 x 480 Pixeln. Der mittlere Fehler in Translation und Rotation übersteigt in allen Szenarien nie Werte über 5 cm bzw. 4° und bleibt meistens sogar deutlich darunter. Mit einer durchschnittlichen Framerate von 18.50 FPS eignet sich das System für Augmented Reality Anwendungen die Echtzeitanforderungen haben. Weitere Tests zeigten, dass die Geschwindigkeit durch neuere Hardware noch deutlich gesteigert werden kann.

Da sich das Trackingsystem robust gegenüber Verdeckungen und Veränderungen der Szene verhielt, könnte es in der Zukunft für die AR-gestützte Durchführung von Wartungs- und Reparaturarbeiten an der Biolabanlage genutzt werden. Durch die Integration des Trackingsystems in eine AR-Autoren-Software könnten die existierenden Biolabprozeduren für Augmented Reality Anwendungen als AR-Prozeduren weiterentwickelt werden, die dann Zusatzinformationen aufnehmen können um somit zu einer höheren Effizienz bei Wartungs- und Experimentierarbeiten zu führen.

Danksagung

Diese Arbeit wurde in Kooperation mit dem Deutschen Zentrum für Luft- und Raumfahrt am Institut für Luft- und Raumfahrtmedizin durchgeführt. Die Evaluierung des Tracking-systems an dem 1:1 Trainingsmodell konnte nur dank Unterstützung der Europäischen Weltraumorganisation ESA ermöglicht werden, die die Räumlichkeiten mit dem Trainingsmodell im Europäischen Astronautenzentrum zur Verfügung stellten. Des Weiteren konnten mit freundlicher Genehmigung der ESA die Bilder des Trainingsmodells in dieser Arbeit veröffentlicht werden.

Literatur

- [BTG06] Herbert Bay, Tinne Tuytelaars und Luc Van Gool. Surf: Speeded up robust features. In *In ECCV*, Seiten 404–417, 2006.
- [FB81] M A Fischler und R C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [FTK10] Paul Furgale, Chi Hay Tong und Gaetan Kenway. Speeded Up SURF. Webseite, 2010.
- [JYQ10] Jia Jun, Qi Yue und Zuo Qing. An Extended Marker-Based Tracking System for Augmented Reality. In *Proceedings of the 2010 Second International Conference on Modeling, Simulation and Visualization Methods, WMSVM '10*, Seiten 94–97, Washington, DC, USA, 2010. IEEE Computer Society.
- [LMNF09] V. Lepetit, F. Moreno-Noguer und P. Fua. EPnP: An Accurate $O(n)$ Solution to the PnP Problem. *International Journal of Computer Vision*, 81(2):155166, 2009.
- [LSFK10] Joo Paulo Lima, Francisco Simes, Lucas Figueiredo und Judith Kelner. Model Based Markerless 3D Tracking applied to Augmented Reality. *SBC Journal on 3D Interactive Systems*, 1:2–15, 2010.
- [LVTF03] V. Lepetit, L. Vacchetti, D. Thalmann und P. Fua. Fully Automated and Stable Registration for Augmented Reality Applications. In *Proceedings of the International Symposium on Mixed and Augmented Reality*, 2003.
- [ML09] Marius Muja und David G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09*, Seiten 331–340. INSTICC Press, 2009.
- [PHMG06] Juri Platonov, Hauke Heibel, Peter Meier und Bert Grollmann. A mobile markerless AR system for maintenance and repair. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR '06*, Seiten 105–108, Washington, DC, USA, 2006. IEEE Computer Society.
- [PLW11] Youngmin Park, Vincent Lepetit und Woontack Woo. Extended Keyframe Detection with Stable Tracking for Multiple 3D Object Tracking. *IEEE Trans. Vis. Comput. Graph.*, 17(11):1728–1735, 2011.
- [ST94] Jianbo Shi und Carlo Tomasi. Good Features to Track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seiten 593 – 600, 1994.

Ambient Assisted Living: Aktueller Stand und Ausblick

Alexander Phleps

Technische Universität Berlin / Fraunhofer HHI
Interactive Media – Human Factors
alexander.phleps@campus.tu-berlin.de

Art der Arbeit: Werkstudententätigkeit in Forschungsprojekt; Masterarbeit
Betreuer/in der Arbeit: Detlef Ruschin

Abstract: Europa investiert in Ambient Assisted Living (AAL). Über das AAL Joint Programme wurden bereits über 100 Projekte mit Budgets zwischen 1-7 Millionen Euro finanziert. Dabei geht es um die Frage, wie man durch intelligente Umgebungen, getrieben von modernen Informationstechnologien, alternde Menschen dabei unterstützen kann länger selbstständig zuhause leben zu können. Der gesellschaftliche Wandel und der daraus resultierende Pflegenotstand sind deutliche Anzeichen dafür, dass Handlungsbedarf besteht. Die Tatsache, dass Menschen so lange wie möglich zuhause leben wollen, ist unumstritten. Dieser Artikel gibt einen Überblick über aktuelle Technologien, die solche Anwendungen überhaupt erst möglich machen. Es werden Entwicklungen vorgestellt, die das Potential dazu haben, Projekte dieser Art voranzutreiben. Anschließend werden einige der wichtigsten Konzepte der AAL Forschung anhand von Lösungsansätzen des EU Projekt USEFIL sowie Alternativen anderer Forschungsprojekte beschrieben. Abschließend werden Fragen diskutiert, die aus Sicht des Autors gestellt werden müssen, wenn man sich mit Technologien beschäftigt, die einen solch großen Einfluss auf das direkte Lebensumfeld von Menschen haben.

1 Technischer Stand

Informations- und Telekommunikationstechnologien sind die treibende Kraft der AAL-Forschung. Computer galten jedoch lange Zeit gerade bei älteren Mitbürgern als viel zu kompliziert. Ein wichtiger Meilenstein wurde hier mit der Marktreife von Smartphone und Tablet-Computern erreicht. Moderne Mobil-Betriebssysteme (iOS, Android, etc.) stehen für einen radikalen Paradigmenwechsel vom klassischen Desktop-Computing mit WIMP-Interface (Windows Icon Menu Pointer). Zum einen verfügen die User-Interfaces (UI) über relativ große Bedienelemente, dadurch wird die Ungenauigkeit des menschlichen Fingers im Vergleich zum Cursor der Maus kompensiert. Zum anderen sind Mobile-Anwendungen (Apps) meist sehr einfach gestaltet und verfügen über weniger Funktionen als Desktop-Anwendungen. Diese Eigenschaften kommen vor allem älteren Benutzern zu Gute. Die Interaktion, vor allem das Tippen und Zoom-Gesten, stellt keine Barriere dar. So erzielten Tablets eine hohe Zufriedenheit und Akzeptanz in dieser Zielgruppe [We12]. Darüber hinaus konnte auch eine hohe Akzeptanz von Pflegenden festgestellt werden [Sc12]. Für AAL-Projekte bedeutet dies, dass digitale Informationen sowohl für Patienten als auch für Pfleger verfügbar gemacht werden können, ohne die Verwendung von Desktop- oder Laptop-Computern vorauszusetzen.

Ein weiterer Meilenstein war die Marktreife günstiger Tiefensensoren (Microsoft Kinect, Asus Xtion, SoftKinetic DepthSense). Dabei handelt es sich um spezielle Kameras welche Bilder mit Tiefeninformationen liefern. Jeder Pixel eines solchen Tiefenbildes enthält Informationen zur Entfernung eines jeden Pixels zur Kamera. Dies eröffnet neue Möglichkeiten für Computer-Vision Algorithmen, da auf eine weitere Dimension zur Informationsgewinnung zurückgegriffen werden kann. So ist es möglich, Personen und ihre Körperhaltung zu erkennen, indem die Position und Ausrichtung jedes Gelenks geschätzt wird [Sh11]. Im AAL-Kontext sind Tiefensensoren besonders für Aktivitäts- und Bewegungsanalyse interessant.

Neben diesen Neuerungen, die sich auf die Interaktion mit technischen Systemen beziehen, sind es vor allem Entwicklungen aus dem Bereich „Internet of things“ die eine treibende Kraft für AAL-Projekte darstellen. Twine [Su13] sind kleine, mit Sensoren bestückte Blöcke, deren Messwerte über verschiedene Kanäle (E-Mail, SMS, HTTP, etc.) weitergereicht werden können. Über ein Breakout-Board können weitere Sensoren hinzugefügt werden. NinjaBlock [Ni13] bieten noch mehr Flexibilität durch Open Source Hardware und Software. Sie können untereinander, sowie mit autarken Aktoren und Sensoren kommunizieren. Beide Systeme verfügen über Sensoren zum Auswerten von Temperatur, Beschleunigung, Feuchtigkeit und können sehr einfach durch IFTTT-Befehle („If-This-Then-That“) über ein Web-Interface programmiert werden. Das StartUp SmartThings hat es sich zum Ziel gesetzt, mit der gleichnamigen Plattform [Sm], eine Lösung schaffen, die verschiedene Funkstandards unterstützt, erweiterbar ist und die Integration von Drittanbieter-Komponenten erlaubt. Als erster kommerzieller Elektronikhersteller bietet Belkin unter dem Namen WeMo [Be13] Komponenten an, mit denen sich das Eigenheim per Smartphone kontrollieren und automatisieren lässt.

Der Implementierung benutzerfreundlicher und günstiger AAL-Lösungen steht angesichts der Vielzahl an vorhandener Lösungen aus technischer Sicht nichts im Wege.

2 Kernkonzepte der AAL-Forschung

Viele AAL-Projekte widmen sich den gleichen Problemen, gehen aber unterschiedliche Wege um Lösungen zu finden [Aa13]. Das USEFIL-Projekt (Unobstrusive Smart Environments For Independent Living) konzentriert sich dabei auf eine unauffällige Integration unter der Verwendung von Open Source Software und kostengünstiger Hardware [Us13]. Die für den Patienten bewusste Interaktion mit dem System findet über einen internetfähigen Fernseher (SmartTV) und einen Tablet-Computer statt. Der Fernseher stellt besondere Inhalte zur Verfügung (z. B. Spiele zum Training der kognitiven Fähigkeiten). Das Tablet dient in erster Linie zur Kommunikation mit Familie und Freunden, erlaubt es dem Nutzer aber auch die persönliche Gesundheitsakte einzusehen. Für die Aktivitätsüberwachung und Aufzeichnung von Vitaldaten kommt ein Armband mit verschiedenen Sensoren bzw. eine SmartWatch sowie ein Vital-Monitoring-Spiegel, Kameras, Tiefensensoren und Mikrofone zum Einsatz. Eine besondere Rolle übernimmt das Decision Support System (DSS), eine Software-Komponente zur Entscheidungsfindung, bei der die Informationen aller Sensoren zusammenfließen [Pa12].

2.1 Sturzerkennung

Im USEFIL-Umfeld verfügt der Patient über ein Armband bzw. eine SmartWatch, welche über verschiedene Sensoren verfügt, die u. a. Puls und Beschleunigung messen können. Anhand dieser Daten werden Muster erkannt die einen Sturz kennzeichnen. Der Eintritt eines solchen Ereignisses führt zu der Alarmierung von Ersthelfer, Familie und Arzt. Zusätzlich kann auch auf die bereits vorhandenen Tiefensensoren zurückgegriffen werden um Stürze zu erkennen. Deren Sichtbereich ist jedoch sehr eingeschränkt. Andere Projekte verwenden z. B. kapazitive Verfahren. Dabei werden Spannungsänderungen registriert, die durch einen menschlicher Körper ausgelöst werden. Die dafür nötige Sensorik kann im Boden verbaut werden [Li13].

2.2 Aktivitätsüberwachung

Die Aktivitäten des täglichen Lebens (ATL bzw. ADL) stellen ein Pflegemodell dar, welches dazu dient den Pflegebedarf eines Patienten zu ermitteln. Der Kriterienkatalog umfasst 12 Aktivitäten (u. a. Bewegung, Kommunikation, Ernährung). Ziel ist die Erhaltung der Selbstständigkeit in diesen Bereichen [Ju93]. Je nach Aktivität ist die Erkennung durch Sensorik mit unterschiedlich Herausforderungen verbunden. Für die Erfassung von Bewegungsdaten werden im USEFIL-Umfeld u. a. Tiefensensoren verwendet. Sie werden in einem Durchgangsbereich der Wohnung platziert der mehrmals täglich passiert wird. Dabei kann die Geschwindigkeit, Frequenz und Stabilität der Schritte abgeleitet werden um etwaige Abweichungen festzustellen [Br12]. Andere Projekte verwenden z. B. Ultraschallechos [St12] oder intelligente Stromzähler [Cl12].

2.3 Sozialer Kontakt

Im USEFIL-Projekt wird versucht der sozialen Isolation im Alter entgegen zu wirken, indem moderne Kommunikationstechnologien erweitert und auf einfache Weise zugänglich gemacht werden. Die Tablet-App zeigt im Hauptbildschirm große Kacheln, in denen sich Bilder der Familienmitglieder befinden. Im Hintergrund ist der aktuelle Kontext bzw. der Ort zu sehen, an dem sich die jeweilige Person befindet. Außerdem ist die aktuelle Aktivität in Form eines Piktogramms dargestellt. Dieser semiotische Ansatz zielt darauf ab, dass Patienten kein Vorwissen benötigen und den Status mit einem Blick erfassen können. Aus welcher Quelle eine Information kommt ist davon abhängig wie stark sich die Familienmitglieder an dem Projekt beteiligen möchten. Sie können festlegen, zu welchen Zeiten sie täglich das Haus verlassen und wann sie üblicher Weise in der Arbeit oder Schule eintreffen. Der Patient sieht in diesem Fall den geplanten, nicht den tatsächlichen Tagesablauf. Entscheidet sich das Familienmitglied dafür aktiver teilzunehmen, kann die App so konfiguriert werden, dass z. B. beim Eintreffen in der Arbeit dieser Kontext automatisch gesetzt wird (Geofencing). Möchten sie ihre derzeitige Aktivität kommunizieren, so kann diese aktiv gesetzt werden (z. B. „Kaffeepause“). Patienten und Familie können anhand dieser Informationen entscheiden wann sie Kontakt zueinander aufnehmen wollen. Die App wird neben den genannten Social Awareness Informationen auch den Austausch von Chat-Nachrichten, Erinnerungen und Fotos unterstützen.

3 Kritische Sichtweise

GI Präsident Oliver Günter forderte einst, dass sich Informatiker grundsätzlich ethische Fragen stellen sollten, denn ihre Disziplin greift bereits Heute in sämtliche Bereiche des täglichen Lebens ein [Gu13]. Nach meiner persönlichen Auffassung gilt dies besonders für den Bereich der AAL-Forschung.

Die aus der AAL-Forschung resultierenden Systeme werden in privaten Bereichen unseres Lebens, ohne aktives Eingreifen der „Benutzer“, rund um die Uhr Daten sammeln. Wie kann das Datenschutz-Problem gelöst werden, damit Patienten und ihre Familien solchen Systemen vertrauen können?

Eine Pflegekraft die durch eine AAL-Umgebung entlastet wird kann sich besser auf das Wesentliche, das Menschliche konzentrieren. Daher sollten solche Technologien die Altenpflege keinesfalls ersetzen sondern unterstützen. Doch wie kann sichergestellt werden, dass unsere Entwicklungen nicht irgendwann falsch eingesetzt werden? Wie weit können Gesetze und Lizenzbestimmungen gehen?

Ist der AAL-Weg, eine technische Strategie um ein soziales Problem zu lösen, überhaupt der richtige Weg oder nur die einzige Idee einer bereits so stark technisierten Gesellschaft?

Literaturverzeichnis

- [Aa12] AAL – Ambient Assisted Living: Catalogue of Projects, 2012.
- [Be13] <http://www.belkin.com/wemo/> (3. März 2013)
- [Br12] Braun, M.: Parametrisierung und Bedeutungsanalyse von Körperbewegungen mit Hilfe von Tiefenkameras. Bachelor Thesis, 2012.
- [Cl12] Clement, J.; Plönnigs, J.; Kabitzsch, K: Intelligente Zähler: ADLs erkennen und individualisieren. 5. Deutscher AAL-Kongress, Berlin, 2012.
- [Gu13] Günther, O.: Auswirkungen der Informatik auf die Gesellschaft erforschen und diskutieren. <http://www.gi.de/aktuelles/meldungen/detailansicht/article/auswirkungen-der-informatik-auf-die-gesellschaft-erforschen-und-diskutieren.html> (24. Januar 2013).
- [Ju93] Juchli, L.: Ganzheitliche Pflege: Vision oder Wirklichkeit. Recom Verlag, Kassel, 1993.
- [Li13] www.igd.fraunhofer.de/Institut/Abteilungen/Interaktive-Multimedia-Appliances/Projekte/Living-Lab (24. Januar 2013)
- [Ni13] <http://ninjablocks.com/> (24. Januar 2013)
- [Pa12] Papadopoulos, H. et.al.: Supporting tele-health and AI-based clinical decision making with sensor data fusion and semantic interpretation: The USEFIL case study. ECAI workshop NetMed, Montpellier, 2012.
- [Sc12] Schmeer et.al.: iPads in der Pflege - Nutzerakzeptanz und Nutzerverhalten von Pflegenden. 5. Deutscher AAL-Kongress, Berlin, 2012.
- [Su12] <http://supermechanical.com/twine/> (24. Januar 2013)
- [Sh11] Shotton, J. et.al.: Real-Time Human Pose Recognition in Parts from Single Depth Images. Computer Vision and Pattern Recognition (CVPR), Colorado Springs, 2011.
- [Sm13] <http://build.smarthings.com/> (24. Januar, 2013)
- [St12] Steen et.al.: Ein neuartiger Ansatz zur Indoor-Lokalisierung mittels dynamischer Veränderungen von Ultraschallechos. 5. Deutscher AAL-Kongress, Berlin, 2012.
- [Us13] <http://www.usefil.eu/> (24. Januar 2013)
- [We12] Werner et.al.: Evaluation eines Tablets (iPad) für und mit SeniorInnen. 5. Deutscher AAL-Kongress, Berlin, 2012.

Mobile Augmented Reality Browser in Combination with Magazines: Problems Concerning Usability and Content

Carolin Reichherzer, Jens Epe

University of Bremen
Department of Mathematics and Computer Science
carolinr@informatik.uni-bremen.de
jens.epe@uni-bremen.de
Type of Work: Seminar Paper in M. Sc. Digital Media
Advisor: Prof. Dr. Andreas Breiter

Abstract: Augmented Reality is present in science since years. Especially through the distribution of mobile devices it has found more and more use cases for users. However, although there are many different offers of applications the acceptance still seems to be low. This paper tries to figure out which problems the user encounters while using one representative mobile application and what could be improved to make it more usable.

1 Introduction

There has been much research and development concerning Augmented Reality (AR) for over forty years [DGB01]. However, despite the intensive research, AR applications are still not common among end-users.

There are several applications available for mobile devices which offer their users different reality enhancing features. Some so-called Augmented Reality browsers offer the different types bundled in one application and free of cost. In most cases the content is user generated and separated into different columns which can be selected by everyone. One type of them rests upon the location-based services and displays POIs of special categories like gastronomy, accommodations or historic sites in the camera image. The other type uses image recognition to augment magazines, printed ads or real objects by additional content.

Although there were already several magazines which made use of this modern technology in an AR Browser or a standalone application, it does not seem to be established yet. In principle it offers a high potential to the users, but there seem to be some problems discouraging them to use it wherefore we want to find out about it in the following research.

Our assumptions were proven to be true by other studies like in *Usability Testing of Augmented / Mixed Reality Systems* by Billinghamurst, who mentioned that only few experiments and user studies in Augmented Reality have been conducted. [Bi08] This is further strengthened by the study *A Survey of Evaluation Techniques Used in Augmented Reality Studies* which reviewed all published papers containing an evaluation on AR. The main findings of their qualitative research were that most of the articles evaluated enabling technologies or prototype applications. Merely 14% considered human computer interaction and further 8% described a formal user evaluation for usability testing within which they figured out the used evaluation techniques. These results and the causes for the rare testing are helpful to find out about the low acceptance of AR in our research. [DGB01]

2 Methodology

Our main research goal is to determine the major problems that appear during the use of AR browser and its content. On one hand we wanted to find out which problems encounter within the application caused for example by interface, content and technology. On the other hand we are interested in what additional augmented information the user expects from a magazine page and how it is valued. As we see these issues as tied together we are not going to focus only on usability, but also on the presentation of content and how the user perceived the augmented information presented to him. To provide a deep user insight we decided to conduct a qualitative user study using the question-asking protocol to interview the users. Our test participants were six German speaking students between 20 and 30 years. As the highest distribution of smartphones is in the ages between 25 and 34 years we focused on this age group to avoid problems regarding missing experience with mobile devices. [Pe11] Furthermore, the observations were recorded by video upon agreement with the participant, and observation notes were taken.

2.1 Representative Application Used for Our Study

To make the study as realistic as possible, we looked for a representative existing application that makes use of different Augmented Reality features in a professional way. A good and often used application field is within magazines and newspapers by augmenting their content in different ways. A considerable demonstration and representation of nowadays possibilities is the Augmented Reality campaign from the German *Stern* magazine (<http://www.stern.de/magazin/>) in August 2011. Their reader got the ability to look through their smartphone camera onto selected pages, where they saw, for example, 3D models, movies, diagrams and even a small game directly displayed upon the magazine. Some other pages acted like a trigger for different actions like opening websites or opening short videos. The *Stern* magazine is a well-established in Germany and has a circulation of about one million sold copies. As it is written in German, the participants needed to understand text and content. Besides the standalone app SternAR it was also possible to access the content by the AR Browser Junaio.

2.2 Test Scenario

In the beginning of the test we introduced us and our research to the participants. After a short explanation about Augmented Reality we asked some concrete question about their person, what experiences they already had with the technology and what they could imagine as possible application fields.

After this introduction into the study the participants received the task to try out the provided magazine on their own. To reduce the high barrier to entry we allocated the participants a smartphone with the preinstalled *Junaio* browser (version 3.1.0). It spared downloading and installing the app from the mobile app market, which was not needed for our research, and created a more realistic procedure in comparison to the standalone app. Furthermore, technical problems depending on different devices could be avoided in this way. If they still had questions they were allowed by the question-asking protocol to ask an assigned tutor. Once the application had been started they could browse the magazine however they pleased. After 10 minutes the participants were interrupted. Afterwards they were guided to look at six pages of different types offered by the magazine like a 3D model, a small game, a video overlay and triggering of a trailer or website. Besides, the user's expectations and valuations about each of these pages were asked.

For the last step they had to answer some final question about their opinion and impression about the application.

3 Results

During our test we observed several problems that users had with the AR browser, application and the corresponding magazine. To get the application running correctly it was necessary to read the instructions inside the magazine, which was easily overlooked or the participants would try to skip the instruction page. This would lead to failure in starting the application and getting to know how it runs. This shows one of the major problems of AR: Getting started does require reading instructions as the knowledge of usage is not well established yet and differs from application to application, which causes insecurity and is quite inconvenient. We had the impression that users rather want to use something they can get into quickly.

The next major problems were found in the magazine and how it was meant to guide the users through it. Besides the already mentioned problem with the instructions, there were misleading icons that could lead to frustration. Content shown as augmented was often perceived as slightly disappointing, as users would have expected more interactivity and interactive content with storytelling elements that scored the best user reactions.

The application itself still has some major technical deficits that can cause irritation among users. The interface of the browser lacks some user guidance, as some of the users were insecure in how to proceed in order to find the channels.

We assume that many of the existing problems are due to the young nature of Augmented Reality and missing experiences inside this field. As we mentioned in the introduction, especially in the field of usability not many studies have been conducted.

4 Conclusion and Thoughts on Improvement

Concluding from our tests and observations one of the most important issues to tackle is the problem of letting the user get easily into the application. In this case there was basically just one way and that wasn't convenient or comfortable for every user. Providing different ways of getting access and keeping in mind that especially tech-savvy readers may want to have another and simpler approach than looking for the instruction page first could be the first step. This could be done by having the QR code show a demonstration video on how to get started. For users who don't like to keep reading instructions a well-defined storyboard with short text could be one solution.

Content-wise big pictures proved to be a huge magnet of interest for readers who wished to know more about the picture he is looking at. The picture could tell a story by being simply animated or offering news and other informational content. Different types of content should be marked differently by striking icons that tell the user what he is going to see. So users could skip the page if it would be presented in a way he is not interested in, for example if he doesn't want to see a video.

There were also several user wishes that concern the interaction. Not having just a picture or a video but a menu that can lead to different informational levels that can be accessed depending on the user needs and thus proving to be really an augmented experience. It is necessary to establish a procedure for AR that is equal for every application, so that users do not need to keep reading instructions for new software they want to use. Augmented Reality browsers are a very good way to start with, as the starting procedure is always the same.

In our study we have seen that Augmented Reality can offer many further possibilities and has been positively accepted, although confronted with technical difficulties. Improvements in usability and user interaction could further improve acceptance of Augmented Reality and its huge potential for the end-user.

References

- [DGB01] Dünser, A., Grasset, R., & Billinghurst, M. (2008). A Survey of Evaluation Techniques Used in Augmented Reality Studies. ACM SIGGRAPH ASIA 2008 courses (p. 5:1-5:27). New York: ACM.
- [Bi08] Billinghurst, M. (2008). Usability Testing of Augmented / Mixed Reality Systems. ACM SIGGRAPH ASIA 2008 courses (p. 6:1—6:13). New York: ACM.
- [Pe11] Pew Research Center,
http://pewinternet.org/~media/Files/Reports/2011/PIP_Smartphones.pdf

Gestairboard: A gesture-based touch typing keyboard using the Kinect camera

Johannes Roith, Tayfur Coskun, Teodora Velikova, Sebastian Kligenbeck, Gudrun Klinker

Fakultät für Informatik
Technische Universität München
Boltzmannstraße 3
85748 Garching bei München

{roith, coskun, velikova, klingeb, klinker}@in.tum.de

Abstract: This paper describes a Kinect-based virtual keyboard using finger gestures. Existing systems usually involve expensive range sensors or custom-built hardware equipment. Furthermore, range sensor-based keyboards that are used to detect taps on imaginary keys suffer from shortcomings that were originally reported for touchscreen keyboards: There is a lack of tactile feedback, which has a negative effect on typing performance. The proposed *Gestairboard* extends the *Gestyboard* touchscreen keyboard concept that was suggested in 2011 to overcome exactly that problem, thus adapting it for devices without a touchscreen. It also uses inexpensive off-the-shelf hardware and to the best of our knowledge is the first implementation of a touch typing keyboard using Kinect. The prototype was compared to the *Gestyboard* and to a physical keyboard and evaluated in terms of performance, reliability, learnability and user acceptance.

1 Motivation

The *Gestairboard* originates from a novel gesture-based keyboard concept, known as the *Gestyboard* that was initially developed for use with multitouch input devices. Despite the success of virtual touchscreen keyboards in the mobile market, typing on them has proved to be slower and more error-prone when compared to physical keyboards. In previous work, we mentioned the lack of tactile feedback on conventional touchscreen keyboards as the main reason for their low performance [CAP⁺11] [LP98]. Additionally, any keyboard where users type on flat surfaces is likely to be affected as well. The stated goal of the *Gestyboard* - to overcome this problem while keeping the well-known 10-finger-system - was to be achieved through the use of finger gestures.

When using a conventional English hardware keyboard, users place their left hand fingers on the keys for the characters 'a', 's', 'd' and 'f' and their right hand fingers on the keys 'j', 'k', 'l' and ';', altogether called the *home row*. The thumbs are resting on the spacebar. In the *Gestyboard* concept each home row key defines a finger region. The surrounding keys, which are usually typed by the same finger, are assigned to the respective region. There are

two types of finger gestures: *Tap gestures* are used to activate the home row keys, whereas *sliding gestures* activate all other keys. Since each key can be triggered using a unique gesture, no tactile feedback is needed.

The proposed *Gestairboard* extends the concept of the *Gestyboard* to camera-based virtual keyboards where the user can type directly on a table while exploiting the familiar set of gestures. The Microsoft Kinect motion controller - an input device combining a VGA camera with a depth sensor - enables the detection of finger movements.

2 Related Work

Besides the *Gestyboard*, described above, there is an enormous amount of related literature concerning the use of computer vision in virtual keyboards and the implementation of finger tracking. A good overview of various virtual keyboards, including two computer vision-based touch typing keyboards, is provided in [KT02]. More recently, Huan Du et al. described a system that implements a virtual keyboard by projecting a pattern of colored light onto the users hands. It observes the distortion of the stripes to recover range information from the image captured by a calibrated conventional camera [DC08]. Another very recent virtual keyboard, suggested by Murase et al. [MMS⁺12], additionally uses finger gestures and operates with an ordinary VGA camera, but still requires manual calibration.

The various Kinect software development kits include tracking of body joints except for finger joints, making a custom implementation necessary. The most noteworthy finger tracking projects using Kinect include work by the MIT¹, the open source *Candescent NUI* library, and a method using a detailed 3D model of all finger joints [OKA11]. However, none of these implementations quite met our requirements.

3 Concept and Implementation

The *Gestairboard* was designed to use only a single Kinect camera, to work without a dictionary or word prediction and to enable blind typing at an acceptable speed and error rate. Our prototype is implemented as a C# library that receives a stream of depth images from the Kinect camera, which are subsequently analyzed to detect relevant features. As part of the feature detection module we implemented our own components to detect outlines and orientation of hands, the center of each palm, fingertips (based on [DC08]) and hand movements and to identify and track individual fingers. Layered on top is a module for gesture detection, which, in turn, forms the foundation for a component that maintains the state of the keyboard. The *Gestairboard* shows a simple keyboard on the screen with the current fingertip positions depicted by green circles and additionally displays the computer vision data. Detected keys are sent to the active application to simulate a keystroke.

¹<http://www.ros.org/wiki/mit-ros-pkg>

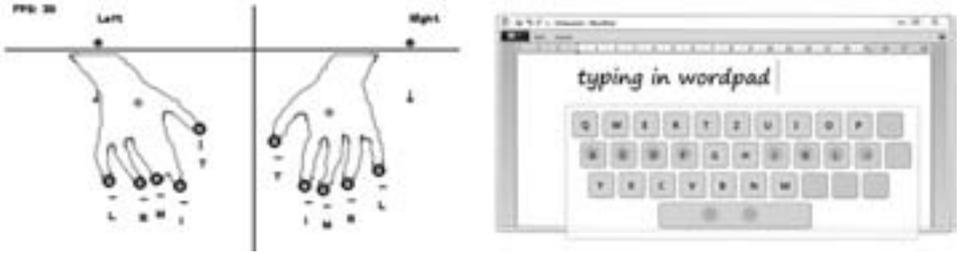


Figure 1: The Gestairboard UI: visualization of detected features (left) and the virtual keyboard overlay (right), which is shown close to the bottom of the display while the keyboard is being used.

4 Evaluation

The prototype was tested with seven participants knowing the 10-finger-system but lacking prior experience with both Gestairboard and Gestyboard. The evaluation included a field-study, an individual interview and a questionnaire. Users had to type 17 English sentences on the Gestairboard, the Gestyboard and a conventional keyboard using the free typing tutor *TIPP10*. Two metrics were used: the error rate (fraction of characters wrongly typed at the first attempt) and the typing speed in characters per minute (*cpm*). The questionnaire included the System Usability Scale (SUS)², the NASA Task Load Index (NASA-TLX)³ and a part with custom questions per tested device. Along with the interviews they enabled us to analyze preferred keys, difficult key sequences, preferred sensitivity of gesture activation, comfort of hand orientation, learnability and mental and physical demand.

5 Results

While the physical keyboard had a significantly lower error rate, we found no significant difference between the error rate of the Gestairboard and the Gestyboard, possibly due to the small number of participants. Tap gestures had a significantly lower error rate than sliding gestures. About 1/3 of all Gestairboard errors were due to unintended spacebar activation. The physical keyboard turned out to be significantly faster than both virtual keyboards (median 184.14*cpm*). Although generally the Gestyboard (median of 27.5*cpm*) appears to be somewhat faster than the Gestairboard (23.99*cpm*), the difference was not statistically significant. This may be due to the sample size. We also found no correlation between typing speed and error rate for either the physical keyboard or the Gestairboard, but could confirm the correlation for the Gestyboard as described in [CAP⁺11].

The Gestairboard obtained a SUS score of 69.9. The NASA-TLX indicated that both tasks were moderately demanding. The Gestairboard has a lower perceived task load with an average score that is just 75% of the Gestyboard score. Temporal demands and

²The SUS is a widely used tool to determine the subjective opinions of the system usability;

³The NASA-TLX is a very popular multi-dimensional scale to determine perceived workload of a task.

frustration were almost twice as high for the Gestyboard prototype. Participants indicated in the Gestairboard questionnaires that they preferred the home row keys. Some keys that required a forward-sliding gesture were disliked, especially if it had to be performed by a ring finger. Some also had problems to lift their ring fingers enough for the system to be able to recognize a tap gesture. Users generally found the Gestairboard pleasant to use and reported little or no frustration. They stressed the importance of better visualization.

Analysis of the recorded computer vision videos (see Figure 1, left) showed that the participants manual dexterity often affected overall performance more than the reliability of the feature detection.

6 Conclusion and Future Work

We have found that the gesture concept works quite well with Kinect, although some users encountered difficulties due to limitations of the human hand anatomy. Most users liked the experience of typing on the Gestairboard. Despite a comparatively high error rate, the evaluation did not indicate that the Gestairboard is significantly less reliable than the Gestyboard. Typing speed is similar to the one of the Gestyboard, although considerably slower than on a physical keyboard. Blind typing is possible as well. It might be worth to reconsider some Gestyboard principles to be able to better adapt the Gestyboard concept to the Kinect hardware. More work is needed to increase reliability, for example by providing better visual feedback. Other future research could include evaluating a modified prototype that features auto-correction or using multiple Kinect cameras to reduce occlusions.

References

- [CAP⁺11] Tayfur Coskun, Eva Artinger, Lorenzo Pirritano, Daniela Korhammer, Amal Benzina, Claudia Grill, Andreas Dippon, and Gudrun Klinker. Gestyboard v1.0: A 10-finger-system and gesture based text input system for multi-touchscreens with no need for tactile feedback. Technical Report 342/22/11 I, 2011.
- [DC08] Huan Du and Edoardo Charbon. A virtual keyboard system based on Multi-Level Feature Matching. *Conference on Human System Interactions, 2008*, 2008.
- [KT02] Mathias Kölsch and Matthew Turk. Keyboards without Keyboards : A Survey of Virtual Keyboards. In *Human Performance, 2002*.
- [LP98] K.T Landauer and P.V. Prabhu. *Handbook of Human-Computer Interaction*. Elsevier Science Inc., New York, New York, USA, 1998.
- [MMS⁺12] Taichi Murase, Atsunori Moteki, Genta Suzuki, Takahiro Nakai, Nobuyuki Hara, and Takahiro Matsuda. Gesture keyboard with a machine learning requiring only one camera. *Proceedings of the 3rd Augmented Human International Conference on - AH '12*, pages 1–2, 2012.
- [OKA11] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. *Proceedings of the British Machine Vision Conference 2011*, pages 101.1–101.11, 2011.

Analysieren und Bewerten der Einsatzmöglichkeiten des EPOC Neuroheadsets von Emotiv

Nicolas Simon

Hochschule Bonn-Rhein-Sieg
IVC
nicolas.simon@smail.inf.h-brs.de

Art der Arbeit: Studienarbeit im Masterstudiengang Computer Science
Betreuer/in der Arbeit: Prof. Dr. Wolfgang Heiden, Dr. Gernot Heisenberg

Abstract: In dieser Arbeit soll mit einem Neuroheadset der Firma Emotiv eine Bildergalerie auf einem Android-Smartphone über Gedanken gesteuert werden. Dies soll in einem ersten Ansatz über motorische Imagination geschehen. Dabei wird die Bewegung der linken oder rechten Hand vorgestellt. In einem weiteren Ansatz wird die dem Neuroheadset mitgelieferte Software und die API von Emotiv verwendet und mit dem ersten Ansatz verglichen.

1 Das Experiment

Im Bereich Brain-Computer Interfaces (BCI) wird die Gehirnaktivität mit spezieller Hardware abgegriffen, um diverse Anwendungsszenarien zu ermöglichen [KM07]. Hier soll über Gedanken eine Bildergalerie gesteuert werden, wofür drei Bereiche zu unterscheiden sind. Veranschaulicht werden diese in Abbildung 1. Es handelt sich dabei um

1. Den Client: Das Android-Smartphone ist über WLAN mit dem Server verbunden und empfängt über TCP/IP die Kommandos, um in der Bildergalerie ein Bild nach links oder nach rechts weiterzuschalten.
2. Der Server: Hier findet die Verarbeitung der vom Neuroheadset empfangenen EEG-Signale statt. Nach der Auswertung werden die entsprechenden Kommandos an den Client gesendet.
3. Das Neuroheadset: Das EPOC Neuroheadset ist ein 14+2 Kanal EEG. 14 Elektroden, die sich an unterschiedlichen Positionen auf dem Kopf befinden, liefern jeweils mit einer Sampling Rate von 128 Hz ihre Werte, die gegen zwei weitere Elektroden referenziert werden. Über Bluetooth gelangen die EEG-Signale dann zu dem Server [RES13].

Der Client ist durch eine beliebige Anwendung ersetzbar. So ist es z.B. auch vorstellbar, einen Roboter nach links oder rechts zu steuern.

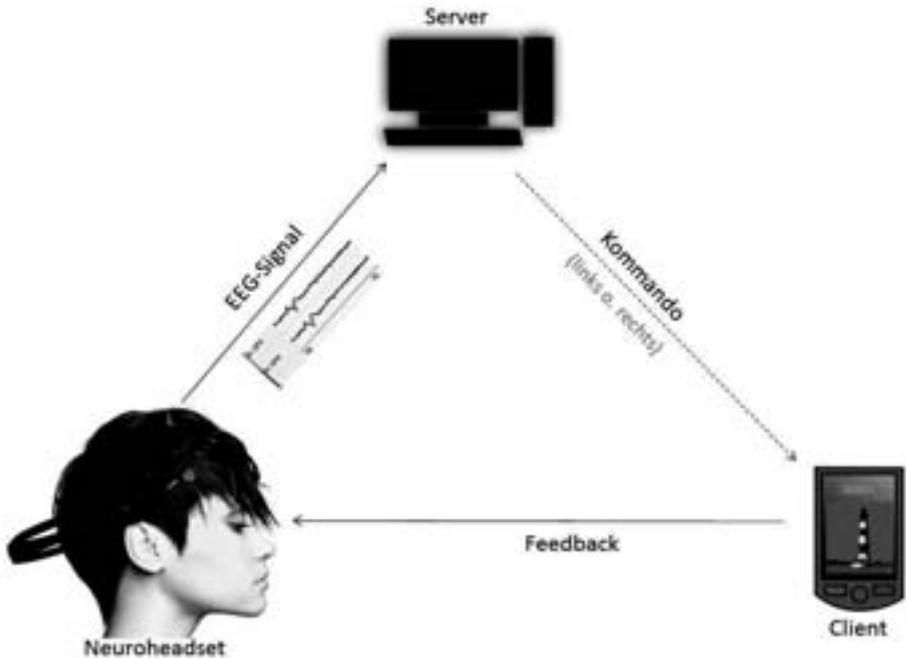


Abbildung 1: Das Projekt

2 Die zwei Ansätze

Für die Analyse der EEG-Signale wird zunächst die Software OpenViBE von Inria eingesetzt und in einem zweiten Ansatz die Emotiv Suite mit der Emotiv API benutzt. Das generelle Vorgehen ist dabei gleich. In einem ersten Schritt müssen Trainingsdaten erhoben werden. Diese durchlaufen eine Filterung, um z.B. Störungen zu eliminieren. Als nächstes werden Features, also eindeutige Merkmale im Signal berechnet, mit denen der Klassifizierer trainiert wird. Nach ausreichendem Training kann die Anwendung eingesetzt werden.

2.1 OpenViBE

OpenViBE ist eine BCI Software die eine Vielzahl von Neuroheadsets unterstützt. Es werden Szenarien erstellt, wobei per Drag & Drop sogenannte Boxen in einem Arbeitsfeld platziert werden. Eine Box implementiert eine Funktion, wie z.B. einen Bandpass Filter, oder ein Fenster, wo die EEG-Signale visuell dargestellt werden. Boxen sind nicht voneinander isoliert, sondern werden über Ports miteinander verbunden. So kann ein modifiziertes Signal zur nächsten Box gesendet werden oder ein Signal empfangen werden, um es dann zu verarbeiten [Op13]. Für diese Anwendung wurden

Szenarien zur Erhebung der Trainingsdaten erstellt, bei dem ein Pfeil nach links den Probanden anweist, sich eine Bewegung der linken Hand vorzustellen und ein Pfeil nach rechts die Vorstellung der rechten Handbewegung anweist. Ein weiteres Szenario behandelt das Trainieren des Klassifizierers und ein Letztes ist die Online-Anwendung selbst, wo bei erkannter imaginerter links- oder rechts-Bewegung das entsprechende Kommando an das Smartphone gesendet wird. Alle Szenarien orientieren sich an den Ergebnissen von Pfurtscheller et al., die in [PNF97] beschrieben sind.

2.2 Emotiv Suite

In der Emotiv Suite werden Kommandos mittels eines Würfels in einem Raum trainiert. Wenn der Proband z.B. seinen Gedanken für links ausführt und der Würfel sich in diese Richtung bewegt, werden von der Emotiv Suite Events mit einer Id für links generiert. Diese Events werden von einer eigenen Anwendung, welche die Emotiv API benutzt, empfangen, um dann die entsprechenden Kommandos an das Smartphone zu senden, in diesem Fall also das Kommando für links [Em13].

3 Die Ergebnisse

Bei dem Versuch mit OpenViBE konnten nur ca. 53% - 57% der Features einer Klasse für links oder rechts zugeordnet werden. Dies ist für eine Erkennung zu wenig. Eine Erklärung besteht darin, dass das Neuroheadset von Emotiv nur 14 Elektroden besitzt und damit für die Erkennung der motorischen Imagination wichtige Positionen auf dem Kopf nicht abdecken kann. Im Vergleich dazu stand Pfurtscheller ein 64-Elektroden Neuroheadset zur Verfügung [PNF97].

Im zweiten Versuch, mit Hilfe der Emotiv Suite, wurde der Würfel zuerst ebenfalls über motorische Imagination gesteuert. Eine Unterscheidung zwischen links und rechts ist jedoch nicht möglich gewesen. Als nächstes wurde ein sich drehendes Rad auf der linken Seite, wie auch auf der rechten Seite, vorgestellt. Die Fehlerquote lag hier bei ca. 46%, d.h. ungefähr die Hälfte der kognitiven Aktionen entsprachen nicht dem gewünschten Steuerbefehl für die Bildergalerie.

In einem weiteren Versuch wurde für das Kommando „left“ das Wegdrücken der Hände vorgestellt und für das Kommando „right“ das nach oben Drücken der Hände. Außerdem wurden die Augen bei Ersterem geschlossen, sodass sich die Gehirnaktivität mehr in den Alpha-Bereich zwischen 8 und 12 Hz verlagerte. So konnte eine Fehlerquote von 20% erreicht werden, da sich die Muster im Signal eindeutiger unterschieden.

Somit ist die Steuerung einer Bildergalerie möglich gewesen, allerdings fällt jedes falsch bewegte Bild sofort negativ auf. Bei anderen Anwendungen wie z.B. dem Roboter wäre eine kurze Linksbewegung anstatt nach rechts weniger auffällig. Da in einem EEG auch die Kontraktion von verschiedenen Gesichtsmuskeln unterscheidbar sind, soll in einem nächsten Ansatz diese für die Steuerung einer Benutzeroberfläche genommen werden.

Literaturverzeichnis

- [Em13] Emotiv User Manual, <http://www.emotiv.com/developer/SDK/UserManual.pdf>, 2013.
- [KM07] Kübler, A.; Müller K.-R.: An Introduction to Brain-Computer Interfacing. In: Toward Brain-Computer Interfacing, MIT Press, Cambridge, Mass., 2007.
- [Op13] OpenViBE: User Documentation, <http://openvibe.inria.fr/documentation-index/#User+Documentation>, 2013.
- [PNF97] Pfurtscheller, G.; Neuper, C.; Flotzinger, D.; Pregenzer, M.: EEG-based discrimination between imagination of right and left hand movement. *Electroencephalography and Clinical Neurophysiology* Volume 103, Issue 6, Dec 1997.
- [RES13] Emotiv SDK Research Edition, www.emotiv.com/upload/manual/sdk/Research%20Edition%20SDK.pdf, 2013.

Ein Fehlertoleranzkonzept für dynamische, verteilte OSGi-Anwendungen

Benjamin Vogler, Jan Schäfer

(benjamin.vogler@student.hs-rm.de, jan.schaefer@hs-rm.de)

Hochschule RheinMain, Fachbereich Design Informatik Medien

Labor für Verteilte Systeme, <http://wwwvs.cs.hs-rm.de>

1 Motivation

Die steigende Menge an Hardware und Software im persönlichen Lebensumfeld ermöglicht es Nutzern, immer mehr IT-Dienste sowohl in den eigenen vier Wänden als auch über das Internet zu nutzen. PCs, Smartphones und Tablets, Medienabspielgeräte und intelligente Heimautomatisierungsgeräte (aus dem Bereich *Smart Home*) werden dadurch immer enger miteinander und mit externen Internetdiensten (z.B. Cloud-Datenspeichern) verflochten. Gemeinsam mit den u.a. durch das Forschungsfeld *Ambient Assisted Living* (AAL) [ADF⁺08] entwickelten Assistenzsystemen entsteht eine Komplexität, die gewöhnliche Nutzer nicht mehr selbständig administrieren können oder wollen.

Die komponentenbasierte *OSGi Dienstplattform* [OSG11] hat in den letzten Jahren auch im AAL-Bereich eine große Verbreitung erfahren. Obwohl bereits einige AAL-Plattformen wie *universAAL* [HMH⁺11] oder *SOPRANO* [SWKB09] für den Einsatz im persönlichen Lebensumfeld entwickelt wurden, spielt der Aspekt Fehlertoleranz bislang keine Rolle. In der Regel wird sogar nur eine zentrale Komponente zur Ausführung von Anwendungen vorgesehen, deren Ausfall das ganze System funktionslos machen würde. Genau hier setzt das in diesem Beitrag vorgestellte Konzept an.

Statt einer zentralisierten Ausführung von Diensten sieht diese Arbeit vor, dass Dienste gemäß ihrer Fehlertoleranzanforderungen dynamisch transparent im Netzwerk verteilt werden können und dabei insbesondere Knoten eines Heimnetzwerks nutzen. Für OSGi existieren bereits Ansätze zur Erreichung von Fehlertoleranz, die jedoch entweder nur die lokale Replikation auf einer physischen Maschine unterstützen [KTA09], Modifikationen der OSGi-Laufzeitumgebung benötigen [AOS06] oder für Dienstanutzer nicht transparent sind und immer eine Anwendungsanpassung erfordern [TaCR09].

2 Konzept

Ausgangspunkt des gewählten Fehlertoleranzansatzes ist die Anforderung, dass aus Sicht eines Dienstanutzers die Aufrufe von normalen und von fehlertoleranten Diensten nicht unter-

schieden werden müssen. Diese Transparenz wird über einen vorgeschalteten Proxy-Dienst ermöglicht, der den ursprünglichen (lokalen) Dienst ersetzt und die zugrunde liegenden Fehlertoleranzmechanismen in der Regel vor dem Dienstanutzer verbirgt.

Fehlertoleranz beruht grundsätzlich auf redundanten Strukturen [PSL80], d.h. in diesem Fall auf verteilten Dienstreplikaten, die auf lokalen und entfernten OSGi-Frameworkinstanzen (*Knoten*) ausgeführt werden. Dabei wird ein konfigurierbares N-aus-M System mit aktiver und passiver Redundanz je Dienst möglich. Passive Replikate, die bei Ausfall eines aktiven Knotens aktiviert werden, erhalten ansonsten ausschließlich Zustandssynchronisationen, wofür Dienstimplementierungen um einen expliziten Synchronisationsaufruf erweitert werden müssen. Durch diesen Ansatz werden transiente und permanente Ausfälle von physischen Servern, OSGi-Frameworkinstanzen und einzelnen Dienstreplikaten toleriert.

Um Konfigurationsdaten austauschen sowie ausgefallene Replikate erkennen und sicher ersetzen zu können, muss auf allen Knoten eine konsistente Sicht auf das System vorhanden sein, was hier durch die Nutzung eines Gruppenkommunikationsframeworks umgesetzt wird. Darüber hinaus wird die Installation und Deinstallation fehlertoleranter Dienste zur Laufzeit über eine in der OSGi-Spezifikation definierte Konfigurationsschnittstelle erreicht. Weitere Knoten können dem Verbund bei Bedarf ebenfalls dynamisch hinzugefügt werden.

Der hier gewählte Ansatz für fehlertolerante OSGi-Dienste kann in jede existierende OSGi-basierte Dienstplattform integriert werden und wird somit breit einsetzbar. Um beliebige standardkonforme OSGi-Implementierungen verwenden zu können, wurden daher keine Änderungen am OSGi-Framework vorgenommen. Alle Mechanismen beruhen entweder auf eigenständigen Komponenten oder werden selbst als OSGi-Dienste implementiert.

3 Prototypische Implementierung

Grundlage der Implementierung bildet das *Eclipse Equinox* [Ecl12b] OSGi Framework. Neben den über OSGi-Dienste realisierten Komponenten wurde die verteilte Konfiguration über das Gruppenkommunikationsframework *Apache ZooKeeper* [Apa12] umgesetzt. Die Ortstransparenz der verteilten Dienstreplikate basiert auf der OSGi *Remote Service Admin* Implementierung des *Eclipse Communications Frameworks* (ECF) [Ecl12a]. Lokal werden die in der OSGi-Plattform-Spezifikation [OSG11] definierten Standarddienste *Config Admin Service* für die dynamische Konfiguration zur Laufzeit und *Event Admin Service* [WHKL08] zur lokalen Kommunikation zwischen Diensten genutzt. Eine Übersicht über die entstandene Grobarchitektur ist in Abbildung 1 dargestellt.

Die **Installation** fehlertoleranter Dienste wird über den *Config Admin Service* vorgenommen. Der *Configurator* reagiert auf Konfigurationsänderungen und leitet die Installation und Konfiguration aller anderen Komponenten ein. Ein über den *ZooKeeper Server* ausgewählter Masterknoten nimmt die verteilte Installation vor. Der *Event Dispatcher* erstellt einen *FTPProxy* (Fehlertoleranzproxy), der die vom *Distribution Manager* erstellten aktiven und passiven Replikate auf lokalen und entfernten OSGi-Knoten versteckt.

Zur **Ausführungszeit** ersetzt der *FTPProxy* den ursprünglichen (nicht fehlertoleranten) Dienst und leitet Dienstaufrufe an die aktiven Replikate weiter. Die Zustandssynchronisation

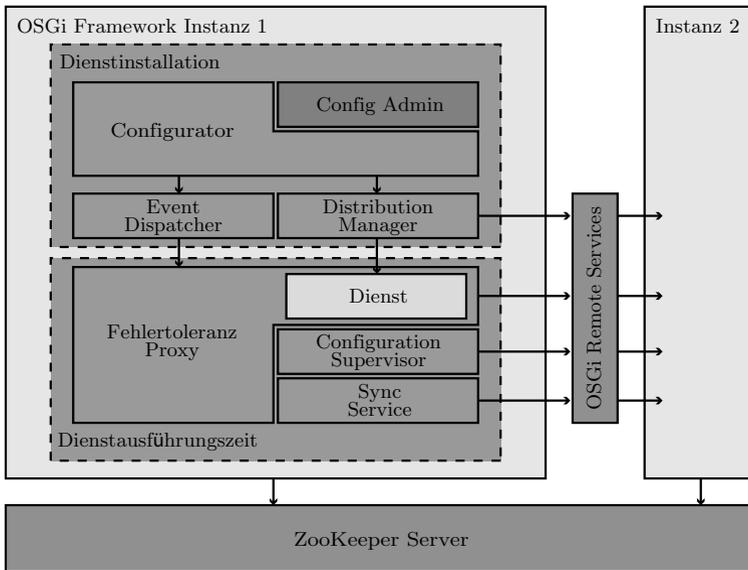


Abbildung 1: Grobarchitektur des Fehlertoleranzkonzepts

zwischen Replikaten wird vom *Sync Service* vorgenommen. Ausfälle von Replikaten werden vom *Configuration Supervisor* kompensiert. Da alle Komponenten über Einträge im *ZooKeeper Server* konfiguriert werden, ist über dessen Verzeichnisstruktur jederzeit eine konsistente, auf allen Knoten präsente Sicht auf das System verfügbar.

4 Stand der Arbeiten

Die Implementierung ist weitgehend abgeschlossen und die Evaluation hat begonnen. Wie erwartet zeigt sich für fehlertolerante Dienste eine erhöhte Antwortzeit, die auf die Verteiltheit und notwendige Koordination der Dienstreplikate zurückzuführen ist. Eine quantitative Performance Evaluation wird derzeit vorbereitet.

In funktionaler Hinsicht kann eine Gruppe von *ZooKeeper*-Servern derzeit noch nicht dynamisch verändert werden, was der hier notwendigen dynamischen Konfigurierbarkeit entgegensteht. Diese Funktionalität ist jedoch für einen kommenden Release eingeplant.

Die hier vorgestellte Arbeit basiert auf der Master Thesis von Patrick Deuster [Deu11]. Das dort entwickelte Konzept wurde im Rahmen eines durch die hochschulinterne Forschungsförderung geförderten Projekts unter der Leitung von Jan Schäfer durch Benjamin Vogler um die dynamische Konfiguration zur Laufzeit erweitert. Im Rahmen der Dissertation von Jan Schäfer und des BMBF-geförderten Projekts *WieDAS* (Förderkennzeichen 17040B10) soll der Ansatz zur Demonstration eines OSGi-basierten Selbstmanagementansatzes genutzt werden, da der Hochverfügbarkeit hier eine große Bedeutung zukommt.

Literatur

- [ADF⁺08] Sahin Albayrak, Eva Susanne Dietrich, Frerich Frerichs, Erhard Hackler, Stefan Jhnicen, Bernd Kriegesmann, Lothar Litz, Peter Oberender, Eduard Sailer, Annette Spellerberg, Elisabeth Steinhagen-Thiessen und Werner Vogt. *Intelligente Assistenz-Systeme: Im Dienst für eine reife Gesellschaft*. VDE-Positionspapier. Deutsche Gesellschaft für Biomedizinische Technik (DGBMT), Frankfurt am Main, 2008.
- [AOS06] Heejune Ahn, Hyukjun Oh und Chang Oan Sung. Towards reliable OSGi framework and applications. In *Proceedings of the 2006 ACM symposium on Applied computing, SAC '06*, Seiten 1456–1461, New York, NY, USA, 2006. ACM.
- [Apa12] Apache Software Foundation. *Apache Zookeeper*, November 2012.
- [Deu11] Patrick Deuster. Ein Fehlertoleranzkonzept für verteilte OSGi-Anwendungen. Diplomarbeit, Hochschule RheinMain, FB Design Informatik Medien, February 2011.
- [Ecl12a] Eclipse Foundation. *Eclipse Communication Framework*, Juni 2012.
- [Ecl12b] Eclipse Foundation. *Eclipse Equinox*, September 2012.
- [HMH⁺11] Sten Hanke, Christopher Mayer, Oliver Hoeflberger, Henriette Boos, Reiner Wichert, Mohammed-R Tazari, Peter Wolf und Francesco Furfari. universAAL - An Open and Consolidated AAL Platform. In Reiner Wichert und Birgid Eberhardt, Hrsg., *Ambient Assisted Living*, Kapitel 10, Seiten 127–140. Springer Berlin Heidelberg, 2011.
- [KTA09] Young-Woo Kwon, Eli Tilevich und Taweessup Apiwattanapong. DR-OSGi: hardening distributed components with network volatility resiliency. In *Proceedings of the ACM/IFIP/USENIX 10th international conference on Middleware, Middleware'09*, Seiten 373–392, Berlin, Heidelberg, 2009. Springer-Verlag.
- [OSG11] OSGi Alliance. *OSGi Service Platform Release 4 (Version 4.3) Core Specification*, April 2011.
- [PSL80] M. Pease, R. Shostak und L. Lamport. Reaching Agreement in the Presence of Faults. *J. ACM*, 27(2):228–234, April 1980.
- [SWKB09] Andreas Schmidt, Peter Wolf, Michael Klein und Dirk Balfanz. SOPRANO Ambient Middleware: Eine offene, flexible und markt-orientierte semantische Dienstplattform für Ambient Assisted Living. In 2. *Deutscher Ambient Assisted Living Kongress*. VDE Verlag, 2009.
- [TaCR09] Carlos Torrão, Nuno A. Carvalho und Luís Rodrigues. FT-OSGi: Fault Tolerant Extensions to the OSGi Service Platform. In *Proceedings of the Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009 on On the Move to Meaningful Internet Systems: Part I, OTM '09*, Seiten 653–670, Berlin, Heidelberg, 2009. Springer-Verlag.
- [WHKL08] G. Wütherich, N. Hartmann, B. Kolb und M. Lübken. *Die OSGi Service Platform: Eine Einführung mit Eclipse Equinox*, Seite 227. Dpunkt.Verlag GmbH, 2008.

ORESTES: ein System für horizontal skalierbaren Zugriff auf Cloud-Datenbanken

Felix Gessert, Florian Bücklers

Universität Hamburg, Fachbereich Informatik
gessert@informatik.uni-hamburg.de, fbuecklers@gmail.com

Art der Arbeit: Bachelorarbeit und Masterarbeit

Betreuer der Arbeit: Prof. Dr. Norbert Ritter

Abstract: Die Nutzbarkeit von Datenbanksystemen in Cloudumgebungen ist derzeit durch drei Probleme stark eingeschränkt: I) fehlende Mechanismen zur elastischen horizontalen Skalierung, II) hohe Netzwerklatenzen und III) Uneinheitlichkeit der Zugriffsprotokolle und -Schnittstellen. Um Database-as-a-Service Modelle attraktiv und umsetzbar zu machen, müssen diese Probleme gelöst werden. Wir stellen deshalb ORESTES (Objects RESTfully Encapsulated in Standard formats) vor. ORESTES ist ein auf REST/HTTP basierendes System für Datenbankzugriff, das in der Lage ist, Lese-Anfragen horizontal zu skalieren. Die Skalierbarkeit wird durch Web-Infrastruktur aus Web-Caches und Load-Balancern erzielt, unter Erhaltung der ACID-Transaktionssemantik durch optimistische Validierung (I). ORESTES gestattet es, Web-Caches in beliebigen Topologien anzuordnen. Das Latenzproblem kann deshalb durch Web-Caches gelöst werden, die geographisch dicht am Client positioniert sind (II). Dies kann u.a. durch Web-Caches im Netzwerk des Clients oder die Nutzung verbreiteter Content Delivery Networks (CDNs) geschehen. Die REST/HTTP-Schnittstelle ist so konzipiert, dass beliebige Datenmodelle und Kombinationen aus Persistenz-APIs und Datenbanksystemen möglich sind (III). Die vergleichende Evaluation unserer Implementierung zeigt, dass ORESTES gegenüber existierenden Systemen eine signifikante Verbesserung der Latenz, Skalierbarkeit, Elastizität, Erweiterbarkeit und Schnittstellenoffenheit erreicht.

1 Motivation und das System ORESTES

Die Wissenschaft und Praxis von Datenbanksystemen befindet sich in einem der bisher radikalsten Umbrüche ihrer Geschichte. Drei Paradigmenwechsel sind eindeutig identifizierbar: der zunehmende Wechsel von dem zuvor uneingeschränkt dominierenden relationalen Datenmodell zu domänenspezifischen, anforderungsgetriebenen Datenmodellen und Datenbanken („NoSQL“, „Polyglot Persistence“), die Ausrichtung vieler Datenbanken auf Technologien des Webs und seine Größenordnungen („Horizontal Scalability“, „Big Data“) und die häufige Verschiebung von Datenverarbeitung und – Speicherung auf Cloud Computing Plattformen (z.B. Amazon Web Services und Windows Azure) [CJP+11]. Um diesem Paradigmenwechsel technisch angemessen zu begegnen, sind viele Neuerungen notwendig. Wir analysieren die gewandelten Anforderungen und stellen das System ORESTES vor, das sie auf Basis existierender Technologien (Web-Infrastruktur und Datenbanksysteme) umsetzt.

Der Erfolg neuer NoSQL-Systeme erklärt sich aus zwei Anforderungen – horizontaler Skalierbarkeit und einfachen Schnittstellen (cf. [Cat11]). Die Fähigkeit zur horizontalen Skalierbarkeit ermöglicht es, Datenbanksysteme auch für Szenarien mit besonders großem Datenvolumen und massive Workloads von hoher Parallelität einzusetzen. Die Beliebtheit alternativer Datenmodelle und einfacher Schnittstellen erklärt sich aus der erhöhten Produktivität der Anwendungsentwicklung und dem Wegfall des *Impedance Mismatch* [INW+09]. Ein domänenspezifisches Datenbanksystem sollte sich für den gemeinsamen Einsatz mit anderen Datenbanksystemen eignen (*Polyglot Persistence*). Viele Anwendungen in Cloudumgebungen kommunizieren mit genutzten Services (zu denen Persistenz zählt) über statusloses HTTP. Die Unterstützung von Web-Standards und -Protokollen wird deshalb zunehmend zu einer zentralen Anforderung für Datenbanksysteme. Datenbanken, die als Service angeboten werden, benötigen die Möglichkeit, Leseanfragen zu skalieren. Außerdem müssen sie die geographische Verteilung von Datenbank und Applikation durch die Gewährleistung einer geringen Netzwerklatenz ermöglichen. Eine geringe Latenz ist die unabdingbare Voraussetzung, um hohen Durchsatz und die Reaktivität eines lokalen Datenbank-Deployments zu erreichen. Häufig wird mit einem Verweis auf das CAP-Theorem (*Consistency, Availability, Partition Tolerance*) [GL02] ein Fehlen von Transaktionsmechanismen und starker Konsistenz entschuldigt. Wir sind jedoch der Überzeugung, dass Transaktionen eines der wichtigsten Ergebnisse der gesamten Datenbankforschung sind und als Option zur Verfügung stehen sollten. Die Anforderungen an ORESTES sind deshalb:

1. Elastische horizontale Skalierbarkeit
2. Erweiterbare und intuitive Datenbankschnittstelle
3. Umsetzung von Polyglot Persistence
4. Nutzung von Web Standards
5. Webbasiertes Monitoring, Deployment und Administrieren
6. Zugriff mit niedriger Netzwerklatenz
7. ACID-Transaktionen und starke Konsistenz als Option

Diese Anforderungen erfüllt ORESTES durch die in Abbildung 1 dargestellte Architektur. Persistenz-APIs (z.B. die Java Persistence API) bilden ihre Operationen auf die generische REST/HTTP Schnittstelle von ORESTES ab. Auf Netzwerkebene können durch die statuslose und cache-optimierte Natur der Kommunikation Web-Caching und Load-Balancing genutzt werden. Auf Serverseite bildet ein ORESTES-Server die REST-Aufrufe auf ein zugrundeliegendes Datenbanksystem ab.



Abbildung 1: Architektur von ORESTES

Die Web-Cache Server agieren als asynchron aktualisierte Replikate des Datenbanksystems (*asynchrone Master-Slave Replikation*). Durch Load-Balancing kann die Anfrage-Last horizontal auf mehrere ORESTES-Server und Web-Caches skaliert werden. In der Nutzbarkeit von Web-Caching und Load-Balancing unterscheidet sich ORESTES fundamental von anderen Cloud Data Management, NoSQL- und DBaaS-Systemen mit REST-Schnittstellen, die beide Mechanismen nicht unterstützen. Dies liegt in der Schwierigkeit begründet, das Caching-Modell von HTTP mit jenem eines Datenbanksystems in Einklang zu bringen. Ressourcen (also z.B. Datenbankobjekte) werden in HTTP für eine fest angegebene Caching-Dauer als aktuell betrachtet. Ad-hoc Invalidierungen sind nur in wenigen Spezialfällen möglich. Um dennoch Konsistenz zu gewährleisten nutzt ORESTES zwei Mechanismen: optimistische Validierung und bloomfilterbasierte Cache-Kohärenz. Während ersteres ein wohlbekanntes Konzept der Datenbankforschung ist, handelt es sich bei zweiterem, um einen neuen Algorithmus, den wir zur Nutzung von Caches einführen, die auf Expiration basieren. Die optimistische Validierung basiert darauf, dass die Persistenz-API des Clients die Versionsnummern aller gelesenen Objekte als Read-Set hält und beim Transaktionscommit an den Server überträgt. Der Server kann so validieren, dass keine veralteten Objekte gelesen wurden und keine Konflikte zu nebenläufigen Transaktionen aufgetreten sind (*Backward-Oriented Optimistic Concurrency Control*).

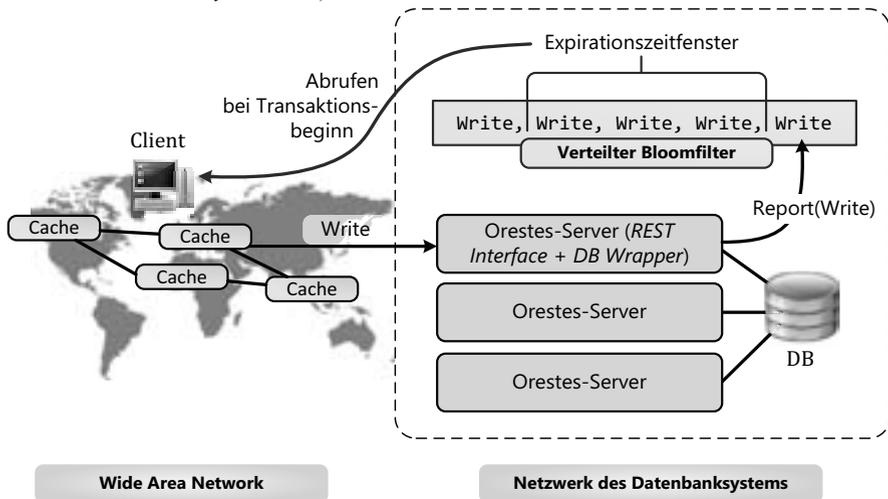


Abbildung 2: Bloomfilterbasierte Cache-Kohärenz in ORESTES

Die von uns eingeführte bloomfilterbasierte Cache-Kohärenz ist ein Algorithmus, der die Cache-Kohärenz von Web-Caches, die eigentlich nach dem Prinzip *Eventual Consistency* arbeiten, zu einem anpassbaren Parameter macht. Wie in Abbildung 2 dargestellt, werden dazu Schreibvorgänge in einem (verteilten) Counting Bloomfilter registriert, einer probabilistischen, sehr kompakten Mengenrepräsentation. Der Bloomfilter enthält für die Periode einer Caching-Dauer alle geänderten Objekte. Bei Transaktionsbeginn wird er optimiert zum Client übertragen, der anschließend für alle Objekte durch einen $O(1)$ Hash-Lookup feststellen kann, ob sie potentiell veraltet sind (d.h. im Bloomfilter enthalten sind) und direkt vom Server geladen werden müssen (*Revalidierung*).

2 Evaluation und Ausblick

Unsere ORESTES-Implementierung unterstützt derzeit zwei Persistenz-APIs: Java Data Objects (JDO) und eine von uns entworfene Portierung der Java Persistence API auf JavaScript (JSPA). Als serverseitige Datenbankbackends werden die beiden objektorientierten Datenbanken Versant Object Database (VOD) und db4o und sowie der NoSQL Key-Value Store Redis unterstützt. Das generische ORESTES-Framework, auf dem die Anbindungen basieren, stellt Funktionen wie Transaktionsverwaltung und ein umfassendes Browser-Interface mit integrierter Entwicklungsumgebung bereit. Abbildung 3 zeigt die in einer Cloudumgebung ermittelten Performancevorteile von ORESTES anhand eines Vergleichs von VOD mit seiner nativen TCP-Schnittstelle und VOD als Backend von ORESTES. Der Benchmark ermittelt die durchschnittliche Ausführungsdauer von 50 dedizierten Clientsystemen unter Operationen wie Lesen, Schreiben und Querying.

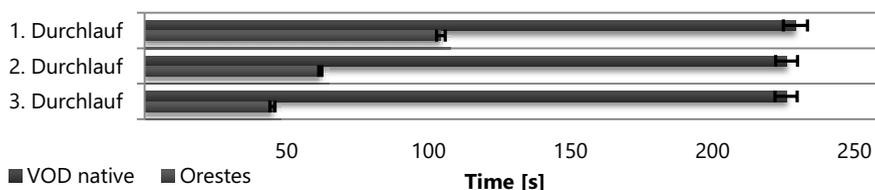


Abbildung 3: Ergebnisse der Evaluation von ORESTES in einer Cloudumgebung

Wir wollen unseren Ansatz künftig in viele Richtungen erweitern, vor allem bezüglich der Nutzung als skalierbaren Cloud Service. Die automatische Allokation von Ressourcen wie Caching-Servern und Sicherstellung von Quality-of-Service Garantien z.B. für die Quantile der Latenzzeiten sind dabei wichtige Punkte. Durch eine Erweiterung von ORESTES auf andere (NoSQL-) Datenmodelle hoffen wir, eine gemeinsame Basis für die sehr unterschiedlichen Klassen von Datenbanksystemen schaffen zu können.

Literaturverzeichnis

- [Fie00] Fielding, R. T.: Architectural styles and the design of network-based software architectures, *University of California*, 2000.
- [WV02] Weikum, G.; Vossen, G.: Transactional information systems: theory, algorithms, and the practice of concurrency control and recovery. *Morgan Kaufmann Publ.*, 2002.
- [Här05] Härder, T.: Caching over the entire user-to-data path in the internet, *Data Management in a Connected World*, pp. 150–170, 2005.
- [Cat11] Cattell, T.: Scalable sql and nosql data stores, *ACM SIGMOD Record*, vol. 39, no. 4, pp. 12–27, 2011.
- [CJP+11] Curino, C. A.; Jones, E. P. C.; Popa, R. A.; Malviya, N.; Wu, E.; Madden, S. R.; Balakrishnan, H.; Zeldovich, N.: Relational cloud: A database-as-a-service for the cloud, *Proceedings of the 5th Biennial Conference on Innovative Data Systems Research*, pp. 235–241, Pacific Grove, CA, 2011.
- [GL02] Gilbert, S., Lynch, N.: Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 2002.

Optimierte Speicherzuteilung an Operatoren in Datenbanksystemen

Jan Kristof Nidzwetzki

knidzwetzki@gmx.de

Abstract: Um Anfragen in DBMS effizient auszuführen, ist es entscheidend, vorhandenen Arbeitsspeicher zwischen Operatoren möglichst gut aufzuteilen. Erhält ein Operator zu viel Speicher, ist dieser vergeudet. Erhält ein Operator zu wenig Speicher, arbeitet er langsamer als möglich. Zudem profitiert nicht jeder Operator gleich stark von zusätzlichem Speicher. Diese Arbeit geht der Frage nach, wie die Charakteristik eines Operators, der Zusammenhang zwischen Laufzeit und benötigtem Arbeitsspeicher, durch eine Funktion approximiert werden kann.

1 Einleitung

Viele *Datenbankmanagementsysteme* (*DBMS*) lassen sich mit deskriptiven Abfragesprachen wie der *Structured Query Language* (*SQL*) ansprechen. Diese Sprachen beschreiben lediglich, wie ein Ergebnis auszusehen hat, nicht aber wie dieses berechnet wird [SSH11].

Es ist Aufgabe des *DBMS*, die Anfrage in einen *Ausführungsplan* zu übersetzen. Enthält ein *Ausführungsplan* mehrere *Operatoren*, so stehen diese in Konkurrenz um die Nutzung von Ressourcen. Das *DBMS* muss entscheiden, wie viel Speicher ein Operator nutzen darf [ND98] [HSS00]. Teilt man einem Operator mehr Arbeitsspeicher als benötigt zu, so ist er vergeudet. Teilt man einem Operator zu wenig Arbeitsspeicher zu, so arbeitet der Operator langsamer als möglich, da z. B. Daten auf die Festplatte ausgelagert werden müssen. Zudem profitiert nicht jeder Operator gleich stark von weiterem Arbeitsspeicher.

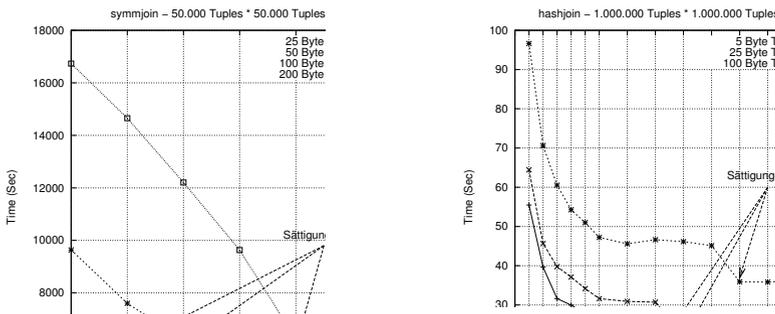
Grundsätzlich erwartet man zwei Reaktionen der Laufzeit bei einer höheren Arbeitsspeicherzuteilung: (1) der Operator kann den Arbeitsspeicher nutzen, um die Arbeit zu beschleunigen, (2) der zusätzliche Speicher kann nicht genutzt werden, da bereits alle Daten im Arbeitsspeicher vorliegen. Ziel des *DBMS* ist es, den Arbeitsspeicher so aufzuteilen, dass die Laufzeit des *Ausführungsplans* minimal wird. Ein tieferes Verständnis des Laufzeitverhaltens von Operatoren kann dem *DBMS* bei der Aufteilung des Speichers helfen.

An der *Fernuniversität in Hagen* ist ein frei verfügbarer *DBMS*-Prototyp mit dem Namen *SECONDO* entwickelt worden [SEC13] [GdAA⁺05]. Dieses *DBMS* wurde genutzt, um zu untersuchen, wie sich verschiedene Arbeitsspeicherzuteilungen auf die Laufzeit von Operatoren auswirken. Der in *SECONDO* enthaltene *Optimierer* kann für viele Operatoren zuverlässige Selektivitätsschätzungen abgeben. Diese Schätzungen ermöglichen es vor dem Ausführen einer Anfrage die Anzahl der zu verarbeitenden Tupel der Operatoren im *Ope-*

ratorbaum zu bestimmen. Zudem ist in SECONDO das Konzept der *Operator basierenden Fortschrittsschätzung* implementiert [Güt08]. Dieses Konzept ermöglicht es Operatoren, Schätzungen über ihre Laufzeit abzugeben. Grundlage hierfür bilden *Kostenmodelle*, die das Verhalten der Operatoren beschreiben.

2 Experimente

Für verschiedene Operatoren wurde im Rahmen dieser Arbeit untersucht, wie die Laufzeiten variieren, wenn die Größe des zugeteilten Arbeitsspeichers verändert wird [Nid12]. Die Experimente haben das erwartete Verhalten bestätigt. Bis zu einem gewissen Punkt kann der Operator von zusätzlichem Arbeitsspeicher profitieren und die Laufzeit sinkt. Nach diesem Punkt bleibt die Laufzeit konstant und der Arbeitsspeicher ungenutzt. Der Punkt, an dem dieser Übergang stattfindet, wird im Folgenden als *Sättigungspunkt* bezeichnet.



(a) Laufzeitverhalten des Join-Operators symmjoin (b) Laufzeitverhalten des Join-Operators hashjoin (Variante eines Nested-Loop-Join)

Abbildung 1: Laufzeitverhalten zweier Join-Operatoren mit verschiedenen Eingabegrößen.

Die Experimente haben gezeigt, dass die Operatoren auf zwei Arten auf weiteren Arbeitsspeicher reagieren. Einige weisen ein lineares Verhalten zwischen Arbeitsspeicher und Laufzeit auf (Abb. 1(a)), andere ein antiproportionales (Abb. 1(b)). Liest man die Abbildung 1(b) von rechts nach links, so sieht man, dass links des *Sättigungspunktes* die Laufzeit sprunghaft ansteigt und zunächst nur eine sehr leichte Steigung aufweist. Der sprunghafte Anstieg lässt sich mit der Implementation der Operatoren erklären. Beim ersten Tupel, welches nicht mehr im Arbeitsspeicher gehalten werden kann, müssen ggf. Daten auf die Festplatte ausgelagert werden und der Operator verändert seinen Algorithmus.

Die Auswertung der Experimente hat ergeben, dass sich das Laufzeitverhalten sehr gut durch die beiden Funktionen $mx + b$ und $\frac{h}{x} + b$ approximieren lässt (Abb. 2). Die Variablen m , h und b stellen Lageparameter dar, die Variable x steht für den zugeteilten Arbeitsspeicher.

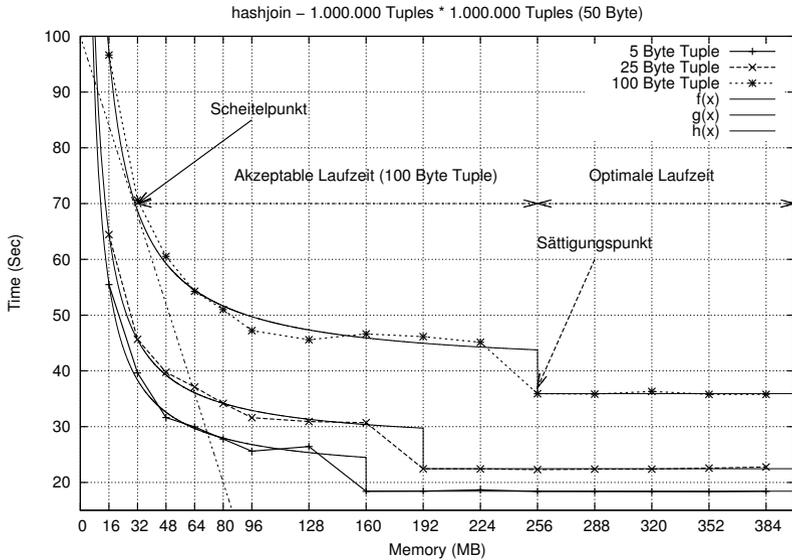


Abbildung 2: Bereich der akzeptablen Laufzeit und der optimalen Laufzeit für einen Join-Operator. Die Funktion $\frac{h}{x} + b$ wurde zur Approximation genutzt. Die parametrisierten Funktionen f , g und h approximieren das Verhalten des Operators bei bestimmten Eingabegrößen. Die optimale Laufzeit liegt rechts des Sättigungspunktes vor. Die akzeptable Laufzeit links des Sättigungspunktes.

Für jede Anwendung müssen diese Funktionen korrekt parametrisiert werden. Durch den *Optimierer* von SECONDO kann abgeschätzt werden, wie viele Tupel jeder Operator zu verarbeiten hat. Die *Fortschrittsschätzung* erlaubt es abzuschätzen, wie lange die Berechnung dauert. Die Schätzungen ermöglichen es, die Lageparameter zu berechnen und dem *DBMS* die parametrisierte Funktion für weitere Optimierungen zur Verfügung zu stellen.

Bei vielen Operatoren kann Arbeitsspeicher eingespart werden, ohne dass die Laufzeit sonderlich stark ansteigt. In Abbildung 2 steigt die Laufzeit von 38 Sekunden im *Sättigungspunkt*, um ca. den Faktor zwei, auf 73 Sekunden im *Scheitelpunkt* an. Der Arbeitsspeicher kann jedoch um den Faktor acht, von 256 MB im *Sättigungspunkt* auf 32 MB im *Scheitelpunkt*, reduziert werden.

3 Fazit

Die beiden vorgestellten Funktionen ermöglichen es, das Laufzeitverhalten von Operatoren gut zu approximieren. Die Kenntnis einer solchen Funktion hilft, den Arbeitsspeicher effizient zuzuteilen. Hierdurch wird es dem *DBMS* möglich, genau den Operatoren mehr Arbeitsspeicher zuzuweisen, die am stärksten davon profitieren. Da andere *DBMS* ähnlich wie SECONDO arbeiten, ist zu erwarten, dass in diesen ebenfalls eine Approximation der Laufzeit möglich ist. Dies sollte in einer weiteren Arbeit genauer untersucht werden.

Literatur

- [GdAA⁺05] Ralf Hartmut Güting, Victor Teixeira de Almeida, Dirk Ansorge, Thomas Behr, Zhi-ming Ding, Thomas Höse, Frank Hoffmann, Markus Spiekermann und Ulrich Telle. SECONDO: An Extensible DBMS Platform for Research Prototyping and Teaching. In *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*, Seiten 1115–1116. IEEE Computer Society, 2005.
- [Güt08] Ralf Hartmut Güting. Operator-Based Query Progress Estimation. In *Informatik Be-richte*, number 343. Fernuniversität in Hagen, Feb 2008.
- [HSS00] A Hulgeri, S Sudarshan und S Seshadri. Memory cognizant query optimization. Proceedings of COMAD, 2000.
- [ND98] Biswadeep Nag und David J. DeWitt. Memory allocation strategies for complex de-cision support queries. In *Proceedings of the seventh international conference on Information and knowledge management, CIKM '98*, Seiten 116–123, New York, NY, USA, 1998. ACM.
- [Nid12] Jan Kristof Nidzwetzki. *Operator-Kostenmodelle für Fortschrittsschätzung und Op-timierung in Datenbanksystemen*. Bachelorarbeit, Lehrgebiet Datenbanksysteme für neue Anwendungen, Fernuniversität in Hagen, 2012.
- [SEC13] SECONDO Webseite, 2013. <http://dna.fernuni-hagen.de/Secondo.html/> - Abgerufen am 10.01.2013.
- [SSH11] Gunter Saake, Kai-Uwe Sattler und Andreas Heuer. *Datenbanken - Implementierungs-techniken, 3. Auflage*. mitp-Verlag, Redline GmbH, 2011.

Bewertung von Diskriminierungsnetzwerken für regelbasierte Systeme

Fabian Ohler, Kai Schwarz

Abstract: Immer größere Mengen von Informationen werden gespeichert, doch je nach Darstellung lassen sich diese nur schwer von Maschinen verarbeiten. Eine Möglichkeit, Informationen mit Hilfe von Maschinen zu interpretieren bietet das Speichern in verarbeitbarer Form, z. B. als Fakt eines Regelbasierten Systemen. Regelbasierte Systeme verwenden Diskriminierungsnetzwerke um Zwischenergebnisse der Auswertung zu speichern und so die Verarbeitung der Regeln zu beschleunigen. Da diese Netzwerke häufig sehr komplex aufgebaut sind, ist es wichtig, Optimierungen oder Vergleiche zweier semantisch gleicher Diskriminierungsnetzwerke bezüglich der strukturellen Effizienz zu ermöglichen. Daher wurde in der Bachelorarbeit zu diesem Abstract ein Algorithmus zur Bewertung der Struktur von Diskriminierungsnetzwerken und deren Effizienz vorgestellt. Diese Bewertungen basieren auf einer vereinheitlichten Form der Netzwerkstruktur und Abschätzungen bezüglich der Änderungsfrequenz von Fakten. Mithilfe den Bewertung lassen sich mehrere semantisch gleiche Diskriminierungsnetzwerke bezüglich ihrer Laufzeitkosten vergleichen.

1 Einführung

Dieses Abstract stellt die Bachelorarbeit mit dem Titel „Bewertung von Diskriminierungsnetzwerken für regelbasierte Systeme“ von Fabian Ohler am Lehrstuhl für Informatik 5 der RWTH Aachen vor.

Immer größere Mengen von Informationen werden gespeichert, doch je nach Darstellung lassen sich diese nur schwer von Maschinen verarbeiten. Prominentes Beispiel dafür ist die riesige, stetig wachsende im Internet verfügbare Informationsmenge, die zu großen Teilen in zwar für den Mensch, nicht aber für Maschinen verständlicher Textform vorliegt. Maschinen können diese Texte zwar durchsuchen, die Bedeutung aber nur sehr begrenzt erfassen.

Eine Möglichkeit, Informationen so zu speichern, dass sie auch von Maschinen interpretiert werden können, bieten Wissensbasierte Systeme. Eine Gattung dieser Systeme ist das Regelbasierte System, in dem Informationen als Fakten gespeichert werden. Neben diesen Fakten besteht das System aus Regeln, die einen Bedingungs- und einen Aktionsteil haben. In erstem werden Fakten geprüft und im Falle der Erfüllung die im zweiten Teil gespeicherten Aktionen ausgeführt, die beispielsweise neue Fakten zum System hinzufügen. Auf diese Systeme wird nun näher eingegangen.

Ein Regelbasiertes System besteht aus den oben genannten Regeln, die die Regelbasis bilden, den Fakten und dazu einem Kontrollsystem, das geeignete Regeln identifiziert und anwendet sowie die Datenbasis (Fakten) aktualisiert. Um festzustellen, welche Regeln auf Grund ihres Bedingungsteils anwendbar sind, ohne dabei für jeden Durchlauf des Systems

alle Regelbedingungen neu auswerten zu müssen, lassen sich Diskriminierungsnetzwerke nutzen. Diese basieren auf dem Grundgedanken der dynamischen Programmierung. In den Netzwerken werden Teilergebnisse der Auswertung von Bedingungen mit dem Ziel zwischengespeichert, diese in Folgedurchläufen für die Berechnung auf veränderten Fakten basierender Regelbedingungen zu benutzen.

Da Diskriminierungsnetzwerke im Fokus dieser Arbeit stehen, wird auf diese nun genauer eingegangen. Ein Diskriminierungsnetzwerk ist ein gerichteter Graph, in dem zu jeder Operation einer Regelbedingung ein Knoten mit internem Speicher für Zwischenergebnisse existiert. Jeder solche Knoten steht somit für einen Teilausdruck einer Regelbedingung. Durch Verbindungen zwischen den Knoten, über die zu verarbeitende Fakten weitergereicht werden, werden diese zur gesamten Regelbedingung verknüpft. In jedem solchen Netzwerk gibt es zusätzlich einen Wurzelknoten ohne Eingänge, der über seine Ausgänge das Hinzufügen und Löschen von Fakten aus der Datenbasis weitergibt. Zu jeder Regel existiert mindestens ein Regelendknoten, dem einzigen Knotentyp ohne Ausgänge, dessen interner Speicher die Fakten enthält, die die Regelbedingung erfüllen. Daher zeigt dieser durch einen nicht-leeren Speicher die Ausführbarkeit der zugehörigen Regel an. Bei den restlichen Knoten lässt sich unterscheiden zwischen Alpha-Knoten mit genau einem Eingang und Beta-Knoten mit mehr als einem Eingang, wobei letztere die Fakten ihrer Eingänge miteinander verknüpfen, um dann ihre Operation darauf anzuwenden.

Bei Regelbasierten Systemen unterscheidet man grundsätzlich zwischen vorwärts- und rückwärtsverkettenden Systemen. Vorwärtsverkettend sind Systeme, die aus der Existenz eines Faktums Folgerungen schließen („WENN Fakt, DANN ...“), um beispielsweise eine Diagnose zu erstellen. Bekannte vorwärtsverkettende Systeme sind CLIPS und JBoss Drools, rückwärtsverkettende Systeme kommen z. B. in der Logikprogrammierung (Prolog) zum Einsatz.

Die in dieser Arbeit zu bewertenden Diskriminierungsnetzwerke wurden für vorwärtsverkettende Systeme entwickelt, weshalb die folgenden Definitionen sich auf diese Systeme beziehen. Rückwärtsverkettende Systeme werden teilweise durch vorwärtsverkettende Systeme simuliert [Fri03] oder setzen auf Diskriminierungsnetzwerken auf [HL]. Je nach Umsetzung treffen die im weiteren Verlauf getroffenen Aussagen daher auch für rückwärtsverkettende Systeme zu.

2 Motivation

Da in Regelbasierten Systemen oft eine große Zahl an Regeln abgebildet wird, ist die Erstellung des Diskriminierungsnetzwerkes von Hand nicht praktikabel. Diese Aufgabe wird daher von Konstruktionsalgorithmen übernommen, welche häufig unterschiedliche Netzwerke produzieren. Die bekanntesten Vertreter sind Rete [For82], Gator [Han93] und TREAT [Mir87].

Schon ein Diskriminierungsnetzwerk, das nur eine Regel repräsentiert, ist in seiner Struktur nicht eindeutig, ein die gesamte Regelbasis abbildendes Netzwerk kann somit viele verschiedene Formen haben. Diese verschiedenen Netzwerke haben somit zwar die gleiche Semantik, unterscheiden sich aber möglicherweise in ihrer Komplexität bezüglich

resultierender Laufzeit und benötigtem Speicherbedarf. Damit sind auch schon die beiden wichtigsten Merkmale genannt, welche die Güte eines Diskriminierungsnetzwerkes beschreiben könnten. In den meisten Fällen ist es erstrebenswert, dass dem regelbasierten System ein Diskriminierungsnetzwerk mit minimaler Laufzeit und Speicherbedarf zu Grunde liegt.

Doch um entscheiden zu können, welches Diskriminierungsnetzwerk sich bezüglich Laufzeit und Speicherbedarf optimal verhält, bedarf es eines Mechanismus, der Diskriminierungsnetzwerke im Hinblick auf bestimmte Kriterien vergleichen kann. Dieser Vergleich kann zum Beispiel angestellt werden, wenn Diskriminierungsnetzwerke bewertet werden können. Eine solche Bewertung soll durch den Algorithmus der in diesem Abstract vorgestellten Arbeit ermöglicht werden.

Mit Hilfe der Bewertungsfunktion lassen sich wie oben erwähnt Diskriminierungsnetzwerke mit identischer Semantik vergleichen. Dies kann zum Beispiel bei der Wahl des passenden Diskriminierungsnetzwerks helfen oder Optimierungsversuche messen und belegen.

3 Vorhandene Ansätze

Das Problem der Bewertung von Diskriminierungsnetzwerken ist nicht neu. Es ergab sich beispielsweise immer dann, wenn gezeigt werden sollte, dass ein Diskriminierungsnetzwerk in bestimmter Weise besser ist als ein anderes. Im Laufe der Jahre wurden immer wieder neue Algorithmen zur Konstruktion von Diskriminierungsnetzwerken vorgestellt, die sich in der Regel als Verbesserung gegenüber schon existierenden Algorithmen sahen. Diese Verbesserungen wurden auf verschiedene Arten gezeigt.

Im Fall von TREAT [Mir87] wurde die Verbesserung gegenüber Rete [For82] mittels Durchführung von Benchmarks gezeigt. Die Verbesserung, die Gator [HH93] gegenüber TREAT und Rete birgt, wurde durch eine aufwendige Kostenfunktion gezeigt, die die Kosten für einzelne Regeln berechnet, indem sie statistische Informationen über Fakthäufigkeiten und Filterselektivitäten nutzt, um Laufzeit und Speicherbedarf vorauszusagen.

Vergleiche durch Benchmarks sind allerdings stets nur so gut wie der Benchmark selbst. Da in Benchmarks nicht alle Situationen gemessen werden können, lassen sich möglicherweise Optimierungen der Netzwerke, die dem Urheber der Benchmarks nicht bekannt waren, nicht hinreichend messen. Die Gator-Kostenfunktion beschränkt den Blick unter anderem stets auf einzelne Regeln, das gesamte Diskriminierungsnetzwerk kann damit nicht bewertet werden. Die Summe der Regelkosten würde beispielsweise die geteilte Nutzung von Knoten in verschiedenen Regeln nicht berücksichtigen.

4 Lösungsansatz

In der Arbeit wurde eine allgemeine Struktur für Diskriminierungsnetzwerke entwickelt, die eine einheitliche Bewertung zulässt. Für die Komponenten der Netzwerke dieser Struk-

tur wurden im weiteren Verlauf Kostenfunktionen erarbeitet. Diese Kostenfunktion benötigt als Eingabewerte einige Angaben über die Eigenschaften bzw. Zusammensetzung der Faktenbasis. Bei einer Regel, welche zwei Zusammenhänge abfragt, gibt es Optimierungspotential bezüglich der Reihenfolge der Prüfung. Dies ist vergleichbar mit der Erstellung eines Join-Plans in Datenbanksystemen. Es ist ersichtlich, dass zuerst der Zusammenhang überprüft werden sollte, der in der zugehörigen Faktenbasis am seltensten vorkommt, um den Aufwand möglich gering zu halten. Daher kann die Bewertung einer gegebenen Reihenfolge nur unter Berücksichtigung der zu Grunde liegenden Faktenbasis erfolgen. Durch Überführung eines Netzwerkes in die allgemeine Struktur und anschließende Bewertung kann dadurch jedes Diskriminierungsnetzwerk bewertet werden.

Im Laufe der Bewertung mussten aus verschiedenen Gründen Vereinfachungen getroffen und Abschätzungen angestellt werden. Dazu zählen unter anderem die Abschätzung eines Mittelwertes für die Dopplung oder die Vereinheitlichung der Faktgröße. Beide Größen hätten als für die Bewertung notwendig deklariert werden können, jedoch schien die Verbesserung der Kostenabschätzung im Vergleich zum Mehraufwand für den die Kostenfunktion Nutzenden nicht gerechtfertigt. Das Gleiche gilt auch für die starke Vereinfachung bezüglich der Verbundgrößen. Eine Angabe aller Größen zu erfordern, hätte die Menge der zu Grunde gelegten Werte rasant wachsen lassen, wodurch der abstrakte Charakter des Algorithmus verloren gegangen wäre. Allerdings lassen sich die getroffenen Vereinfachungen ohne größeren Aufwand durch die korrekten Werte ersetzen, so dass eine genauere Bewertung durchgeführt werden kann.

5 Evaluation

Die Bewertungsfunktion wurde in Java für das quelloffene Regelbasierte System Jamocha implementiert und getestet. Diese Tests zeigten, dass die Bewertungsfunktion die Unterschiede von semantisch äquivalenten Diskriminierungsnetzwerken bezüglich Laufzeit und Speicherbedarf korrekt wiedergibt. Auch relativ komplexe Netzwerkkomponenten wie negierte Bedingungelemente werden korrekt bewertet.

Literatur

- [For82] Charles L. Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19(1):17 – 37, 1982.
- [Fri03] Ernest Friedman. *Jess in action: rule-based systems in java*. Manning Publications Co., Greenwich, CT, USA, 2003.
- [Han93] Eric N. Hanson. Gator: A Discrimination Network Structure for Active Database Rule Condition Matching. Bericht, University of Florida, 1993.
- [HH93] Eric N. Hanson und Mohammed S. Hasan. Gator: An Optimized Discrimination Network for Active Database Rule Condition Testing. Bericht, University of Florida, 1993.
- [HL] Peter V. Homeier und Thach C. Le. ECLIPS: An Extended CLIPS for Backward Chaining and Goal-Directed Reasoning.
- [Mir87] Daniel P. Miranker. TREAT: A Better Match Algorithm for AI Production Systems; Long Version. Bericht, University of Texas at Austin, Austin, TX, USA, 1987.

Thesis: The Distributed Radix Join Algorithm – Efficient Distributed Join Processing on Modern Hardware

Alexander van Renen

Wolf Rödiger

Technische Universität München · Boltzmannstraße 3 · D-85748 Garching
{renen, roediger}@in.tum.de

Abstract: The ever increasing volume of data in scientific as well as commercial information systems calls for scalable databases. Distributed database systems can satisfy this demand as they are scalable to higher performance requirements simply by adding more computing nodes. However, they require efficient ways to query distributed relations in order to achieve high performance.

We describe an efficient distributed algorithm for joining two large distributed relations. Our algorithm achieves a throughput of 140 M tuples per second on a cluster, consisting of 16 standard desktop machines. Furthermore, an analysis of the scalability of the algorithm is provided and it is shown that its performance on today’s commodity hardware is bound by the underlying network bandwidth.

1 Motivation

Main memory database systems are one of the most important recent developments in the field of database research. Systems like HyPer [KN11] achieve unprecedented OLTP and OLAP performance which is many orders of magnitude higher than traditional disk-based database systems. However, the amount of memory which can be installed in a single server is inevitably limited. While one server might be sufficient even for large online warehouses like Amazon, some applications still exceed this limit, e.g., research facilities like CERN which need to analyze an enormous amount of data. A solution to this problem are distributed main memory database systems which allow to increase the available main memory capacity by adding more computing nodes.

The most expensive operator in distributed query processing is the join and therefore its optimization should be the primary concern when trying to achieve high query performance. This paper presents the Distributed Radix Join which adapts the centralized radix join described by Kim et al. [KKL⁺09] to a distributed setting. Our parallel algorithm utilizes the full capabilities of today’s multi-core architectures.

2 The Distributed Radix Join

The Distributed Radix Join is designed for shared-nothing [Sto86] commodity-hardware clusters, which are common in today’s data centers. It computes an equi-join between two relations \mathcal{R} and \mathcal{S} . Distributed database systems fragment the input relations across the compute nodes of a cluster. Consequently, each node holds a part \mathcal{R}_i and \mathcal{S}_i of both relations. We call these nodes the worker nodes as they will later perform the actual join.

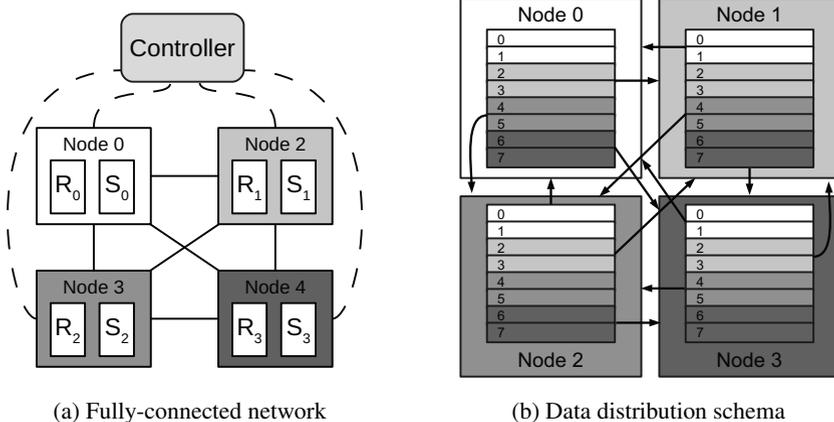


Figure 1: The setup of the Distributed Radix Join.

An additional coordinator node synchronizes the workers. All nodes are connected to each other via a TCP network, as illustrated in Figure 1a.

The fundamental idea of hash join algorithms is to divide the two input relations into many small disjunct partitions. As a consequence, the join can be computed independently for each partition pair [DKO⁺84]. We extend this idea to a distributed system, allowing a parallel join execution on separate worker nodes.

The Distributed Radix Join starts by computing the partitions on the worker nodes using the parallel radix cluster algorithm which is described in [MBK02]. Figure 1b shows an example for a radix partitioning with three bit, which results in eight partitions. After the relations are partitioned, the coordinator globally assigns the partitions to nodes, as indicated by the shades of gray. For example the white partitions 0 and 1 are assigned to node 0. In the next phase the worker nodes use the network to redistribute the currently scattered partitions to the assigned nodes. In the example, all tuples of partitions 2 and 3 are sent to node 1. The join of partition 2 is able to start as soon as all tuples of partition 2 (both R and S) are completely available at node 1.

Our implementation processes the tuples while they arrive over the network in order to both maximize the throughput and minimize the time until the first results are available. Therefore the join of two partitions of one node can be performed in parallel. The local join takes full advantage of the multi-processing capabilities of the worker node.

3 Evaluation

The Distributed Radix Join is implemented using C++11 and compiled with the open source compiler GCC-4.7. For our experiments we used a homogeneous cluster consisting of 16 computing nodes, each running a linux kernel 3.2. The nodes are equipped with an Intel Core 2 Quad Q67000 (four cores, each 2.66 GHz), 8 GB RAM main memory and a 1 GBit/s Ethernet network (in practise around 870 Mbit/s). The algorithm computes the join between two relations which each contains a 32 bit join key and a 32 bit payload.

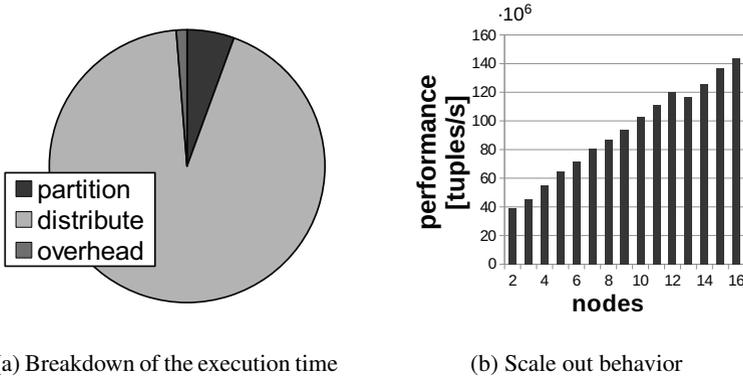


Figure 2: Evaluation of the Distributed Radix Join on a cluster of 16 nodes.

The centralized radix join by Kim et al. [KKL⁺09] achieves 100 M tuples per second on a machine equipped with an Intel Core i7 with four cores at 3.2 GHz and 6 GB RAM. Our algorithm achieves a throughput of 140 M tuples per second in our low-end cluster when joining two relations of $1600 \cdot 2^{20}$ tuples each. Figure 2a shows the breakdown of the execution time for each phase. The performance is clearly dominated by the network phase. Additionally, the local join computation utilizes only 35% of the CPU. Therefore, we expect that the join performs even better on future networks, e.g., 10 GBit Ethernet.

It is difficult to achieve the theoretical limit of the network bandwidth as several intricate effects emerge and affect the performance. An example is cross traffic which is caused by two nodes sending to the same node, thus reducing the speed of both transmissions while leaving another one unused. Our algorithm synchronizes the usage of the connections via the coordinator to avoid cross traffic and is able to achieve a throughput of 640 MBit/s on each connection. More details on these techniques can be found in [vR12].

The scale out behavior is an important measure for every distributed algorithm, as it describes how well it performs when the input and the number of nodes is increased proportionally. Figure 2b shows the performance in a cluster with up to 16 nodes on the x axis. The size of both relations is also increasing with $|\mathcal{S}| = |\mathcal{R}| = 100 \cdot 2^{20} \cdot n$, where n is the number of nodes. Within the bounds of our limited cluster size it seems as if the Distributed Radix Join exhibits a linear scale out behavior, as the performance per node is stabilizing at about 9 M tuples per second, as depicted in Figure 2b.

4 Conclusion

The Distributed Radix Join is part of our research effort to build a distributed main memory database system and will be integrated into the ScyPer system [MRR⁺13]. The linear scale out behavior makes it an excellent candidate for larger networks, as long as the network is fully-connected. We have shown that the join performance is bound by the network bandwidth and should therefore be able to achieve even higher performance in future networks.

Acknowledgments

Wolf Rödiger is a recipient of the Oracle External Research Fellowship, and this research is supported in part by this Oracle Fellowship.

References

- [DKO⁺84] David J. DeWitt, Randy H. Katz, Frank Olken, Leonard D. Shapiro, Michael R. Stonebraker, and David A. Wood. Implementation techniques for main memory database systems. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, SIGMOD '84, pages 1–8, New York, NY, USA, 1984. ACM.
- [KKL⁺09] Changkyu Kim, Tim Kaldewey, Victor W. Lee, Eric Sedlar, Anthony D. Nguyen, Nandathur Satish, Jatin Chhugani, Andrea Di Blas, and Pradeep Dubey. Sort vs. Hash revisited: fast join implementation on modern multi-core CPUs. *Proc. VLDB Endow.*, 2(2):1378–1389, August 2009.
- [KN11] Alfons Kemper and Thomas Neumann. HyPer: A hybrid OLTP&OLAP main memory database system based on virtual memory snapshots. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, ICDE '11, pages 195–206, Washington, DC, USA, 2011. IEEE Computer Society.
- [MBK02] Stefan Manegold, Peter Boncz, and Martin Kersten. Optimizing Main-Memory Join on Modern Hardware. *IEEE Trans. on Knowl. and Data Eng.*, 14(4):709–730, July 2002.
- [MRR⁺13] Tobias Mühlbauer, Wolf Rödiger, Angelika Reiser, Alfons Kemper, and Thomas Neumann. ScyPer: A Hybrid OLTP&OLAP Distributed Main Memory Database System for Scalable Real-Time Analytics. In *GI-Fachtagung Datenbanksysteme für Business, Technologie und Web*, BTW '13, 2013.
- [Sto86] M. Stonebraker. The case for shared nothing. *Database Engineering Bulletin*, 9(1):4–9, 1986.
- [vR12] Alexander van Renen. Efficient Distributed Join Processing on Modern Hardware. *TUM, Bachelor's Thesis*, October 2012.

Analyse von Mikro-Blog-Daten

Ricardo Usbeck
Martin-Luther-Universität Halle-Wittenberg
Research & Development - Unister GmbH
ricardo.usbeck@googlemail.com

Abstract: Seit dem Entstehen des Web 2.0 gehören Blogs zum Kern der neuen Medienlandschaft. Insbesondere der Mikro-Blog-Dienst *Twitter* stellt eine der schnellst wachsenden Informationsquellen unserer Zeit dar. Über 200 Millionen regelmäßige Benutzer schreiben täglich mehr als 340 Millionen Nachrichten¹. Den Überblick über mögliche Verbindungen zwischen Hashtags und deren Themen zu behalten, fällt schwer. Twitter-Nutzer haben die Möglichkeit, über mit # *annotierte Worte* (sog. *Hashtags*) Nachrichten explizit zu verschlagworten. Diese Schlagworte sind nicht immer eindeutig und können für mehrere Themen verwendet werden. Der hier vorgestellte Ansatz erfasst mittels Data-Mining-Methoden Zusammenhänge zwischen Hashtags und visualisiert diese. Dadurch soll der Nutzer unterstützt werden, einen Überblick über besprochene Themen und deren Hashtags zu bekommen.

1 Motivation

Twitter ist ein seit 2006 existierender Mikro-Blogging-Dienst². Die Besonderheit hier ist, dass die *Tweets* maximal 140 Zeichen umfassen dürfen. Dies prägt die Art und Weise der Verwendung von Twitter entscheidend.

Ein durch die Nutzergemeinde erstmals 2008 eingeführtes Feature sind Hashtags [HTE10]. Diese von Twitter seit Ende 2008 übernommenen, besonderen Links werden durch '#' in Tweets annotiert. Nachrichten mit einem Hashtag können nicht nur von befreundeten Nutzern gesehen werden, sondern auch von allen Twitter-Nutzern, die dieses Hashtag verfolgen bzw. suchen.

Da es keine einheitlichen Richtlinien oder Mechanismen gibt, die den Gebrauch von Hashtags steuern, ist der mit ihnen verbundene Inhalt nicht eindeutig. So kann es passieren, dass das gleiche Thema unter *zwei oder mehr* Hashtags diskutiert wird. Einige Nutzer jedoch sehen nur die jeweiligen Teilinhalte ihrer beobachteten Hashtags, da ihnen die anderen thematisch verbundenen Hashtags unbekannt sind.

Ein Beispiel für ein solches Thema ist die Occupy-Bewegung in den USA, für die die Twitter-Gemeinde viele verschiedene Hashtags verwendet. Noch heute tauchen diverse Hashtag-Formen wie z. B.: #OccupyWallSt, #OWS und #occupywallstreet auf.

Die hier vorgestellten Untersuchungen liefern Ansätze zur verbesserten Verfolgung von

¹<https://business.twitter.com/basics/what-is-twitter/>, Stand: 27.12.2012

²<https://www.twitter.com>, Stand: 27.12.2012

Hashtags und den zu ihnen thematisch äquivalenten Hashtags, basierend auf den sie verbindenden Inhalten.

2 Verwandte Arbeiten

Eine der meist erwähnten Web-Applikationen in diesem Bereich ist `hashtagify.me` [Inf, Ben]. Bei Besuch der Seite erhält man einen durch ein Popularitätsmaß angeordneten Graphen aller Hashtags, die miteinander in Tweets vorkommen. `Hashtagify.me` ermöglicht auf sehr intuitive Weise die Suche nach einem für den eigenen Inhalt geeigneten Hashtag.

Ebenfalls ist die Benutzung von Hashtags auf Twitter und Schlagworten in anderen sozialen Netzen interessant. Huang et al. [HTE10] untersuchen bspw. die Unterschiede zwischen Hashtags und den Schlagworten in `delicious.us`, einem sozialen Bookmarkservice.

Der in dieser Arbeit entwickelte Beitrag zur Detektion von Korrelationen zwischen Hashtags wurde nach eingehender Literaturrecherche in bestehender Forschung noch nicht untersucht.

3 Methodik

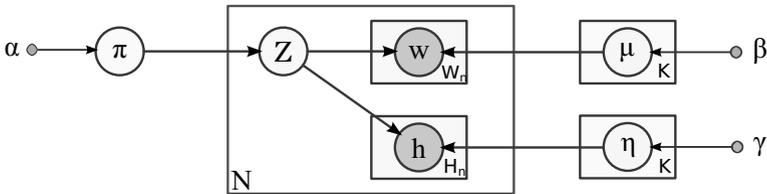


Abbildung 1: Grafisches Modell des angepassten Dirichlet-Multinomial-Clustering-Modells

Als Grundlage der Berechnung der Korrelation zwischen Hashtags dient ein *Dirichlet-Multinomial-Clustering-Modell (DMC)* [SMY07, Bis07], das in Abbildung 1 dargestellt ist. Die N beobachteten Tweets werden als Dokumente erfasst. Dabei wird jeder Tweet unterteilt in Hashtags H_n und andere Worte W_n . Das Entstehen von Tweets modelliert der Parametervektor Z , welcher die Zuordnung eines Dokumentes zu einem Cluster vornimmt. Die Größe der einzelnen Cluster regelt der Vektor π . Pro Cluster werden Prototypenvektoren durch das Dirichlet-Multinomial-Clustering-Modell gelernt. Dieses bedient sich der Dirichlet-Verteilungen (α, β, γ) als A-priori-Verteilungen für die Multinomialverteilungen μ und η .³

Die Parameter des angepassten Dirichlet-Multinomial-Clustering-Modells werden mittels *Variationaler Inferenz* nach Euler [Bis07] durch Aktualisierungsgleichungen approximiert. Diese Approximationen werden dann genutzt, um mittels der Pearson-Korrelation [RN88] die Zusammenhänge zwischen Hashtags zu errechnen.

³Zur Vereinfachung wurden Vektorpfeile stets weggelassen.

4 Experimente und Visualisierung

Neben verschiedenen Experimenten mit synthetischen Datensätzen, bei denen die Effektivität des Modells zum Finden von Zusammenhängen erfolgreich getestet wurde, soll hier insbesondere auf die Visualisierung eingegangen werden.

Das zuvor vorgestellte DMC-Modell ermöglicht die Auswertung der Korrelation zwischen zwei und mehr Hashtags. Allerdings sind – ohne visuelle Hilfen – Analysen nur schwer durchführbar und Korrelationsmatrizen für den interessierten, aber stofffremden Nutzer kaum zu verwenden.

Die Visualisierung zeigt die Gemeinsamkeiten und Unterschiede zweier Modelle. Es werden zwei radial nebeneinander angeordnete Graphen dargestellt. Die Beziehungen zwischen dem im Mittelpunkt des jeweiligen Graphen dargestellten Hashtag und aller mit ihm über ein Kriterium zusammenhängenden Hashtags werden visualisiert. Die Farbe der Kante gibt die Stärke des Zusammenhangs auf einer Hitzeskale an. Die Visualisierung kann verschiedene Modelle entlang einer Zeitachse darstellen und so Themenverläufe anzeigen.

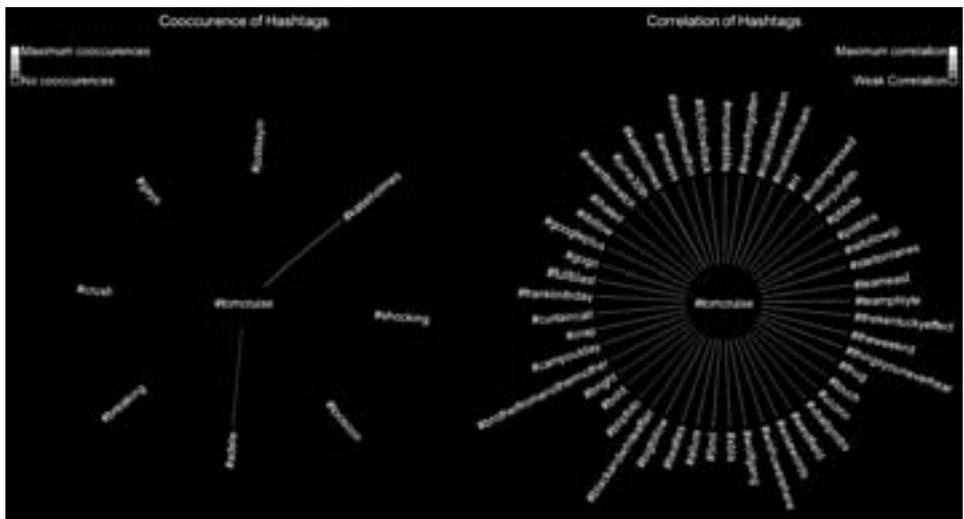


Abbildung 2: Links: Mehr als zweimal im Beobachtungszeitraum eines Tages miteinander getwitterte Hashtags. Rechts: Zusammenhang über die Pearson-Korrelation, errechnet über das DMC-Modell, 29. Juni, beobachtetes Hashtag '#tomcruise'

Wie in Abbildung 2 zu sehen, zeigt sich die effiziente Auswertbarkeit der Visualisierung durch die radiale Anordnung, alphabetische Sortierung und Abbildung der Korrelationsstärke auf eine Hitzeskale. Ebenfalls zeigt sich die Stärke des DMC-Modells gegenüber der Kookkurrenz. Existieren keine oder nur wenige direkt miteinander getwitterten Hashtags, kann dieses Modell über das Hintergrundvokabular Zusammenhänge herstellen (z. B. #curtaincall). Der Nutzer kann somit recht einfach den Verlauf ausgewählter Zusammenhänge im zeitlichen Voranschreiten visuell auswerten. Allerdings versagt diese Visualisierung bei einer größeren Anzahl an Hashtags und Zusammenhängen, welche

meist durch größere Zeiteinheiten bedingt sind. Trotzdem kann der Betrachter daraus lohnenswerte Erkenntnisse ziehen, beispielsweise die stärksten Einflüsse auf ein bestimmtes Hashtag.

5 Zusammenfassung und Ausblick

Es wurde ein Ansatz zur Identifikation von Zusammenhängen zwischen Hashtags gezeigt. Hierzu wurden Data-Mining-Modelle angepasst, implementiert, getestet, optimiert und miteinander verglichen. Dabei wurde die Verwendbarkeit eines mit drei A-priori-Verteilungen versehenen DMC-Modells zur Korrelation von Hashtags nachgewiesen.

Insbesondere sind verbesserte Korrelationsmaße nötig, um aus den geclusterten Daten sinnvolle Erkenntnisse ziehen zu können. Weitere Verbesserungen sind im Bereich der Vorverarbeitung anzusiedeln – so beispielsweise das Auflösen von Links in Tweets zum Nutzen weiterer Inhalte. In Anbetracht der steigenden Datenmenge ist vor allem der Bereich Parallelisierung und Implementierung einer Online-Variante des Algorithmus sinnvoll.

Aus diesem Ansatz können beispielsweise Anwendungen im Bereich des wissenschaftlichen oder historischen Journalismus entstehen, sowie tiefere Einblicke in das Nutzerverhalten auf Twitter erlangt werden. Derzeit wird der Ansatz im Bereich der Medien-Entwicklung von uns weiter untersucht und eingesetzt, insbesondere im Hinblick auf das Verschmelzen und Verdrängen bestimmter Journale.

Literatur

- [Ben] Shea Bennett. Visually Explore Twitter Hashtags and Their Relationships With Hashtagify. <http://www.mediabistro.com/alltwitter/>, Stand 20. Juli 2012.
- [Bis07] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing. Auflage, Oktober 2007.
- [HTE10] Jeff Huang, Katherine M. Thornton und Efthimis N. Efthimiadis. Conversational tagging in twitter. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, HT '10, Seiten 173–178, New York, NY, USA, 2010. ACM.
- [Inf] Infosthetics. Hashtagify: Exploring the Connections between Twitter Hash-tags. http://infosthetics.com/archives/2011/05/hashtagify_exploring_the_connections_between_twitter_hashtags.html, Stand 31. Juli 2012.
- [RN88] Joseph L. Rodgers und Alan W. Nicewunder. Thirteen Ways to Look at the Correlation Coefficient. *The American Statistician*, 42(1):59–66, 1988.
- [SMY07] Kugatsu Sadamitsu, Takuya Mishina und Mikio Yamamoto. Topic-based language models using Dirichlet Mixtures. *Systems and Computers in Japan*, 38(12):76–85, 2007.

Query Optimization for Large Sets of Predicates in the Database Management System SECONDO

Gero Willmes

gero.willmes@gmx.net

Zusammenfassung meiner Masterarbeit an der FernUniversität in Hagen, Lehrgebiet Datenbanksysteme für neue Anwendungen, Prof. Dr. Güting (abgegeben am 26. Januar 2012)

Abstract: In this thesis three graph based algorithms for optimizing conjunctive queries with large sets of predicates ($n > 50$) were developed and compared. All three algorithms make use of Kruskal's minimum spanning tree algorithm. They combine cost based optimization and heuristic optimization and run in the Database Management System SECONDO. In a number of benchmark tests it could be shown that the algorithms perform remarkably fast.

1 The Problem of Optimizing Large Sets of Predicates

SECONDO is an extensible Database Management System (DBMS) platform with the purpose of teaching architecture and implementation of database systems and for building research prototypes. SECONDO is briefly described in [GdAA⁺05] and in detail in [GBT04].

In general, query processing consists of three major steps: parsing, optimizing and evaluation (execution) (see Figure 1.1).

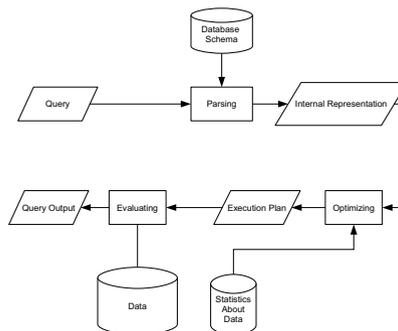


Figure 1.1: Query Processing

The **SECONDO** optimizer performs optimization of conjunctive queries in a relational environment. A conjunctive query is a restricted first-order query. That means, join conditions (predicates) can be connected by logical AND, but not by logical OR. Negation is also not allowed in a conjunctive query. Query optimization in **SECONDO** means the optimizer is given a set of relations and a set of predicates and produces a plan that is executable by the relational algebra module of the **SECONDO** kernel. The query language of the optimizer is very similar to SQL, in a notation adapted to PROLOG.

Optimizing in this context means ordering the predicates of a query and translating them into executable plan operators. The executable plan determines the predicate evaluation order. For the efficiency of an executable plan (and thus for the resulting execution time of a query) the predicate evaluation order is essential. Changing the evaluation order of two predicates may raise or drop the execution time by orders of magnitude.

At present, **SECONDO** calculates an optimal plan by considering all possible predicate orders of a query.

To achieve this, the optimizer searches the shortest path in a predicate order graph. In a predicate order graph (POG) edges represent predicates and nodes represent sets of evaluated predicates. A POG for n predicates represents all possible evaluation orders of these predicates. The cheapest executable plan is equivalent to the shortest path through the POG [GBT04]. As a shortest path algorithm Dijkstra is used.

Given a query with n predicates, there are $n!$ possible orders of evaluating the predicates. At present the **SECONDO** optimizer considers all $n!$ possible evaluation orders of a query so as to find the optimal evaluation order. This search strategy is called an *exhaustive search strategy* and works fine for queries with about 10 predicates: In a query that has 10 predicates, there are $10! = 3.628.800$ predicate orders to be considered. In a query that has 11 predicates, there are already $11! = 39.916.800$ possible predicate orders. Thus, it is NP-hard to calculate the optimal evaluation order of a predicate set. Figure 1.2 shows how optimization time depends on the number of predicates in the **SECONDO** optimizer (measured was the average optimization time of 100 random generated join queries).

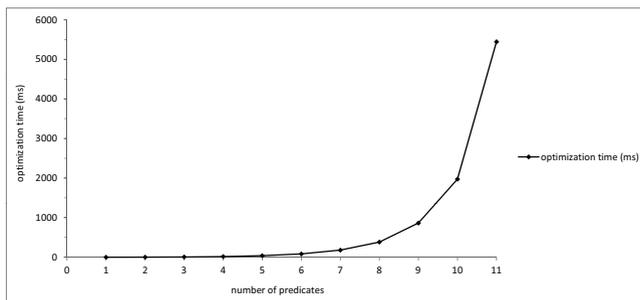


Figure 1.2: Optimization time related to the number of predicates in **SECONDO** optimizer

Nevertheless, there are practical needs to process queries that have more than 10 predicates (e. g . for data warehousing, query generating database system front ends, or complex views). 50 or even 100 predicates are desirable for this purpose. Hence, from the above explanations it is obvious, that an exhaustive search strategy (i.e. considering all possible predicate evaluation orders) is impractical for much more than 10 predicates. The search space is simply too large. Another procedure or heuristic needs to be developed which reduces the search space.

Thus the goal of this thesis is to develop and implement a strategy to extend the SECONDO optimizer in such a way as to enable it to produce preferably fast executable plans in reasonable time for conjunctive queries containing 50 predicates or more.

2 Related Work

Query optimization is a well studied field. In the literature there are many interesting and different approaches to solve this problem. A depicted introduction into the general complex of the problem is given in [Ioa97]. In the more special field of optimizing large join queries, the number of articles is considerably smaller. In [Feg98] Fegaras describes a heuristic polynomial time algorithm for join ordering, which minimizes the total costs of intermediate results. In [IK90] Ioannidis et al. introduce randomized algorithms for optimizing large join queries. [SMK97] discuss various heuristics, randomized and genetic algorithms, and perform detailed benchmarks on them. [Neu09] proposes a graph based algorithm that simplifies the query graph as much as needed for solving the optimization problem. Therefore, the algorithm successively simplifies the query graph until a given maximum complexity budget is reached. The complexity is measured by the number of disjoint subgraphs of the query graph. It seems that an intuitive, comprehensible, and complexity reducing query graph based approach is also suitable as a solution for optimizing large sets of predicates in SECONDO.

3 Optimization Algorithms and Results

Three algorithms for optimizing large sets of predicates were implemented and tested. All three algorithms combine *cost based optimization* and *heuristic optimization*. They build up on graph based join ordering using Kruskal's minimum spanning tree algorithm. I.e. first of all a query graph is built from the query. The relations in the query's from-clause become nodes of the query graph and the predicates in the query's where-clause become edges of the query graph. This leads to a non-weighted query graph in a first instance. After that, each join edge is optimized individually with the standard optimization algorithm. Now we have a weighted query graph where each edge is annotated with a cost. After that, all edges are processed in ascending cost order. This processing can be done in three different ways:

1. In the first algorithm (ACO), all spanning edges are simply ordered by their cost, the cheapest edge first, the most expensive edge last. An executable plan is generated by successively nesting join operators for the predicates. After that selection predicates are translated into filter operators and embedded into the plan. Finally all non-spanning edges are translated into filter operators and nested in the result plan.
2. In the second algorithm (QGD), the spanning tree is decomposed into smaller tree components that can be handled by the standard optimization algorithm. Therefore, a component should include maximally 10 edges. In a final step the optimized components are merged again. A component represents a connected sub graph of the query graph and contains a sub plan. In a first instance, the components are optimized individually with the standard optimization algorithm. This leads to a number of disjoint query graph components and a number of remaining edges as an intermediate result. Then the disjoint components are successively merged. This is done by nesting their sub plans with join operators in ascending cost order (cost of the individually optimized predicates). Finally all non-spanning edges are translated into filter operators and are embedded into the result plan.
3. The third algorithm (QGDM) is almost identical to the second algorithm. Except the last steps. Instead of embedding the components sub plans in ascending cost order, the sub plans are executed and their results are stored in the database as temporary relations in order to have a better cost estimation for the further optimization proceeding. Then the remaining edges are evaluated on the temporary relations. This is a kind of intermediate execution. If the number of remaining edges is ≤ 10 , the standard optimization algorithm is applied on the temporary relations and remaining edges. If it is > 10 , the algorithm is invoked recursively (on the temporary relations and remaining edges).

A number of tests were applied to the three introduced algorithms. In the INP benchmark (Increasing Number of Predicates) the number of join predicates was successively increased. For this purpose 6 test databases were created (containing 12, 20, 40, 80, 120 and 160 relations). Each table contains just a single attribute of type integer that contains consecutive numbers from 1 to n (n is the relation's cardinality). Each database contains relations of class S, M, L, and XL, similar to the benchmark in [SMK97]. In each database 25 % of the relations are of class S, 25 % of type M, 25 % of type L and 25 % of type XL (see Table 3.1). For example, in the test database "testdb160" there are 40 relations of type S, 40 relations of type M, 40 relations of type L and 40 relations of type XL.

On 25 % of all relations an index was created on its single attribute in order to facilitate read access.

Class	Relation Cardinality	Domain Size	Percentage %
S	100	100	25
M	1000	1000	25
L	10000	10000	25
XL	100000	100000	25
			Σ 100

Table 3.1: Database setup for testing increasing number of predicates

Randomly generated join queries over all relations of each database were created with a query generator. For example, if the database contains 120 relations, each query has 119 predicates (119 predicates join all 120 relations of the database). Each measured time value is an average value of 30 randomly generated queries in order to balance both expensive and cheap queries. Optimization time and execution time were measured separately. The processing time is defined as the sum of optimization time and execution time.

Figure 3.1 and Table 3.2 show the resulting Query Processing time related to the number of joined relations.

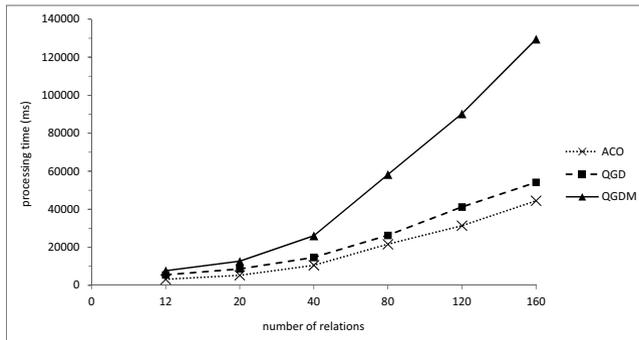


Figure 3.1: Query Processing time related to the number of joined relations

Number Relations	ACO	QGD	QGDM
12	3134,5333	5462,3967	7697,4667
20	5257,3333	8594,13	12649,4
40	10527,2833	14680,9	26058,9
80	21604,058	26289,4	58287,1
120	31344,317	41198,9	90245,2
160	44479,55	54183,2	129512,375

Table 3.2: Query Processing time related to the number of joined relations

4 Conclusion and Future Work

The objective of this thesis was to develop and implement a strategy to extend the SECONDO optimizer in such a way as to enable it to produce preferably fast executable plans in reasonable time for conjunctive queries containing 50 predicates or more. As a result three graph based algorithms (ACO, QGD and QGDM) for optimizing queries with large sets of predicates were developed, introduced, and compared. Each of the algorithms follows a different optimization strategy, but all three combine cost based and heuristic optimization and make use of Kruskal's minimum spanning tree algorithm. The ACO algorithm optimizes each predicate individually, sorts the predicates by their costs and generates a plan by nesting executable plan operators for each predicate in ascending cost order. The QGD algorithm decomposes the query graph into smaller components that can be optimized by the SECONDO standard optimization algorithm. The QGDM works nearly identical to the QGD algorithm, but QGDM additionally materializes the components as temporary relations in the database in order to have a better cost estimation for the further optimization process. A number of benchmark tests was applied to all algorithms. For example, in the INP benchmark (Increasing Number of Predicates) the number of join predicates was successively increased. In the OPT (Optimality) test the three algorithms were compared to the SECONDO standard optimization algorithm. In the XXL test extremely large relations (containing 1000000 tuples) were joined by a query. In all of these tests optimization time, execution time and total query processing time (= optimization time + execution time) was measured separately for each of the three algorithms and then the times were compared. All three algorithms are capable to optimize predicate sets of unlimited size. There is no theoretical upper limit for the number of predicates.

One can see from the test results that there is an approximately linear relationship between the number of predicates and the optimization time. This could be shown for all three algorithms. The ACO algorithm turned out to be remarkably fast in the INP benchmark. In one test case queries with more than 150 predicates were optimized and executed. Such queries can be optimized in about one second and executed in about 40 seconds with the ACO algorithm. In the OPT test all three algorithms were compared with the the SECONDO standard optimization algorithm. For this, the number of predicates in the query was reduced to 10. As expected, the standard optimization algorithm generates the best plans (which means the execution time is the shortest one). But when regarding the total query processing time, the QGD algorithm is even faster than the standard optimization algorithm. It could also be shown by the test results that the decomposition of the query graph (and thus the enormous reduction of the search space for finding the optimal predicate order) does not have a huge influence on the quality of the generated executable plan. The optimality test showed that the QGD algorithm produces executable plans, which have just a slightly longer execution time than those of the standard optimization algorithm even though the query graph was decomposed into at least three individually optimized components.

The presented test results can serve as suggestions for deciding, which algorithm should be applied to a given query. A general statement, which algorithm is the best cannot be made. But if a great number of extremely large relations have to be joined, either the QGDM or

the QGD algorithm should be used. In every other case the ACO algorithm should be used. In future work a function should be developed that in a kind of preprocessing analyzes a query and decides, which algorithm has to be applied for optimizing this query. Future work could also follow completely different approaches (for example genetic algorithms or randomized methods) for optimizing queries with large sets of predicates.

References

- [Feg98] Leonidas Fegaras. A new heuristic for optimizing large queries. In Gerald Quirchmayr, Erich Schweighofer, and Trevor Bench-Capon, editors, *Database and Expert Systems Applications*, volume 1460 of *Lecture Notes in Computer Science*, pages 726–735. Springer Berlin / Heidelberg, 1998. 10.1007/BFb0054528.
- [GBT04] SECONDO: An Extensible DBMS Architecture and Prototype. Technical report, Fernuniversität in Hagen, 3 2004.
- [GdAA⁺05] Ralf Hartmut Güting, Victor Teixeira de Almeida, Dirk Ansorge, Thomas Behr, Zhiming Ding, Thomas Höse, Frank Hoffmann, Markus Spiekermann, and Ulrich Telle. SECONDO: An Extensible DBMS Platform for Research Prototyping and Teaching. In *Proceedings of the 21st International Conference on Data Engineering*, pages 1115–1116, 2005.
- [IK90] Y. E. Ioannidis and Younkyung Kang. Randomized algorithms for optimizing large join queries. *SIGMOD Rec.*, 19:312–321, May 1990.
- [Ioa97] Yannis E. Ioannidis. Query Optimization. In *The Computer Science and Engineering Handbook*, pages 1038–1057. 1997.
- [JK84] Matthias Jarke and Jürgen Koch. Query Optimization in Database Systems. *ACM Comput. Surv.*, 16:111–152, June 1984.
- [KRB85] Won Kim, David S. Reiner, and Don S. Batory, editors. *Query Processing in Database Systems*. Springer, 1985.
- [Kru56] Joseph B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, February 1956.
- [Neu09] Thomas Neumann. Query Simplification: Graceful Degradation for Join-Order Optimization. In Carsten Binning and Benoit Dageville, editors, *SIGMOD-PODS'09 : Compilation Proceedings of the International Conference on Management of Data & 28th Symposium on Principles of Database Systems*, pages 403–414, Providence, USA, June 2009. Association for Computing Machinery (ACM), ACM.
- [SEC12] http://dna.fernuni-hagen.de/Secondo.html/content_home.html, 2012.
- [SKS96] Abraham Silberschatz, Henry Korth, and S. Sudarshan. *Database Systems Concepts*. McGraw-Hill, Inc., New York, NY, USA, 3 edition, 1996.
- [SMK97] Michael Steinbrunn, Guido Moerkotte, and Alfons Kemper. Heuristic and randomized optimization for the join ordering problem. *The VLDB Journal*, 6:191–208, 1997. 10.1007/s007780050040.

Modellbasierte Hardwareentwicklung am Beispiel eingebetteter Prozessoren für die optische Messtechnik

Sebastian Berschneider

sebastian.berschneider@informatik.uni-erlangen.de

Abstract: Die Entwickler von eingebetteten Systemen sehen sich immer schwierigeren Anforderungen, wie steigende Leistungsfähigkeit und Entwicklungs-Komplexität, ausgesetzt. Besonders Änderungen während der Design-Entwicklung fallen sehr stark ins Gewicht, da deren Umsetzung und Verifizierung viel Zeit benötigt.

In dieser Arbeit wurde eine Möglichkeit entwickelt, den Entwicklern bei solchen Projekten zu unterstützen. Durch die Plattform-Simulation OVP kann mit Hilfe von Prozessor- und Peripheral-Modellen eine funktionale Simulation der Hardware durchgeführt werden. Ziel der Arbeit war es, dem Anwender zu ermöglichen, mit so wenig Anpassungen wie möglich eine automatische Simulation von mehreren Architekturen mit unterschiedlicher Prozessor-Anzahl und Kommunikations-Möglichkeiten durchzuführen. Anschließend wurde ein mathematisches Modell erstellt, mit dessen Hilfe und den Messungen der Simulationen Zeitabschätzungen für die entwickelte Anwendung berechnet werden können.

1 Einleitung

Aufgrund der immer weiter steigenden Leistung und kleineren Fertigungstechniken können eingebettete Systeme mittlerweile Aufgaben übernehmen, die vor einigen Jahren noch großen Desktop-Computern mit erheblich höherem Platzbedarf und Stromverbrauch vorbehalten waren, wie z.B. bei der optischen Messtechnik [GB03]. Hierbei werden Bilddaten automatisch gelesen, analysiert und verarbeitet. Besonders bei der Qualitätssicherung sind solche Verfahren immer wichtiger geworden. Die zu analysierenden Bilder werden dabei von einem Sensor eingefangen und anschließend weiterverarbeitet. Für die Weiterverarbeitung der Bilddaten ist ein leistungsstarkes System notwendig. Ein stationärer Rechner hat dabei den großen Nachteil, dass eine sehr große Bandbreite zwischen dem Sensor und dem stationären Rechner notwendig ist. Aus diesen Performance-Gründen befinden sich die Hardware-Komponenten, welche die Berechnungen durchführen, nah am Sensor, wie es bei den *Smart Camera*-Systemen der Fall ist. Diese nehmen die Bilddaten durch ein Objektiv auf und können sie anschließend durch einen eingebauten Chip, z.B. FPGA oder Prozessor, weiterverarbeiten, ohne dass ein stationärer Computer nötig wäre [WOL02]. Die Umsetzung der Berechnungen erfolgt beispielsweise durch den Einsatz von Stencil-Codes [RSLF11].

Um den steigenden Anforderungen der Anwendungen im Bereich der eingebetteten Systeme, wie bspw. in der optischen Messtechnik, gerecht zu werden, geht die Entwicklung

immer mehr in Richtung der Multicore-Prozessoren. Durch deren Einsatz kann die Leistungsaufnahme und dadurch der Stromverbrauch gesenkt, aber auch die Effizienz der Anwendungen gesteigert werden [MGRR11]. Die Entwicklung durch Hardware-Software-Codesign stellt sich besonders im Falle von Multicore-Architekturen als schwierig heraus, da nicht nur eine parallele Anwendung entwickelt werden muss, sondern auch eine mit dieser Anwendung im Einklang funktionierende parallele Architektur. Daher sind eine Weiterentwicklung der Design-Tools sowie vor allem der Herangehensweise und des Ablaufs der Design-Entwicklung erforderlich [NLC09, Bai08]. Für den Entwickler wäre es hilfreich, wenn durch Modellierungen und Simulationen vorher mögliche Schwachstellen und Ineffizienzen bei Anwendung und Architektur erkannt und ausgebessert werden könnten.

Eine Möglichkeit wäre die Verwendung einer Hardware-Beschreibungssprache wie VHDL, die sehr genaue Zeitabschätzungen ermöglicht. Erweiterungen an der Simulations-Architektur, z.B. die Evaluierung mit zusätzlichen Prozessoren, sind allerdings mit einem erheblichen Mehraufwand verbunden, da diese auf dem *Register Transfer Level* (RTL), also Takt-genau, erstellt werden müssen, um mit dem restlichen VHDL-Entwurf zusammenarbeiten zu können. Vergleiche zwischen mehreren möglichen Architekturen (z.B. unterschiedliche Prozessor-Anzahl) fallen somit sehr schwer. Dazu kommt noch die vor allem bei komplexen Multicore-Architekturen sehr langsam ablaufende Simulation.

Um diese Nachteile so weit wie möglich zu minimieren, setzt man vermehrt auf eine simulationsbasierte Entwicklung, wodurch Anwendung und Architektur Anpassungen mit einfachen Mitteln in Software umgesetzt werden können. SystemC und das dazugehörige TLM 2.0 wurden zum Quasi-Standard. Aufgrund der hardwarenahen Simulation können relativ genaue Zeitangaben unter Einbeziehung der Hardware-spezifischen Besonderheiten erzielt werden. Diese Simulation läuft dabei deutlich schneller ab. Die Geschwindigkeit reichte aber vor allem bei komplexen Systemen immer noch nicht aus, um sie für die Anwendungs-Entwicklung und -Verifikation für die gewünschte Zielarchitektur verwenden zu können.

2 Problemstellung und Lösungsansätze

Diesem Problem nimmt sich die Simulations-Plattform *Open Virtual Platform* (OVP) an, welche eine wesentlich höhere Flexibilität und Geschwindigkeit als SystemC [Bai08] bietet. Es existieren bereits viele Modelle von Prozessoren im Bereich der eingebetteten Systeme wie z.B. ARM oder OpenRISC. Der Anwender kann mit wenig Aufwand eine Architektur mit den gewünschten Prozessoren-, Speicher- und Peripherie-Modellen aufbauen. Aufgrund der funktionalen Simulation können im Gegensatz zur Simulation mit VHDL allerdings keine genauen zeitlichen Abschätzungen der Architektur vorgenommen werden.

Ein interessanter Vertreter in den eingebetteten Systemen ist der Leon3-Prozessor. Dieser genießt besonders im universitären Bereich große Verbreitung, da er durch den mitgelieferten VHDL-Quellcode frei konfiguriert, verändert und erweitert werden kann. Ein

Beispiel für eine solche Erweiterung des Prozessors stellt die Registerkopplung als alternative Lösung für den Datenaustausch in Multicore-Architekturen dar. Diese soll einen Geschwindigkeitsvorteil gegenüber des Datenaustausches über gemeinsamen Speicher bringen. Dabei existieren mehrere globale Registerfiles, auf welche die Prozessoren lesend oder schreibend zugreifen können. Die Zugriffsberechtigungen können dabei statisch verändert werden. Diese Leon3-Erweiterung wurde bereits für zwei Leon3-Prozessoren mit zwei Registerfiles realisiert (siehe Abbildung 1) [KP12]. Ein solch flexibler Prozessor wie der Leon3 wäre als Prozessor-Modell in einer Simulations-Umgebung wie OVP eine große Hilfe für den System-Entwickler, da dieser nicht nur theoretisch im Modell, sondern auch praktisch als Hardware in VHDL angepasst werden kann, wie das Beispiel der Registerkopplung zeigt.

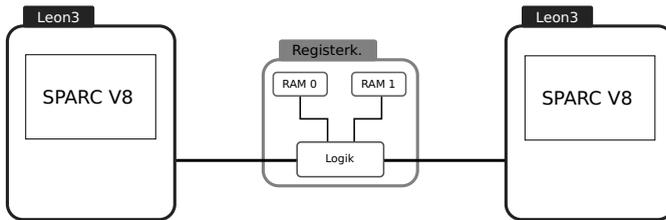


Abbildung 1: Verbindung der Registerkopplung mit zwei Leon3-Prozessoren

Da weder vom Leon3-Prozessor, noch von der Registerkopplung ein OVP-Modell existierte, wurden diese neu entwickelt. Dabei wurde auf einen möglichst abstrakten und generalisierten Entwurf geachtet. So besteht die Möglichkeit, das Registerkopplungs-Modell im Gegensatz zum VHDL-Entwurf mit mehr als zwei Registerfiles auszustatten und eine beliebige Anzahl an Prozessoren mit ihr zu verbinden. Verschiedene Parameter können bei den Modellen verändert werden, u.a. Anzahl der Prozessoren, Art des Datenaustausches, Anzahl der Register in der Registerkopplung und Topologie der Registerkopplung. Durch die Auslagerung der Parameter in eine Konfiguration kann der Anwender sie schnell und ohne Eingriff in den Quellcode der Modelle bearbeiten und dadurch seine gewünschte Architektur zusammenstellen und simulieren lassen. Eine erstellte Anwendung kann hierdurch ebenfalls automatisch auf mehreren Architekturen mit unterschiedlichen Parametern ausgeführt und die Ergebnisse evaluiert werden, ohne dass ein weiteres Eingreifen durch den Anwender von Nöten ist. Dadurch können eventuelle Engpässe und Fehler schneller erkannt und Verbesserungsansätze für die Architektur oder die Software gefunden und umgesetzt werden.

Da es sich bei OVP um eine funktionale Simulation handelt, sind zeitgenaue Messungen sehr schwierig. Um diesem Problem entgegenzutreten wurde ein einfaches mathematisches Modell erstellt, das mit Hilfe der messbaren Ergebnissen der Simulation (Parallelitätsgrad, ausgeführte Instruktionen, Anzahl der Speicherzugriffe) die Laufzeit der simulierten Architektur abschätzen kann.

3 Implementierung und Ergebnisse

Die entwickelten Modelle und Analyse-Methoden wurden durch den Einsatz eines Stencil-Codes evaluiert. Anhand der Zeitabschätzungen des mathematischen Modells kann erkannt werden, ob bei der gegebenen Architektur mit den gewählten Konstanten eine Bandbreiten- oder eine Leistungs-Beschränkung vorliegt. Auch der optimale Parallelitätsgrad lässt sich damit herausfinden. Außerdem wurde der Speedup der Registerkopplungs-Variante gegenüber der *shared memory*-Variante abgeschätzt. Hierbei konnte beobachtet werden, dass der Speedup abhängig von der gewählten Bildgröße ist. Bei einer Anwendung mit viel Datenaustausch bringt die Registerkopplung mit steigender Prozessor-Anzahl einen Speedup von bis zu 6 Prozent. Bei Anwendungen mit nur wenig Datenaustausch kann es dagegen beim Einsatz der Registerkopplung zu Performance-Einbrüchen kommen.

Ziel dieser Arbeit war es, ein generalisiertes Modell des Leon3 und der Registerkopplung als alternative Datenaustauschmethode zu erstellen, um dem Anwender eine einfache Möglichkeit an die Hand zu geben, Anwendungen schnell und mit so wenig Anpassungen wie möglich auf mehreren Multikern-Architekturen zu analysieren. Durch ein mathematisches Modell sind trotz rein funktionaler Simulation ungefähre Zeit-Abschätzungen möglich. Als weitere Untersuchung wäre der Aufbau und Simulation von heterogenen Systemen interessant. Eine Verbesserung der Zeitabschätzungen für die simulierten Architekturen wäre ebenfalls erstrebenswert. Hierbei müsste ein Kompromiss zwischen Simulationsgenauigkeit und -geschwindigkeit gefunden werden. Wünschenswert wäre der Aufbau einer „abstrakten Maschine“, die als Eingabe einen Algorithmus erhält und nach einiger Simulationsarbeit automatisch ausgibt, welcher Architektur-Aufbau am Besten für diesen Algorithmus geeignet wäre, wie z.B. homogen/heterogen, Art der Prozessoren, Anzahl der Prozessoren, Art des Datenaustausches und ggf. Registerkopplungs-Struktur.

Literatur

- [Bai08] Brian Bailey. System Level Virtual Prototyping becomes a reality with OVP donation from Imperas. *White Paper - System Level Virtual Prototyping and OVP*, 2008.
- [GB03] Jack Ganssle und Michael Barr. *Embedded Systems Dictionary*. Mcgraw-Hill Professional, 2003.
- [KP12] Jochen Keil und Benjamin Pfundt. Registerkopplung für den Leon3-Prozessor, 2012.
- [MGRR11] A. Munir, A. Gordon-Ross und S. Ranka. A Queueing Theoretic Approach for Performance Evaluation of Low-Power Multi-core Embedded Systems. *IEEE*, 2011.
- [NLC09] Julian Nita, Vasile Lazarescu und Rodica Constantinescu. A New Hw/Sw Co-Design Method For Multiprocessor System On Chip Applications. *IEEE*, 2009.
- [RSLF11] Marc Reichenbach, Michael Schmidt, Andreas Loos und Dietmar Fey. A Smart Camera Processing Pipeline for Image Applications utilizing Marching Pixels. *SIPIJ*, 2011.
- [WOL02] Wayne Wolf, Burak Ozer und Tiehian Lv. Smart Cameras as Embedded Systems. *IEEE*, 2002.

SLOTH ON TIME: Effiziente Zeitsteuerung in einem unterbrechungsgesteuerten Echtzeitbetriebssystem

Daniel Danner
danner@cs.fau.de

Diplomarbeit im Fach Informatik
Friedrich-Alexander-Universität Erlangen-Nürnberg

Abstract:

In Standardimplementierungen zeitgesteuerter Echtzeitbetriebssysteme wird der statisch vorkonfigurierte Ablaufplan üblicherweise auf einen einzelnen Hardware-Zeitgeber abgebildet, welcher die CPU zum Zeitpunkt einer geplanten Einlastung unterbricht. Anhand der Ablaufabelle programmiert das Betriebssystem den Zeitgeber neu und lastet den aktuellen Task ein. Im Rahmen dieser Arbeit wurde untersucht, inwiefern der damit verbundene Rechenaufwand durch den Einsatz von Zeitgebereinheiten minimiert werden kann, die auf modernen 32-Bit-Mikrocontrollerplattformen in großer Anzahl verfügbar sind. Es wird gezeigt, wie es durch die geschickte Konfiguration dieser Einheiten möglich ist, die Verantwortung für die Abarbeitung eines Zeitplans an die Peripherie auszulagern, und dadurch die Latenzen zeitgesteuerter Einlastung stark zu reduzieren. Die Evaluierung der Referenzimplementierung für die Plattform Infineon TriCore TC1796 im Vergleich mit kommerziell angebotenen Systemen zeigte neben einem vorteilhaftem Laufzeitverhalten durch Vermeidung von Prioritätskehr und unnötigen Unterbrechungen eine durchwegs verbesserte Leistung gegenüber den Vergleichssystemen um bis zu dreistellige Faktoren.

1 Hintergrund: Das SLOTH-Konzept

In der Entwicklung von Betriebssystemen ist es stets ein wesentliches Ziel, die durch den Kern verursachten Kosten so klein wie möglich zu halten. Insbesondere im Bereich der eingebetteten Echtzeitsysteme gilt es, neben einem möglichst geringen Bedarf an RAM und ROM, die Ereignislatenzen kurz zu halten und Anwendungen nicht unnötig zur Wahrnehmung von Verwaltungsaufgaben zu unterbrechen.

In bisherigen Arbeiten am SLOTH-Echtzeitkern wurde ein System vorgestellt, welches die Unterscheidung zwischen durch Hardware ausgelösten Interrupt Service Routines und in Software verwalteten Threads aufhebt, und sämtliche Kontrollflüsse als Unterbrechungsbehandlungen modelliert [HLSSP09, HLSP11]. Die in traditionellen Systemen problematische Zweiteilung des Prioritätenraums entfällt, und die bisher in Software getroffenen Einplanungsentscheidungen werden an den Interrupt-Controller ausgelagert, wodurch die Latenzen bei Einlastung und Terminierung von Threads drastisch reduziert werden.

Ermöglicht wird dieser Entwurf durch mit vielen Prioritätsstufen ausgestattete Interrupt-

Controller, die auf vielen modernen Plattformen verfügbar sind. Da sich SLOTH bisher auf die Domäne der ereignisgesteuerten Echtzeitsysteme beschränkte, wurde im Rahmen dieser Arbeit untersucht, ob sich diese Grundidee der geschickten Ausnutzung vorhandener Hardwarekomponenten zur Entlastung des Betriebssystems auch auf zeitgesteuerte Systeme übertragen lässt. In Anlehnung an den von SLOTH implementierten OSEK-Standard wurden hierfür die Modelle der verwandten Standards OSEKtime und AUTOSAR OS herangezogen [AUT10, OSE01]. Die Ergebnisse der Arbeit und das entstandene System mit dem Namen SLOTH ON TIME wurden auf dem 33. Real-Time Systems Symposium (RTSS '12) vorgestellt [HDM⁺12].

2 Entwurf und Implementierung auf dem TriCore TC1796

Um das Ziel der möglichst geringen Verwaltungskosten zur Laufzeit zu erreichen, basiert der Entwurf von SLOTH ON TIME darauf, bereits zur Kompilierzeit die statisch bekannten Anforderungen der Anwendung den Hardwarekomponenten zuzuordnen, welche dann in einer Initialisierungsphase entsprechend ihren spezifischen Aufgaben konfiguriert werden. Zur Laufzeit werden die Ablauftabellen der Anwendung schließlich ohne weitere, verwaltende Softwareeingriffe umgesetzt.

Auf der für die Referenzimplementierung gewählten Mikrocontrollerplattform Infineon TriCore TC1796 bietet das Zeitgebermodul *General Purpose Timer Array* eine Vielzahl an Timerzellen, die individuell für die periodische Auslösung eines bestimmten Interrupts programmiert werden können. So ist es möglich, im Rahmen einer statischen Analyse die in der Anwendung festgelegten Ablauftabellen mit ihren Aktivierungspunkten und Zeitschranken jeweils einer Timerzelle zuzuordnen und diese entsprechend der Periode und des Offsets während der Initialisierung zu programmieren. Mit Ausnahme von Umschaltungen des Betriebsmodus findet das zeitgesteuerte Abarbeiten der Ablauftabelle dann vollständig ohne Neuprogrammierung der Zeitgeber zur Laufzeit statt.

Im Unterschied zur rein prioritätsabhängigen Einplanung in ereignisgesteuerten OSEK-Systemen ist in OSEKtime eine stapelbasierte Ablaufplanung vorgesehen. Dies bedeutet, dass ein neu aktivierter Task stets den bereits laufenden unterbricht, welcher wiederum erst bei Terminierung des neuen Tasks fortgesetzt wird. In SLOTH ON TIME wird dieses Verhalten erreicht, indem die Unterbrechungsbehandlung eines zeitgesteuerten Tasks unmittelbar vor dem Sprung in den Anwendungscode die Ausführungspriorität der CPU auf eine niedrigere, allen zeitgesteuerten Tasks gemeinsame Ebene abgesenkt wird. Stellt man nun sicher, dass die Unterbrechungspriorität, auf welcher neue Tasks ausgelöst werden, höher eingestellt ist als diese Ausführungspriorität, ist eine Verdrängung des laufenden Tasks durch einen neuen Tasks garantiert. Auch für den von der Spezifikation vorgesehenen Mischbetrieb eines OSEK-Kerns unterhalb eines OSEKtime-Systems eignet sich diese Aufteilung des Prioritätsraums. Es muss lediglich dafür gesorgt werden, dass sämtliche Aktivitäten des OSEK-Kerns – also sowohl Tasks als auch Kernsynchronisation – auf Prioritäten unterhalb der OSEKtime-Ausführungspriorität stattfinden.

3 Evaluation

Zur Bewertung von SLOTH ON TIME wurde die Referenzimplementierung sowohl einer qualitativen als auch einer quantitativen Evaluation unterzogen und dabei mit zwei kommerziell angebotenen OSEKtime- beziehungsweise AUTOSAR-Systemen verglichen. Die quantitativen Messungen wurden mit Hilfe einer PowerTrace-Einheit von Lauterbach durchgeführt und beinhalten unter anderem die Kosten (in Zyklen) der Einlastung und Terminierung zeitgesteuerter Tasks in unterschiedlichen Ausführungsszenarien. Die Ergebnisse zeigen signifikante Zeitvorteile gegenüber traditionellen Implementierungen zeitgesteuerter Echtzeitsysteme in Software. SLOTH ON TIME erzielt Beschleunigungsfaktoren im Bereich von 1,1 bis 8,6 im Vergleich zum OSEKtime-Vergleichssystem sowie zwischen 6,0 und 171,4 gegenüber dem AUTOSAR-Vergleichssystem.

Untersucht man das Laufzeitverhalten der Systeme in qualitativer Hinsicht, offenbaren sich weitere Vorteile des Entwurfs von SLOTH ON TIME im Zusammenhang mit unnötigen Unterbrechungen des Anwendungsprogramms und des Phänomens der Prioritätsinversion. Eine solche Situation tritt zum Beispiel bei zeitgesteuerten Aktivierungen niedrigerer Tasks in einem AUTOSAR-System auf. Wie in Abbildung 1 sichtbar, wird im Vergleichssystem ein hochpriorer Task („Task3“) zwei Mal durch niedripriore Aktivierungen unterbrochen, da die Einlastungsentscheidung und das Vormerken der beiden neuen Tasks zur späteren Einlastung in Software stattfindet. Die Kosten dieser Unterbrechungen belaufen sich auf jeweils 2.075 Zyklen. In SLOTH ON TIME hingegen ist eine solche Prioritätsinversion – also die Unterbrechung einer wichtigen Aktivität im Sinne einer weniger wichtigen – von vorneherein ausgeschlossen, da der Vergleich der Prioritäten beider Tasks nebenläufig im Interrupt-Controller stattfindet.

4 Fazit und Ausblick

Basierend auf dem Entwurf von SLOTH ON TIME gelingt demnach eine Implementierung eines zeitgesteuerten Echtzeitsystems, die durch sehr geringe Verwaltungskosten auffällt und mit einem gegenüber traditionellen Systemen verbesserten Laufzeitverhalten mehr Planbarkeit erlaubt. SLOTH ON TIME kommt dabei ohne speziell angefertigte Peripherie aus und baut allein auf die geschickte Ausnutzung der Zeitgeberhardware, wie sie auf vielen modernen Mikrocontrollerplattformen vorhanden ist.

Als zukünftige Erweiterung von SLOTH ON TIME kommt beispielsweise die Möglichkeit zur flexibleren Kombination des SLOTH-Ansatzes mit traditionellem Timer-Multiplexing in Frage. Damit gäbe es je nach Aufgabe des spezifischen Tasks die Option zur individuellen Abwägung zwischen minimaler Latenz – durch Zuweisung einer dedizierten Timerzelle – und dem Kompromiss aus höherer Latenz bei Nutzung einer Timerzelle für mehrere Aktivierungen. Ein Einsatz von SLOTH ON TIME wäre damit auch in Szenarien denkbar, in welchen die geringe Anzahl verfügbarer Timerzellen auf der Zielplattform eine Zuordnung sämtlicher Aktivierungspunkte auf eigene Zellen unmöglich machen würde.

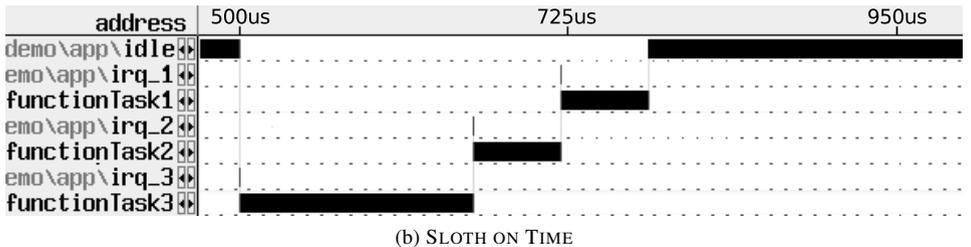
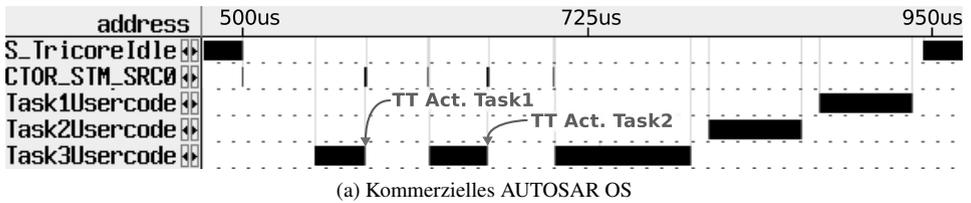


Abbildung 1: Prioritätsinversion in einem kommerziellen AUTOSAR-System (oben) im Vergleich zu SLOTH ON TIME (unten). Der hochpriorie Task3 wird zwei Mal für je 2,075 Zyklen unterbrochen, da die zeitgesteuerten Aktivierungen zweier niederpriorer Tasks jeweils eine neue Einplanungsentscheidung in Software erfordern; bei SLOTH ON TIME wird Task3 nicht unterbrochen.

Literatur

- [AUT10] AUTOSAR. Specification of Operating System (Version 4.1.0). Bericht, Automotive Open System Architecture GbR, Oktober 2010.
- [HDM⁺12] Wanja Hofer, Daniel Danner, Rainer Müller, Fabian Scheler, Wolfgang Schröder-Preikschat und Daniel Lohmann. Sloth on Time: Efficient Hardware-Based Scheduling for Time-Triggered RTOS. In *Proceedings of the 33rd IEEE International Symposium on Real-Time Systems (RTSS '12)*, Seiten 237–247. IEEE Computer Society Press, Dezember 2012.
- [HLSP11] Wanja Hofer, Daniel Lohmann und Wolfgang Schröder-Preikschat. Sleepy Sloth: Threads as Interrupts as Threads. In *Proceedings of the 32nd IEEE International Symposium on Real-Time Systems (RTSS '11)*, Seiten 67–77. IEEE Computer Society Press, Dezember 2011.
- [HLSSP09] Wanja Hofer, Daniel Lohmann, Fabian Scheler und Wolfgang Schröder-Preikschat. Sloth: Threads as Interrupts. In *Proceedings of the 30th IEEE International Symposium on Real-Time Systems (RTSS '09)*, Seiten 204–213. IEEE Computer Society Press, Dezember 2009.
- [OSE01] OSEK/VDX Group. Time Triggered Operating System Specification 1.0. Bericht, OSEK/VDX Group, Juli 2001. <http://portal.osek-vdx.org/files/pdf/specs/ttos10.pdf>.

Entwicklung und Evaluation eines Navigationssystems für Elektrorollstühle

Bachelorarbeit

Dženan Džafić

Dominik Franke

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

Abstract: Für Fahrer von elektrisch betriebenen Fahrzeugen ist eine energieeffiziente Nutzung der Akkukapazität von großer Bedeutung. Ein Navigationssystem, das Routen in Abhängigkeit der Steigung, welche einen bedeutenden Einfluss auf den Verbrauch nimmt, berechnet, bietet einen sinnvollen Beitrag zu einer effizienteren Nutzung der Energie. Insbesondere gilt dies für Fahrer von Elektrorollstühlen, für welche die Nutzung der Akkukapazität eine entscheidende Rolle spielt, da ein Ausfall des Akkus dazu führt, dass der Fahrer auf einem Streckenabschnitt stehen bleibt. Diese Abschlussarbeit beschäftigt sich mit der Entwicklung eines Navigationssystems für Elektrorollstühle, eNav.

1 Einleitung

Energieeffizienz ist in Zeiten steigender Energiepreise und hinsichtlich beschränkter Akkukapazitäten ein hervortretendes Thema. Diese Tatsache betrifft insbesondere Benutzer von Elektrorollstühlen, deren eingeschränkte Mobilität durch verlängerte Akkulaufzeiten teilweise kompensiert werden kann. Zahlreiche Variablen beeinflussen den Verbrauch. Darunter befinden sich beispielsweise Steigung, Fahrbahnbelag sowie Temperatur. Diese Arbeit konzentriert sich ausschließlich auf die Berücksichtigung der Streckensteigung und erstellt eine Verbrauchsfunktion unter Einbeziehung von Steigungsinformationen. Das Navigationssystem eNav nutzt die Verbrauchsfunktion zusammen mit einer Modifikation des A*-Suchalgorithmus. Energieeffizientes Routing wird ermöglicht, indem eNav den Fahrer eines Elektrorollstuhls durch die Berechnung einer energieeffizienten Route assistiert [FDWK11a].

In der Regel bieten Navigationssysteme eine Routenberechnung für den kürzesten oder schnellsten Weg an. Dies beruht darauf, dass eine solche Berechnung für die meisten Anwender ausreichend ist und gewöhnlich keine Steigungsinformationen im Kartenmaterial vorhanden sind. Gleiches gilt bisher für das Routing auf dem Kartenmaterial von OpenStreetMap (OSM), welches frei verfügbar ist und durch eine aktive Community kontinuierlich erweitert wird. Eine Ausnahme bezüglich der Verfügbarkeit von Höheninformationen sind die OSM-Daten um die Städte Bonn und Heidelberg, welche durchaus viele Höheninformationen enthalten. Der Grund hierfür liegt in Universitätsaktivitäten in dem

Bereich OSM, wodurch ein abgewandelter A*-Stern Algorithmus, der eNav Algorithmus, über die Schnittstelle von OpenRouteService (ORS) von der Universität in Heidelberg in der Realität auf den Daten dieser Städte angewendet werden kann. Dazu nutzt das Navigationssystem eNav die ermittelte Verbrauchsfunktion und berechnet Routen, welche unter der Berücksichtigung der Steigung der Streckenabschnitte so konstituiert sind, dass der Stromverbrauch minimiert wird. Die Berechnung bedient sich dabei einiger weiterer Parameter, wie Normalverbrauch und maximal passierbare Steigung, die dem Handbuch eines Elektrorollstuhls entnommen werden können.

2 Grundlagen

OpenStreetMap (OSM) ist ein globales Projekt, gegründet im Jahr 2004, welches das Ziel verfolgt, eine freie Weltkarte in digitaler Form zu erschaffen [RT10]. Die Benutzer von OSM tragen aktiv durch Änderungen und Erweiterungen des Kartenmaterials dazu bei, dass eine hohe Datendichte erreicht und eine verlässliche Datengrundlage geschaffen wird. Sie fügen zum Kartenmaterial hauptsächlich neue Straßen, Gebäude und Points of Interest (POI) hinzu. Zusätzlich werden weitere Informationen gesammelt, beispielsweise eine Klassifizierung des Straßenbelags sowie Steigungs- und Höheninformationen. Stufen und Hindernisse können ebenfalls eingetragen werden, wodurch bestimmte Barrieren z.B. von Fahrrad- und Rollstuhlfahrern umfahren werden können. OSM bietet eine gute Grundlage für ein Navigationssystem, da sich die Daten problemlos um weitere Informationen durch den Gebrauch von *Tags* anreichern lassen [RT10].

OpenRouteService (ORS) ist ein Webservice, der OSM-Daten als Kartenmaterial für die Berechnung von Routen verwendet. Eine Routenberechnung geschieht dabei unter der Verwendung des A*-Algorithmus. Die Benutzer können den Dienst von ORS auf zwei unterschiedliche Arten nutzen: Zum Einen über ein Webinterface, zum Anderen über direkte Anfragen an die Webservice API. Das Navigationssystem eNav nutzt ORS, indem es die eigenen Routenanfragen an den Webservice weiterleitet, um die Berechnung der effizientesten Route zu starten.

Der A*-Suchalgorithmus berechnet den kürzesten Pfad zwischen zwei Knoten A und B. Als Eingabe erhält der Algorithmus einen Graphen, welche eine Abstraktion eines Straßennetzwerks darstellt. Abbildung 1 visualisiert auf der linken Seite einen Beispielgraphen mit der Distanz als Kantengewicht. Anhand dieser Grundlage und einer Terminierungsstrategie in Form einer heuristischen Funktion berechnet der Algorithmus über die Kantengewichte eine optimale Lösung, falls eine solche existiert. eNav verwendet zur Berechnung der energieeffizientesten Route eine modifizierte Variante des A*-Algorithmus [FDBK12].

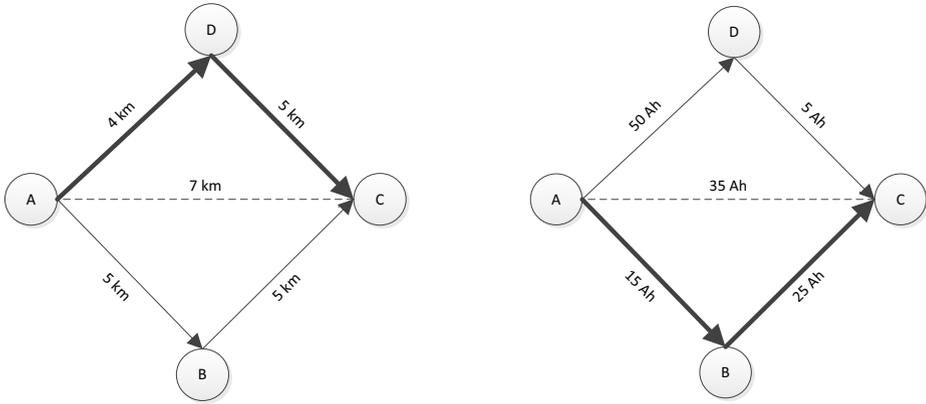


Abbildung 1: Beispielgraphen

3 Das Navigationssystem eNav

Das Navigationssystem eNav basiert auf der Android Applikation AndNav2. Diese Software ist frei über den Android Market erhältlich und wird für die Zwecke von eNav angepasst. Zusätzlich zu den üblichen Eingabeparametern, wie Start- und Zieladresse, werden der Normalverbrauch, die maximal passierbare Steigung des Elektrorollstuhls und die Höchstgeschwindigkeit benötigt. Der Server von ORS wird mit den Parametern bedient und implementiert die Modifikation des A*-Algorithmus. Bei der Integration der Änderungen wird im A*-Algorithmus die Straßenlänge durch den Energieverbrauch ersetzt und ermöglicht dadurch eine Berechnung der effizientesten Route. Abbildung 1 visualisiert auf der rechten Seite einen Graphen mit Verbrauch als Kantengewicht. Der Verbrauch wird über die folgende Verbrauchsfunktion berechnet:

$$f(x) := \begin{cases} \text{Verbrauch} := \text{Distanz} \cdot \text{Normalverbrauch} \cdot 0.05, & \text{falls Steigung} < -0.5, \\ \text{Verbrauch} := \text{Distanz} \cdot \text{Normalverbrauch}, & \text{falls } -0.5 < \text{Steigung} < 0, \\ \text{Verbrauch} := \text{Distanz} \cdot \text{Normalverbrauch} \cdot 1,15^{\text{Steigung}}, & \text{sonst} \end{cases}$$

Der Algorithmus findet in dem angegebenen Graphen den effizientesten Weg von A nach C über B mit einem Verbrauch von 37,5 Ah. Auffällig ist, dass der effizienteste Weg nicht gleich dem kürzesten (A über D nach C 62,5 Ah) Weg ist. Diese Tatsache ist häufig zu beobachten [FDWK11].

Die Interaktion zwischen den Akteuren bei der eNav-Navigationslösung wird durch das Sequenzdiagramm in Abbildung 3 abstrakt dargestellt. Die Routenanfrage wird mit einem Start- und Zielknoten an den Server von ORS weitergeleitet. ORS extrahiert diese Parameter aus der von eNav generierten URL und initiiert eine Routenberechnung, indem es auf das Kartenmaterial von OSM zurückgreift. Die Berechnung liefert als Ergebnis sowohl eine effizienteste Route bezüglich der Steigung als auch eine kürzeste Route. Der Benutzer von eNav soll auch weiterhin die Wahl haben, welche Route für ihn angemessener ist.

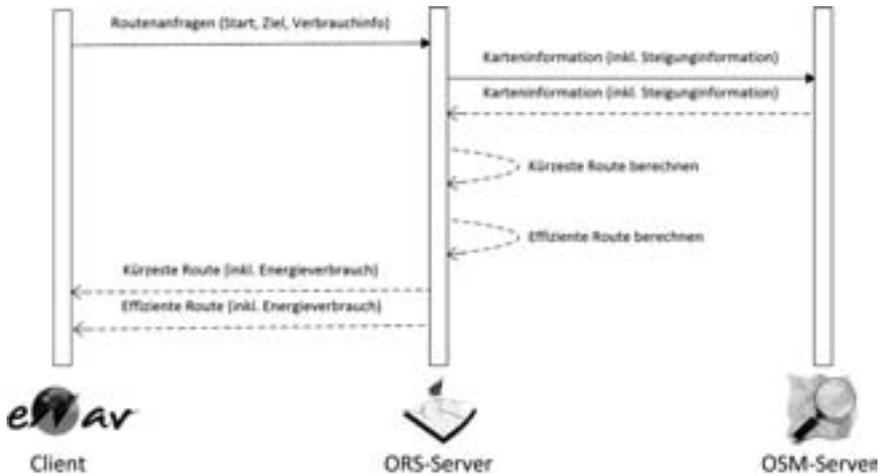


Abbildung 2: eNav Sequenzdiagramm

4 Fazit

Das eNav-System nutzt hauptsächlich Daten, die frei in OSM, rollstuhlrouting.de und wheelmap.org verfügbar sind. Es handelt sich dabei um Informationen über Straßen, Streckensteigung, Höhenangaben, Bodenbeläge und Zugänglichkeit. Eine Einschränkung besteht darin, dass diese Informationen (insbesondere Höhen- und Bodenbeläge) nicht vollständig verfügbar sind. Diese Tatsache führt dazu, dass die Qualität der Routenberechnung sinkt, da z.B. Höheninformationen auf der gesuchten Strecke nur teilweise verfügbar sind. Eine genauere Untersuchung der Verbrauchsfunktion und das Aufdecken von neuen Datenquellen stehen für künftige Arbeiten in diesem Projekt im Vordergrund. Zusätzlich werden weitere Parameter für die Routenberechnung berücksichtigt, wie beispielsweise der Fahrbahnbelag, die Temperatur und der Reifendruck.

Literatur

- [FDBK12] Franke, D.; Dzafic, D.; Baumeister, D.; Kowalewski, S.: Energieeffizientes Routing für Elektrorollstühle. In 13. Aachener Kolloquium Mobilität und Stadt (AMUS/ACMOTe), Aachen 2012; S. 65-67
- [FDWK11] Franke, D.; Dzafic, D.; Weise, C.; Kowalewski, S: Entwicklung eines mobilen Navigationssystems für Elektrofahrzeuge auf Basis von OpenStreetMap-Daten. In Konferenz für Freie und Open Source Software für Geoinformationssysteme (FOSSGIS), Heidelberg 2011; S. 92-99
- [FDWK11a] Franke, D.; Dzafic, D.; Weise, C.; Kowalewski, S.: Konzept eines Mobilen OSM-Navigationssystems für Elektrofahrzeuge. In Angewandte Geoinformatik 2011 - Beiträge zum 23. AGIT-Symposium (AGIT), Salzburg 2011; S. 148-157
- [RT10] Ramm, F.; Topf, J.: OpenStreetMap. Lehmanns, Berlin, 2010.

Browser-basierte SIP-Telefonie auf Grundlage von WebRTC

Thomas Falkenberg

Hochschule Darmstadt
Fachbereich Informatik
thomas.falkenberg@stud.h-da.de

Art der Arbeit: Bachelorarbeit

Betreuer/in der Arbeit: Prof. Dr. Michael Massoth, Jürgen Copenhaguen Bormuth

Abstract: Video- und Audio-Telefonie im Internet benötigen die Installation von Software oder Plug-Ins. Um Telefonie auf Webseiten nutzbar zu machen wurde der offene Standard WebRTC (Web Real Time Communication) entwickelt. Diese Arbeit verbindet WebRTC mit der bestehenden Voice-over-IP-Telefonie, durch die Entwicklung einer WebRTC-SIP-Bibliothek. Diese auf JavaScript basierende Bibliothek generiert SIP-Nachrichten (Session Initiation Protocol) und übermittelt diese über das WebSocket-Protokoll an einen Server, welcher die Nachrichten über das User Datagram Protocol (UDP) an SIP-Telefone weiterleitet. Durch ein geeignetes Back-End ist die Telefonie zwischen dem World Wide Web und bekannten Telefonnetzen möglich. Diese Funktionalität bietet verschiedenste Einsatzgebiete. Neben dem Einsatz in Büros, anstelle von Hardware-Telefonen, ist auch „Hotline-direkt-Anrufen“-Funktion auf verschiedensten Webseiten denkbar.

1 Motivation

Um Video- und Audio-Kommunikation aus dem Browser heraus zur Verfügung zu stellen, wie dies zum Beispiel bei Google-Hangout der Fall ist, hat Google die Software-Bibliothek WebRTC entwickelt und unter OpenSource Lizenz gestellt. WebRTC¹ bietet webbasierte Echtzeit Kommunikation (Real Time Communication) aus dem Browser (z.B. Google Chrome), ohne dass eine zusätzliche Installation von Software notwendig ist. Somit ist es möglich, dass Webseiten Mikrofon und Webcam des Nutzers, nach dessen Zustimmung nutzen. Dies ermöglicht zum Beispiel eine direkte Verbindung mit der Support-Hotline der Webseite herzustellen. Darüber hinaus wird die Nutzung von Audio- und Video-Telefonie über das Internet wesentlich erleichtert. So kann ein Benutzer von jedem beliebigen Computer ohne großen Aufwand mit einem Wunschpartner kommunizieren. Dies ermöglicht auch technischen Laien, durch einfaches Öffnen einer Internetseite den Zugriff auf die meist kostenlosen Dienste. Um Browser-basierte Telefonie als Service anbieten zu können, sollten Telekommunikationsanbieter WebRTC mit vorhandenen Voice-over-IP-Lösungen (VoIP) verbinden.

Aus diesem Grund wurde in dieser Arbeit eine WebRTC-SIP-Bibliothek entwickelt, die Telefonie vom Browser zu SIP-basierten VoIP-Telefonen ermöglicht

2 Grundlagen

2.1 SIP

Das Session Initiation Protocol (SIP) ist ein Signalisierungsprotokoll für Internet Telefonie (Voice over IP). SIP dient der Steuerung von Kommunikationsverbindungen von zwei oder mehr Teilnehmern und verwendet dafür eine Request-Response-Nachrichten-Kommunikation. Darüber hinaus bietet SIP Authentifizierung und Nutzerverwaltung, welche vom SIP-Registrar durchgeführt wird. Um die Kommunikation zu ermöglichen wird bei VoIP mittels SIP das Real-Time-Transport-Protocol (RTP) zur Übermittlung der Media-Daten eingesetzt.

2.2 WebRTC

Der von Google entwickelte offene Standard WebRTC bietet Echtzeit Kommunikation (Real-Time-Communication) im World Wide Web. WebRTC befindet sich momentan im Standardisierungsprozess beim World Wide Web Consortium (W3C), wie auch bei der Internet Engineering Task Force (IETF). Es bietet die Möglichkeit Media-Hardware, wie Webcam und Mikrofon eines Rechners, aus dem Browser zu nutzen. Die eingehenden Daten der Geräte werden kodiert, verschlüsselt und können gespeichert oder über Direktverbindungen (Peer-to-Peer) mit Kommunikationspartnern geteilt werden. Um auf die Funktionalitäten von WebRTC zugreifen zu können, sieht der Standard eine JavaScript-Bibliothek vor, die von Webentwicklern genutzt werden kann.

¹ <http://www.webrtc.org/>

3 Umsetzung

3.1 Bibliothek

Im Laufe der Arbeit wurde eine Bibliothek entwickelt, welche es ermöglicht, mittels WebRTC zu SIP-basierten Voice-over-IP-Clients zu telefonieren oder Instant-Messaging Nachrichten zu verschicken. Die Bibliothek ist vollständig JavaScript basiert und wurde mit Hilfe des Frameworks AngularJS² entwickelt. Zum Versenden von Chat-Nachrichten, wie auch zum Aufbau von Anrufen, verwendet die Bibliothek SIP. Die SIP-Nachrichten werden mithilfe der Bibliothek SIP-JS³ als JavaScript-Objekte erzeugt und über eine WebSocket-Verbindung versendet. WebSockets ermöglichen bidirektionale Verbindungen zwischen Browser und Server, sodass auch eingehende Anrufe realisiert werden können. Nachdem ein Anruf mittels SIP initialisiert wurde, wird durch WebRTC eine Media-Daten-Verbindung aufgebaut. Hierfür werden die Daten verschlüsselt und mittels RTP⁴ übermittelt.

Neben vorgeschriebenen Codecs, wie unter anderem G.711 und VP8, schreibt der WebRTC-Standard die Verschlüsselung und die Verwendung des Interactive Connectivity Establishment (ICE)⁵ als obligatorisch vor. Da nicht alle SIP-Telefone diese Funktionalitäten bieten, kann ein WebRTC-Client nicht direkt mit diesen Geräten kommunizieren. Aus diesem Grund wird ein geeignetes Back-End benötigt.

3.2 Back-End

Um die Funktionalität der WebRTC-SIP-Bibliothek nutzen zu können, ist das in *Abbildung 1* gezeigte Back-End notwendig.

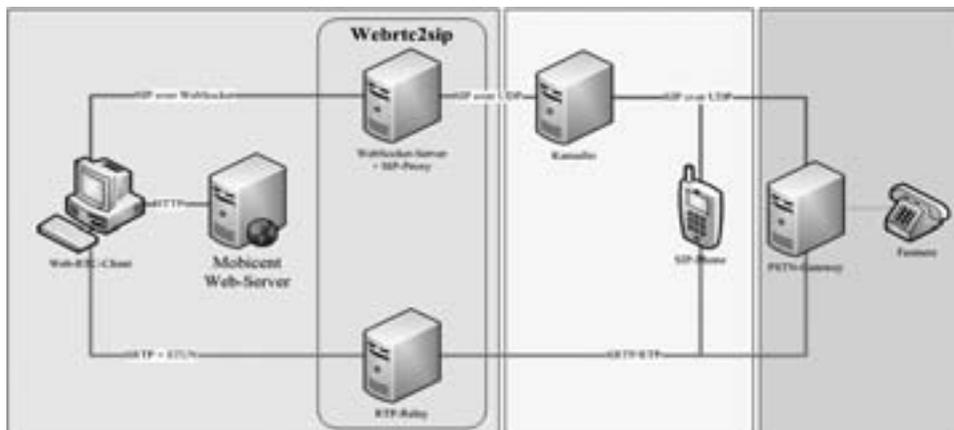


Abbildung 1: Architektur

² <http://angularjs.org/>

³ <http://code.google.com/p/sip-js/>

⁴ Real-Time Transport Protocol: RFC 3550

⁵ <http://tools.ietf.org/html/draft-ietf-mmusic-ice-19>

Die Architektur teilt sich in drei wesentliche Bereiche. Der Bereich rechts ist die Verbindung zwischen Voice-over-IP und klassischen Telefonnetzen und dient dem Anrufen von Festnetz- oder Mobilfunk-Telefonen. Der mittlere Bereich zeigt das Voice-over-IP-Netz. Als Teilnehmer dienen Hard- oder Software-SIP-Telefone, welche sich beim SIP-Registrar, hier der Kamailio⁶, anmelden. Für die Signalisierung wird das Session Initiation Protocol (SIP) und zum Übermitteln der Sprachdaten das Real-Time Transport Protocol (RTP) oder auch das verschlüsselte Secure RTP verwendet. Der Bereich links zeigt die für WebRTC notwendigen Komponenten. Der Web-Server liefert den WebRTC-Client an den Browser des Nutzers aus. Dieser baut eine WebSocket-Verbindung zum webrtc2sip-Server auf, über die die SIP-Nachrichten versendet werden. Webrtc2sip⁷ ist eine von Doubango Telecom entwickelte Software, die ein WebSocket-Server, einen SIP-Proxy und ein RTP-Relay bietet. Das RTP-Relay ist notwendig, da einige obligatorische Eigenschaften von WebRTC von den meisten SIP-Telefonen nicht unterstützt werden. Aus diesem Grund manipuliert webrtc2sip die am WebSocket-Server eingehenden SIP-Nachrichten und leitet die Kommunikation zwischen WebRTC- und SIP-Client über das Relay um. Die klare Trennung von WebRTC und SIP in der Architektur ermöglicht die Integration der WebRTC-SIP-Bibliothek in bereits bestehende Systeme.

Dadurch kann, z.B. in Call-Centern, Telefon-Hardware eingespart oder auf Service-Webseiten eine „direkt Anrufen“-Funktion ohne großen Aufwand hinzugefügt werden.

4 Ergebnisse und Fazit

Durch die beschriebene Bibliothek und das vorgestellte Back-End konnte ein Web-Client (siehe *Abbildung 2*) entwickelt werden, mit dem die Telefonie vom Browser in verschiedene Telefonnetze möglich ist.



Abbildung 2: WebRTC-SIP-Client

Sowohl Anrufe zwischen zwei Browsern, als auch zu SIP-basierten Voice-over-IP-Telefonen können mit dem Client durchgeführt werden. Da der Client gegenüber anderen Teilnehmern wie ein klassischer SIP-Client auftritt, können existierende Gateway-Lösungen verwendet werden. Dies ermöglicht dem Client auch die Telefonie zu klassischen Festnetz- oder Mobilfunk-Telefonen. Darüber hinaus bietet die Bibliothek, wie auch der Client, eine SIP-Standard konforme Instant-Messaging-, eine Kontakt- und eine Anruf-Protokoll-Funktion. Somit ist der Einsatz der Bibliothek sowohl für End-Kunden, als auch als Telefon-Ersatz in Firmen möglich.

Da die hier entwickelte Software auf Internet-Standards beruht, bietet sie gegenüber herkömmlichen Lösungen den Vorteil, weitestgehend Plattform unabhängig zu sein.

⁶ <http://www.kamailio.org>

⁷ <http://webrtc2sip.org/>

Self-Optimizing Binary Search Trees – A Game Theoretic Approach

Thim Frederik Strothmann

thim@mail.upb.de

Abstract: Binary search trees can be used as simple overlay networks, which incorporate an easy routing strategy. They suffer from the disadvantage that they are static, in the sense that they cannot adapt to changing communication interests of the nodes. Therefore a new network creation game which utilizes a binary search tree as the network structure and in which selfish nodes want to communicate with only a certain subset of all nodes is proposed. Each player of the game can change the structure of the network by using mechanics of self-optimizing binary search trees. Interestingly, in many situations the game does not converge and convergence can be related to certain structures of the communication interests. Furthermore, the performance of selfish nodes in convergent situations is compared to a social optimum by computing the Price of Anarchy. To analyze the game in non-convergent situations, the solution concept of sink equilibria is used. It can be observed that selfishness can lead to quadratic social cost, although a social optimum only has cost of $\Theta(n \cdot \log(n))$. Consequently, the Price of Sinking is $\mathcal{O}(\frac{n}{\log(n)})$.

Type of work: Master Thesis

Adviser: Prof. Dr. Christian Scheideler

1 Introduction

Since the early days of networks in computer science, scientists have been interested in finding good structures for networks and overlay networks. The goal to connect computers in efficient ways has led to many results in graph theory, network topology and routing strategies. Despite all those beautiful results, which aimed for intelligent design of a network, the internet has evolved into a massive network without a well-defined structure. To get new insights on the question of network creation with many participants (users, computers, etc.) that have different demands towards the network, game theory is a tool with good prospects. Instead of analyzing general networks in which players pay for the network edges, a shift to binary search tree networks in which the nodes (or players) are allowed to perform so-called rotations to change the structure of the network tree seems promising. The rotations are similar to rotations from Splay trees [ST85], which have an excellent data structure performance. The proposed game is called the *self-optimizing binary search tree game* or short *SOBST*-game.

2 The Model

$G_C = (V, E_C)$ is the connection graph of the *SOBST*-game. It is an undirected binary search tree, to allow communication between arbitrary node pairs. G_C needs to be a binary search tree, because they imply a straightforward routing strategy. The nodes are the players of the game and the private costs of the players are induced by the distances in the connection graph. The communication interests are modeled as a directed graph $G_I = (V, E_I)$. An edge (u, v) is in E_I if the node u wants to communicate with the node v . G_I is allowed to be any arbitrary directed graph. However, Chapter 3 takes a closer look at how the structure of G_I affects properties of the game like convergence or the Price of Anarchy.

The neighborhood $N_{G_I}(v)$ of a node v in the interest graph G_I is called the *interest set* of the node. The *communication cost* of a player v is the term $\sum_{u \in N_{G_I}(v)} d(v, u)$. Players want to minimize their communication cost by performing rotations and thereby changing G_C . An example for a single rotation is presented in Figure 1. Each node is allowed to perform multiple rotations per timestep on itself or on nodes from its interest set to move them upwards in G_C .

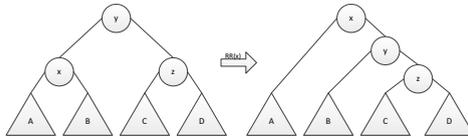


Figure 1: A single rotation of the node x .

The *SOBST*-game is played in rounds. One round is finished when all players have played a response at least once. However, no order is enforced inside a single round. The players play their responses in a randomized fashion. In the *SOBST*-game a best response of player i is a sequence of rotations that minimizes the communication cost of i . Furthermore, we make the assumption, that a node v wants the nodes in its interest set to be in one of its subtrees.

An example instance of a *SOBST*-game is presented in Figure 2. The *SOBST*-game can be interpreted as a variant of a Basic Network Creation Game (BNCG) as introduced by Alon et al. [ADHL10].

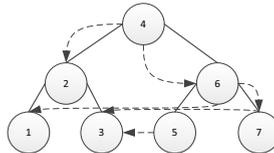


Figure 2: An instance of a *SOBST*-game. Interest graph edges are dashed; connection graph edges continuous.

The *SOBST*-game converges if independent from the initial connection graph every se-

quence of best responses converges. Otherwise the *SOBST*-game *oscillates*.

Similar to the common known solution concept *Nash equilibrium*, the connection graph $G_C = (V, E_C)$ of the *SOBST*-game is in a *rotation equilibrium* if there is no node $v \in V$ that can decrease its private cost unilaterally via an allowed rotation (or a sequence of rotations).

3 Results

3.1 Initial Observations

It is possible to construct simple instances of the *SOBST*-game which cannot converge. In fact, convergence can be related to certain properties of the interest graph. An instance of the game oscillates if the interest graph contains either a *direct* or an *indirect conflict*. Two edges $(u_i, v_i), (u_j, v_j)$ from E_I induce a direct conflict, if those two edges intersect, e.g. $u_i < u_j < v_i < v_j$. G_I has an indirect conflict, if G_I contains a directed circle of edges or if a node has indegree greater than 1.

Furthermore it is possible to compute the social optimum of the game, which is the connection graph that minimizes the sum over all connection costs, via dynamic programming. The recursive definition for the dynamic program takes advantage of the fact that the computation of the social cost (which is the sum over all private costs) can be subdivided into the computation of the social cost that is induced by the left and right subtree of a root node x plus the cost that are incurred by the root.

3.2 Convergence & Rotation Equilibria

According to the definition of conflicts there are only two possible classes of interest graphs that can converge: binary search trees (BST) and stars (a center node that has many outgoing edges). For both classes convergence can be proven. BST interest graphs converge to a social optimum. Therefore the so-called Price of Anarchy (PoA) (as defined by Koutsoupias and Papadimitriou [KP99]), which compares the equilibrium with highest cost to a social optimum, is 1. Interest stars cannot converge to a social optimum, but only to a connection graph that has twice the social cost, thereby leading to PoA of 2.

Independent of the structure of the interest graph, two subgraphs of the interest graphs that are separated from each other will also separate in the connection graph over time. Two subgraphs of the interest graphs are separated if they are neither connected by an edge, nor have a direct conflict with each other. Consequently, the assumption that the interest graph is either connected or induces a direct conflict for two non-connected subgraphs is reasonable. Moreover, results that hold for connected graphs transfer to interest graphs which have separated subgraphs.

3.3 Oscillations & Sink Equilibria

In non-convergent situations the PoA cannot be computed because rotation equilibria do not exist. Consequently, we need another solution concept to analyze the performance of players in oscillating instances of the *SOBST*-game. *Sink equilibria*, as introduced by Goemans et al. [GMV05], can be applied to non-convergent games. Sink equilibria are strongly connected components without outgoing edges in the transposition graph of a game instance. The corresponding measure for the social cost of the lack of coordination is called *Price of Sinking (PoS)*. In contrast to the original PoS definition, where the expected social cost of a sink equilibrium is computed, the PoS for the *SOBST*-game is computed by taking the worst-case state of a sink equilibrium.

The PoS for interest graphs with only direct conflicts is $\mathcal{O}(\frac{n}{\log(n)})$, where n is the number of players in the game. This upper bound deduces from the fact that there exists a sequence of best responses which results in a sink equilibrium state with social cost $\mathcal{O}(n^2)$ and the actuality that a social optimum only has costs of $\Theta(n \cdot \log(n))$. The PoS for interest graphs with only indirect conflicts is $\mathcal{O}(\frac{n}{\log(n)})$. The general proof idea for interest graphs with only direct conflicts can be transferred to interest graphs, where nodes have an indegree greater than 1. Furthermore, interest circles have a constant PoS, because every state in a sink equilibrium has the same social cost as a social optimum.

4 Conclusion

Even though binary search trees are simple networks with limited possibilities to change their structure, selfish nodes can still perform extremely bad compared to a centralized solution. $\mathcal{O}(n^2)$ is a tight upper bound for social cost, meaning that selfish nodes can achieve the worst possible connection graph in some instances of the game.

References

- [ADHL10] Noga Alon, Erik D. Demaine, MohammadTaghi Hajiaghayi, and Tom Leighton. Basic network creation games. In *Proceedings of the 22nd ACM symposium on parallelism in algorithms and architectures*, SPAA '10, pages 106–113, New York, NY, USA, 2010. ACM.
- [GMV05] Michel Goemans, Vahab Mirrokni, and Adrian Vetta. Sink Equilibria and Convergence. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 142–154, Washington, DC, USA, 2005. IEEE Computer Society.
- [KP99] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *In proceeding of the 16th annual symposium on theoretical aspects of computer science*, pages 404–413, 1999.
- [ST85] Daniel Dominic Sleator and Robert Endre Tarjan. Self-adjusting binary search trees. *J. ACM*, 32(3):652–686, July 1985.

Animation-based requirements analysis of a map application user interface to support critical MCI situations

Teodora Velikova, Tayfur Coskun, Sebastian Klingenbeck,
Johannes Roith, Eva Artinger, Gudrun Klinker

Fakultät für Informatik
Technische Universität München
Boltzmannstraße 3
85748 Garching bei München

{velikova, coskun, klingenb, roith, artingee, klinker}@in.tum.de

Abstract: In *mass casualty incidents (MCI)* the emergency medical services resources are overwhelmed by the number and severity of casualties. Unpredictable catastrophic events require a lot of information about the situation and an extremely well-organized emergency management response system. To increase the chance of survival, planning and preparedness are essential. Emergency response is undergoing a technology driven transformation that calls for an intuitive and self-explanatory *user interface (UI)* visualization. We explore the current real-time map application UI for MCI to make it more user-oriented and extend the available features. The main purpose of this research is to obtain a thorough and detailed understanding of the emergency response needs and interests. Furthermore, this work concentrates on creating a detailed functional specification defining the full set of visualization modalities to be developed. Our goal is to derive concrete requirements, based on realistically conceptualized MCI scenarios, which are presented as animations. As part of the SpeedUp¹ project this paper is a foundation for further insights and research in this area.

1 Motivation and Related Work

A *mass casualty incident (MCI)* is an event, which generates more patients than locally available resources can manage simultaneously. This requires an exceptional emergency response system that provides additional information about the number, location and medical condition of patients and the emergency scene. In this paper one major goal in map application design is to display all relevant information in a more intuitive and self-explanatory way and increase the efficiency of the situational awareness. Additionally, we aim to further develop the currently existing visualization techniques and explore alternative methods and concepts. In collaboration with the Arbeiter-Samariter-Bund München²

¹The project SpeedUp is funded by the German Federal Ministry of Education and Research (BMBF) within the program „Research for Civil Security“. Website: <http://www.speedup-projekt.de>

²The Worker’s Samaritan Federation (Arbeiter-Samariter-Bund Deutschland, ASB) is an aid and welfare organization devoted to rescue services.

we identified key areas of improvement in the map visualization considering the emergency response needs and interests. The main focus was on conducting a requirements analysis that has continuously evolved through two iterations of field-studies, adapted concepts and focus-group interviews. The chain of events in the medical and resource coordination poses some serious challenges. In our research we looked for typical occurrences and problems in MCI and conceptualized them in emergency situations to create realistic animation-based scenarios. This includes issues such as communication loss, insufficient information, poor data collection, resources planning, patient monitoring and documentation.

Most of related work focus on the communication infrastructure, patient tracking and monitoring, electronic documentation, recording of patient data and GPS integration. The visualization itself, usually based on symbols, is rarely an object of interest. One of the first universal symbol languages was the *NATO Military Symbols for Land Based Systems* that has been established for years. Killeen et al. [JKL06] present the WIISARD system, which uses PDAs to store and send the patients information. However, one disadvantage of this system is the lack of real-time patients information. Gao et al. [GW06] introduce an *e-Triage* system, which includes electronic triage tags, wearable vital sign sensors, software for tracking patients and communication support. The *e-Triage* project allows the technicians to acquire patients information, electronic documentation and patient tracking. This system was further enhanced [MIS⁺12] through a new visualization approach presenting three different views. The *inter-site view* shows a map overview of the latest status in multiple first-aid stations including the number of technicians and patients of each triage category. The *intra-site view* reveals more detailed information including location, triage category and vital signs of each patient on a 3D map. Another view, the *individual view*, shows the vital signs of a single patient on a tablet PC using augmented reality techniques.

2 Concept

We developed three different MCI scenarios, which describe typical emergency situations with their most frequent challenges. Our approach was to design realistic concepts in order to visualize circumstances which reveal weaknesses in MCI and focus on the whole chain of events - from the triage to the transportation of the patients. We considered improvements of the following visualization areas in real-time: 1) communication loss, 2) patient and paramedics data, 3) various activities and events, and 4) additional information and processes. We used the open source 3D software Blender³ and satellite pictures from Google Earth for the scenarios. Previously, we designed symbols and attributes for patients (represented by colored pins in Figure 1) and vehicles in the context of the SpeedUp project. We now revisited different alternatives for the visualization techniques, which underwent more than one evaluation iteration with constructive feedback. For the representation of various situations we used traffic signs because they are self-explanatory. In addition, we rely on military symbols as recommended by the ASB.

³<http://www.blender.org/>

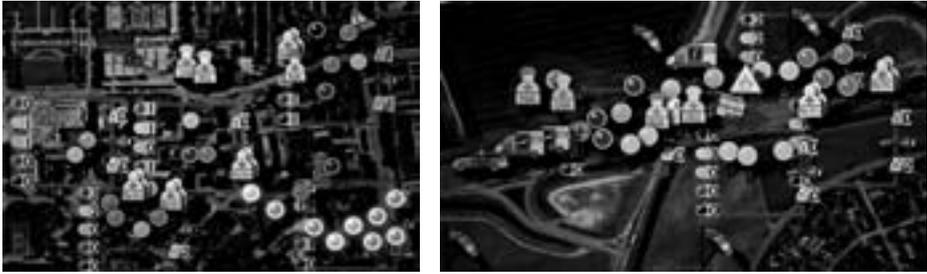


Fig. 1: Current UI visualization of typical critical MCI scenarios with the respective symbol for each patient, unit or object (high-resolution images are on our website⁴).

The system organization includes various visualization techniques. Fig. 1 (left) shows the collected information in detail such as location and number of patients per triage category, distinguished by color. The patient data and the allocated technicians are included in the appropriate *triage station* (depicted as a black square), which is important for the *on-scene commander*. This location overview is meant for the *ambulance incident commander (AIC)* who decides the appropriate dispatch of the medical personnel to each station. In addition, we consider possible loss of communication to some of the paramedics, which are painted in grey. Furthermore, we take into account the occurrence of street obstacles which can hinder the access to patients. We use tilted figures to avoid occlusions in the scenery. Fig. 1 (left) also adds a glow to newly detected patients with higher priority. The orange arrow on the ambulance in Fig. 1 (right) indicates the vehicle direction. Additional information about events or patients condition can be added to every object. We also considered another group visualization, which omits the patients location and ensures a clear overview of the number of patients and technicians for the AIC. For the representation of the *staging area* and of the *interim-care center* we use military symbols. In the third scenario in Fig. 1 (right) we consider a fire emergency, which is specified by its danger zone (an orange spot). Thus, we introduced new symbols for fire trucks, fire and wind direction.

In the preliminary evaluation, we presented the initial concepts and ideas to the medical personnel in order to get a first impression about their requirements. Subsequently, we were able to perform a detailed requirements analysis and to examine the user acceptance of the map application. Our final evaluation was done with five participants, who underwent individual interviews while observing the three animations without a prior introduction. During that they were asked to give feedback about the visualized data.

3 Assembled User Interface Requirements and Discussion

Through the variance of the results we were able to greatly extend the set of requirements. All of the participants agreed that the use of two different symbols for paramedics, namely

⁴<http://campar.in.tum.de/Main/TeodoraVelikova>

vehicles and persons, is too confusing and the second one is preferred. Another popular requirement is to avoid different symbol languages and to simply settle on the universal military symbols. The symbol for additional information (the orange *i*) was rated as highly useful. One very important missing feature is the possibility to manually define the shape of various stations. The visualization of objects movements is unnecessary. Moreover, the amount of displayed information should be determined by the map scale. In case of a small scale map, it should show just the number of paramedics and patients per triage category for each station. This information is required by the AIC, who needs to have an overview of the scene. On the other hand, the exact location of the patient is only important for the on-scene commander and should be displayed electively. Additionally, symbols for uninjured patients and other medical personnel should be developed. Radio call names should be displayed on every object. Furthermore, the current status of the available ambulances should be visualized. A simple map legend for military symbols could be very useful. We could visualize the transport regulations and prioritize patients according to transport emergency. Another feature that allows filtering the symbols to be displayed could be implemented. Vehicles which are not able to transport patients should be assigned a different role. Moreover, we could include a feature that allows us to displace overlapping patients to improve clarity. Finally, whenever an alert has been shown, the system should send back an acknowledgement message to all other users.

We have enhanced the whole process visualization system with some new concepts and believe that our study provides valuable insight into the future design of the map application UI. The main conclusion is that lower levels of the emergency hierarchy should be described in greater detail and include more information about the scenery. In contrast the bigger the overview, the smaller the amount of data to be visualized. This requirement makes sure that only the important information is perceived by the user. Our study is build on collaborative learning and defines some new research topics in the field of map applications. For future work we recommend to maintain a close relationship with the medical personnel in order to make more progress not just on a functional, but also on a visualization level.

References

- [GW06] Tia Gao and D. White. A Next Generation Electronic Triage to Aid Mass Casualty Emergency Medical Response. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, volume Supplement, pages 6501–6504, 30 2006-sept. 3 2006.
- [JKL06] C. Buono W. Griswold J. Killeen, T. Chan and L. Lenert. A Wireless First Responder Handheld Device for Rapid Triage, Patient Assessment and Documentation during Mass Casualty Incidents. In *AMIA Annual Symposium Proceedings*, volume vol. 2006, page 429. merican Medical Informatics Association, 2006.
- [MIS⁺12] T. Mizumoto, S. Imazu, Weihua Sun, N. Shibata, and K. Yasumoto. Emergency medical support system for visualizing locations and vital signs of patients in Mass Casualty Incident. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 740–745, march 2012.

Konzeption und Implementierung einer medizinischen Multi-Touch-Anwendung

Natalie Wenz
Anja Rothbart

Hochschule Reutlingen
Medien- und Kommunikationsinformatik

Natalie.Wenz@Student.Reutlingen-University.de
Anja.Rothbart@Student.Reutlingen-University.de

Art der Arbeit: Projektbeschreibung

Betreuer/in der Arbeit: Prof. Dr. Gabriela Tullius, Prof. Dr. Uwe Kloos

Abstract: Im Rahmen des Projekts VRLab¹ an der Hochschule Reutlingen wurde eine Multi-Touch-Anwendung MeViTouch entwickelt, die speziell für die Visualisierung medizinischer Bilddaten konzipiert wurde. In diesem Artikel werden grundlegende Ansätze aufgezeigt, die eine Integration medizinischer Bilddaten in eine Anwendung ermöglicht und die sich gleichzeitig mit Hilfe von Multi-Touch Gesten bedienen lässt.

1 3D-Visualisierung medizinischer Daten

Speziell in der Medizin wuchs die Datenmenge in den letzten Jahrzehnten enorm an. Dank der ständigen Weiterentwicklung von Visualisierungstechniken kann heutzutage eine sehr realistische Darstellung medizinischer Daten erreicht werden, welche für Diagnosen enorme Verbesserungen liefern [PPA10]. Die Visualisierung dreidimensionaler Daten bietet in der Operationsvorbereitung, Therapieunterstützung sowie in der Diagnoseunterstützung einige Vorteile. Systeme, wie die Computertomographie oder ähnliche erfassen computergesteuert eine Vielzahl medizinischer Daten. Räumliche Strukturen und anatomische Informationen des Patienten können somit besser analysiert werden [L11].

¹ Virtual Reality Laboratory: <http://www.vrlab.reutlingen-university.de>

Für die Darstellung dreidimensionaler Daten wurde in der Vergangenheit eine Vielzahl an Frameworks entwickelt, die sowohl oberflächenbasierte Techniken als auch Volumenvisualisierungen bieten [PB07]. Meist werden für 3D Anwendungen Rohdaten von CT- oder MRT-Aufnahmen im medizinischen Standardformat DICOM verwendet. Bei CT- -Daten lässt sich die Darstellung der Gewebestruktur anhand von Schwellwerten definieren, die jeweils für einen bestimmten Bereich des Volumens stehen. Die Werte werden in der so genannten „Hounsfield Skala“ mittels Hounsfield Einheiten definiert [SPL96].

2 Multi-Touch in der Medizin

Medizinische Einsätze werden immer fortschrittlicher und benötigen moderne Techniken zur Visualisierung der Patientendaten. CT- und MRT-Aufnahmen bieten hierzu die grundlegenden Informationen. Jedoch sind sie als Rohdaten oft nicht einfach zu interpretieren. Eine dreidimensionale Volumenvisualisierung bietet erweiterte Kenntnisse über das Innere des Volumens und gibt den Ärzten eine realistischere Übersicht. Gemeinsames Arbeiten an den Volumendaten ist durch traditionelle per Maus gesteuerte Desktopanwendungen nur bedingt machbar. Durch den vorhandenen Multi-Touch-Tisch im VRLab lassen sich die Patientendaten von allen Seiten manipulieren, was zur kollaborativen Arbeitsweise beiträgt [Lu11].

Touch-Anwendungen werden oft als „intuitiv“ und als „einfach zu bedienen“ bezeichnet [PPA10], was sich durch den großen Hype um Smartphones und Tablets deutlich gezeigt hatte [Lu11]. Bestimmte Interaktionen lassen sich per Touch deutlich schneller durchführen, als dies mit der traditionellen Maus möglich ist [Be10].

Entwicklungen medizinischer Anwendungen auf einem MT-Gerät finden in der Forschung immer häufiger Anklang. Meist finden diese Anwendungen Einsatz in der Ausbildung medizinischer Studenten, Diagnosefindung, Therapieplanung, Operationsvorbereitung oder während der Operation selbst [Lu11].

3 Design der Anwendung

Für die Implementierung der medizinischen Multi-Touch-Anwendung wurde das Visualisierungstool VTK² (Visualization Toolkit) eingesetzt, welches eine umfangreiche Bibliothek mit unterschiedlichen Formaten bereitstellt. VTK besitzt dabei eine bestimmte Abfolge von Vorgängen, die als Pipeline bezeichnet wird. MeViTouch baut auf den beiden Frameworks VTK und MT4J³ auf, denen wiederum die OpenGL Bibliothek zu Grunde liegt. MT4J entspricht einem Multi-Touch Framework basierend auf Java. Eine übersichtliche Darstellung bietet Abbildung 1.

² <http://www.vtk.org/>

³ <http://www.mt4j.org>

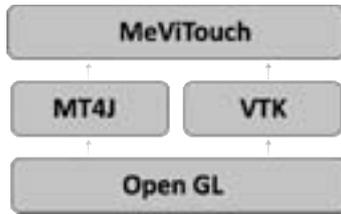


Abbildung 1: Aufbau von MeViTouch auf den Frameworks MT4J und VTK

Die Anwendung MeViTouch sollte eine einfach zu bedienende Multi-Touch-Anwendung werden. Aus diesem Grund wurde das Benutzer Interface schlicht gehalten und als Interaktionsmedium ein einzelnes Kreismenü verwendet. Über dieses zentrale Bedienelement, lassen sich die Schwellwerte, beispielsweise nach der Hounsfield Skala bei CT-Daten, filtern.

Die Rohdaten im DICOM Format werden durch die bereits in VTK vorgegebene Import Schnittstelle eingeladen und in ein 3D Objekt transformiert. Die Aufnahmen entsprechen dabei Pixel oder Voxel Daten, die mit Hilfe eines 3D Volumen Visualisierungs-Algorithmus bearbeitet werden können [RC94]. In Bezug auf die Anwendung MeViTouch wird das 3D Bild auf einer 2D Fläche definiert. Die VTK Klasse ist an sich eine Komponente, die über den SwingTextureRenderer in die MT4J Szene integriert werden kann. Das Benutzermenü wird anschließend über das VTK Objekt gelegt. Das Kreismenü ist eine Komponente die einem MTRectangle hinzugefügt wird, um somit die Funktionalität von MT4J zu erhalten. Damit lassen sich die Schwellwerte für die Darstellung von Haut, Knochen und Fettgewebe setzen. Beispielhaft wurden in dieser Anwendung CT-Daten eines Kopfes verwendet. Abbildung 2 stellt einen Ausschnitt des Kopfsvolumens sowie das Benutzermenü dar.



Abbildung 2: MeViTouch

In den nächsten Schritten der Entwicklung soll es möglich sein anhand eines Slice-Werkzeuges Teile des 3D Modells zu extrahieren. Weiterhin ist es angedacht eine Möglichkeit zu bieten, markante Bereiche zu selektieren und diese gegebenenfalls einzufärben, um somit eine Hervorhebung von operationsrelevanten Teilen zu bieten.

Literaturverzeichnis

- [Be10] Bello, F. et. al: "The Need to Touch Medical Virtual Environments?" Workshop of Medical Virtual Environments at IEEEVR2010
- [LRZ10] Laufs, Uwe; Ruff, Christopher; Zibuschka, Jan: MT4j – A Cross-platform Multi-touch Development Framework. In: Transactions of the IRE Professional Group on, Bd. 2010.
- [Li11] Liang, Ying-Zong et. al.: Design of an Interactive 3D Medical Visualization System. In: Proceedings of the 2011 International Conference on Wavelet Analysis and Pattern Recognition, Guilin, 10-13 July 2011.
- [Lu11] Lundström, Claes et. al.: Multi-Touch Table System for Medical Visualization: Application to Orthopedic Surgery Planning. In: IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 17, NO. 12, Dezember 2011.
- [PPA10] Paolis, Lucio T. de; Pulimeno, Marco; Aloisio, Giovanni: Visualization and Interaction Systems for Surgical Planning. In: Information Technology Interfaces (ITI), 32nd International Conference 2010, S. 269–274.
- [PB07] Preim, Bernhard; Bartz, D. (2007): Visualization in medicine. Theory, algorithms, and applications. Amsterdam ;, Boston: Morgan Kaufmann.
- [RC94] Rouxl, Christian; Coatrieux, Jean Louis: Visualization in Medicine: VIRTUAL Reality or ACTUAL Reality ? In: Visualization, 1994., Visualization '94, Proceedings., IEEE Conference on, Bd. 1994.
- [SPL96] Schneider, Uwe; Pedroni, Eros; Lomax, Antony: The calibration of CT Hounsfield units for radiotherapy treatment planning. In: Physics in Medicine and Biology 1996, Volume 41, Number 1.

Abschätzung des Disparitätsraums für das Stereo-Matching mithilfe von Bildmerkmalen

Christoph Drexler

FernUniversität in Hagen
Lehrgebiet Mensch-Computer-Interaktion
christoph@drexler.eu

Art der Arbeit: Masterarbeit im Studiengang Master of Science (Praktische Informatik)
Betreuer/in der Arbeit: Dr. Klaus Häming, Prof. Dr. Gabriele Peters

Abstract: Beim Stereo-Matching zwischen zwei Bildern einer Szene werden Disparitäten zwischen einzelnen Bildpunkten ermittelt. Dabei wird die Größe des Disparitätsraumes, in dem die Disparitäten gesucht werden, üblicherweise global vorgegeben. Diese Vorgangsweise stößt an ihre Grenzen, wenn die tatsächliche Größe des Suchraumes unbekannt ist, wie beispielsweise bei der Freihandfassung mit unkalibrierten Kameras. Um die Anzahl der zu berechnenden Disparitätsstufen zu reduzieren, wird hier ein Algorithmus vorgeschlagen, der den Disparitätsbereich unter Zuhilfenahme von ohnehin vorhandenen Merkmalskorrespondenzen iterativ und für lokale Bildbereiche adaptiv abschätzt. In einem Benchmarking mit bekannten Datensätzen hat sich der Algorithmus als effizient und genau erwiesen.

1 Problemstellung

Die meisten einschlägigen Forschungsarbeiten zum Stereo-Matching¹ gehen davon aus, dass die beiden verwendeten Bilder von kalibrierten Kamerapositionen in exakt paralleler Position, aus demselben Abstand und unter kontrollierten Beleuchtungsbedingungen erstellt wurden. Aufgrund der parallelen Ausrichtung kann die Differenz dann durch einen einzelnen Wert („Disparität“), der die horizontale Verschiebung der beiden Bildpunkte in Pixeln angibt, beschrieben und mittels sog. Disparitätskarten optisch dargestellt werden. Die Anzahl der in Frage kommenden Disparitätsstufen hängt dabei nur von den Abständen der beiden Kamerapositionen zueinander und zum fotografierten Objekt ab und wird beispielsweise bei den oft zum Testen verwendeten Middlebury-Datensätzen (vgl. <http://vision.middlebury.edu/stereo/submit/>) vorgegeben.

Bei der Freihandfassung von 3D-Objekten treffen einige dieser idealisierten Bedingungen nicht zu. Vor allem ist es äußerst unwahrscheinlich, dass die Kameraachsen der beiden Bilder parallel zueinander verlaufen, sie werden vielmehr fast immer windschief

¹ Beim Stereo-Matching geht es darum, in zwei Bildaufnahmen derselben Szene jene Pixelpaare zu identifizieren, die denselben Punkt der Szene abbilden, um aus der Positionsdivergenz der beiden Punkte auf Tiefeninformationen bzgl. der abgebildeten Objekte zu schließen, vgl. [Sz11].

zueinander stehen. Während sich das Problem, dass die zusammengehörigen Bildpunkte dann meistens nicht mehr auf derselben Bildzeile liegen, relativ einfach durch eine sog. Rektifizierung lösen lässt, bleibt ein weiterer Nebeneffekt bestehen: Bei parallelen Kameraachsen kommen nur relativ wenige Disparitätswerte in Betracht, sofern von einem bestimmten Mindestabstand der Kamera zur aufgenommenen Objektszene ausgegangen werden kann (was meistens der Fall ist). Bei windschiefer Position der Kameraachsen sind die in Frage kommenden Disparitätswerte vorerst nur durch die horizontale Pixelanzahl der Bilder begrenzt. Aber nicht nur theoretisch, sondern auch praktisch führt der unterschiedliche Betrachtungswinkel zu einer deutlich höheren Zahl an Disparitätsstufen. Da bei fast allen Verfahren zum Stereo-Matching die Berechnungen für eine bestimmte Anzahl möglicher Disparitätsstufen durchgeführt werden und die Rechenkomplexität bei der Freihandfassung von 3D-Objekten möglichst gering gehalten werden soll, stellt sich an dieser Stelle die Frage, ob und wie die Anzahl der zu berechnenden Disparitätsstufen effizient reduziert werden kann.

2 Algorithmus zur Abschätzung des Disparitätsraums

Während das genannte Problem u. U. auch gelöst werden könnte, indem die Disparitätswerte für das Bildpaar zuerst in geringerer Auflösung berechnet und dann durch Hochskalieren verfeinert werden (ähnlich [WJ11], das v. a. die Minimierung des Speicherbedarfs anzielt), wird hier ein Lösungsweg vorgeschlagen, der bei der Überlegung ansetzt, dass ja bereits vor dem eigentlichen Stereo-Matching, nämlich bei der Rektifizierung der Bilder, eine Reihe von Bildpunkten einander zugeordnet wurden, deren Disparitätswerte sich sehr einfach berechnen lassen. Um ein Bildpaar überhaupt rektifizieren zu können, werden nämlich üblicherweise zuvor mittels eines sog. Feature-Detektors besonders auffällige Merkmale der Objektszene ermittelt. Erst wenn zumindest sieben „sichere“ Merkmalspaare identifiziert worden sind, die einander mit hoher Wahrscheinlichkeit zugeordnet werden können, ist eine Rektifizierung der Bilder möglich.

Allerdings werden bei dieser Suche meist nicht nur sieben Merkmalspaare ermittelt, sondern sehr viel mehr. Die Idee des hier vorgeschlagenen Algorithmus ist es, von diesen Disparitätswerten auszugehen und auf deren Grundlage einen Disparitätsraum für die übrigen Bildbereiche festzulegen, der die korrekten Disparitätswerte mit hoher Wahrscheinlichkeit enthält, aber trotzdem deutlich kleiner ist als der theoretisch mögliche Suchraum, sodass der Rechenaufwand – bei annähernd gleich guten Ergebnissen – deutlich verringert werden kann.

Dazu werden in einem ersten Schritt zu jedem Merkmal, dessen Zuordnung als korrekt angenommen werden kann, fünf Nachbarmerkmale gesucht. Als Kriterium der „Nähe“ wird die maximale Farbdistanz² zwischen je zwei Pixeln, die auf einer Linie zwischen den beiden Merkmalen liegen, verwendet. Diese Vorgangsweise soll sicherstellen, dass Punkte, die zur selben Fläche gehören, gegenüber Punkten, die an verschiedenen Seiten von Diskontinuitäten liegen, bevorzugt werden. An der Position des Merkmals wird nun

² Die Farbdistanz wird als $D(p, q) = \max(|r_p - r_q|, |g_p - g_q|, |b_p - b_q|)$ bestimmt, wobei es sich bei $r_p, r_q, g_p, g_q, b_p, b_q$ um die Farbwerte der beiden Bildpunkte handelt.

die jeweilige Disparität als untere und obere Schranke für den Suchraum eingetragen. Die Disparitätswerte für die Bildpunkte auf den Geraden zwischen je zwei benachbarten Merkmalspaaren werden linear interpoliert³. Diese Ersteinschätzung der Disparität wird nun im nächsten Schritt zu einem Suchraum erweitert, indem die untere Schranke um einen fixen Wert erniedrigt, die obere Schranke um denselben Wert erhöht wird.

Als nächstes werden die ermittelten Disparitätssuchbereiche auf benachbarte Bildpunkte übertragen (Abb. 1a). Dabei sollen die Suchbereiche möglichst nur an jene Nachbarpixel weitergegeben werden, die ungefähr die gleiche Disparität aufweisen. Dabei möchte man den Suchraum nicht über Diskontinuitäten hinaus ausweiten. Da man bei der für lokale Stereo-Matching-Algorithmen wichtigen Kostenaggregation auch bemüht ist, die Kosten nicht über Diskontinuitäten hinaus aufzuaddieren, ist es sinnvoll, dort Anleihen zu nehmen. Daher werden gemäß der Logik der „Cross-based Aggregation“ (vgl. [Me11]) Unterstützungsregionen gebildet, die allerdings nicht wie im ursprünglichen Verfahren zur Kostenaggregation verwendet werden, sondern zur Ausweitung des Disparitätsraums. Dabei wird der Suchraum zuerst in horizontaler und dann in vertikaler Richtung so lange ausgedehnt, wie die Farbdistanz zweier benachbarter Pixel sowie die Farbdistanz zum ursprünglichen Pixel, dessen Suchbereich ausgedehnt wird, einen festgelegten Schwellenwert nicht überschreitet. Der Schwellenwert wird umso niedriger gewählt, je weiter das in Frage kommende Pixel vom Ausgangspixel entfernt ist, sodass die Ausbreitung des Suchraums in stärker strukturierten Bildregionen schneller stoppt als in texturlosen. Falls der Bildpunkt, auf den hin der Suchraum bei diesem Verfahren ausgedehnt wird, bereits eine Einschätzung für eine untere und obere Schranke besitzt, wird diese nicht überschrieben, sondern entsprechend erweitert. Um bereits in der ersten Iterationsstufe Disparitätswerte für möglichst viele Pixel berechnen zu können, wird dieser Schritt dreimal – immer zuerst horizontal und dann vertikal – durchgeführt.

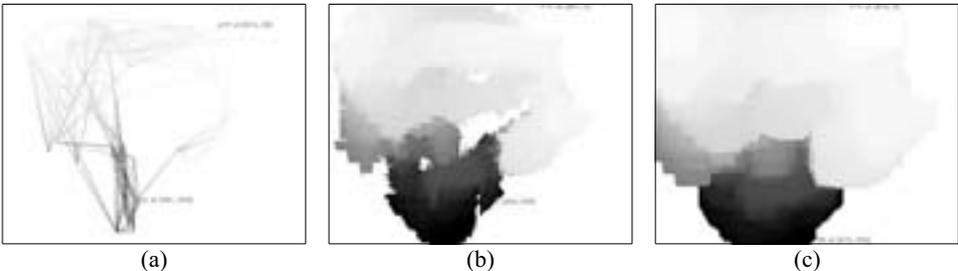


Abb. 1: Schrittweise Einschätzung des Disparitätsraums (obere Suchraumgrenze) anhand des Bildpaars „Wohnraum“ (Abb. 2): (a) Initialisierung mittels Interpolation zwischen benachbarten Merkmalen; (b) Ausdehnung des Suchraums mittels der Unterstützungsregionen; (c) Ausdehnung um eine fixe Distanz in allen vier Richtungen. Dunkel = niedrige, hell = hohe Disparitätswerte.

Auf diese Weise entsteht – abhängig von der Anzahl der zur Verfügung stehenden Merkmale sowie von den Eigenschaften der Szene und der Fotografien – eine zusammenhängende Teilmenge an Pixeln, für die eine erste Einschätzung bzgl. des Disparitätsraums getroffen wurde (Abb. 1b). Diese Teilmenge weist allerdings an den Rändern viele Fransen auf, was sich für die nachfolgende Kostenaggregation nachteilig auswirkt.

³ Die Verwendung der Farbdistanz als Kriterium trägt an dieser Stelle dazu bei, dass auch weiter entfernte Punkte miteinander verbunden werden und so eine größere Zahl an Disparitätsvermutungen zustande kommt.

Deshalb werden in einem abschließenden Schritt (Abb. 1c) die Disparitätsräume noch einmal auf eine fixe Anzahl von Nachbarpixeln in allen vier Hauptrichtungen ausgedehnt, wiederum zuerst horizontal und dann vertikal.

Für jene Pixel, für die bereits ein Suchraum festgelegt worden ist, wird nun das Stereo-Matching durchgeführt. Die dabei ermittelten Disparitäten werden verwendet, um damit den nächsten Iterationsschritt der Suchraumbestimmung zu initialisieren. Der Prozess wird bis zur Erfüllung eines definierten Abbruchkriteriums (z. B. bis eine bestimmte Prozentzahl an Pixeln berechnet wurde) wiederholt.

3 Evaluation

Der o. g. Algorithmus wurde in einer Referenzimplementierung⁴ erprobt und evaluiert, und zwar mittels der Bildpaare „Venus“, „Teddy“ und „Cones“ der Middlebury-Datensätze sowie mit Bilddaten des Lehrgebiets Mensch-Computer-Interaktion der Fernuniversität in Hagen⁵. Die Ergebnisse zeigen, dass der Suchraum durch den beschriebenen Algorithmus effizient eingeschätzt und eingeschränkt werden kann, weil die untere und obere Disparitätsschranke nicht nur für das Gesamtbild, sondern für jedes einzelne Pixel bestimmt werden. Probleme ergeben sich, wenn nur wenige Merkmalspaare gefunden werden, allerdings würde das Stereo-Matching dann wohl schon zuvor an der Rektifizierung scheitern.

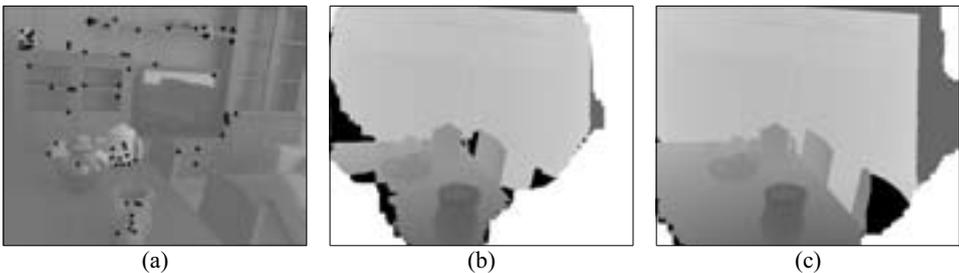


Abb. 2: Ergebnisse für das Beispiel „Wohnraum“: (a) verwendete Merkmale; (b)–(c) geschätzte Disparitätsräume im ersten und zweiten Iterationsschritt (grau: korrekte Disparität liegt innerhalb des Suchraums; schwarz: falsche Schätzung. Fehler des ersten Schritts werden im zweiten korrigiert. Schwierigkeiten hat der Algorithmus v. a. mit kontrastarmen, weil schattigen Bildbereichen.

Literaturverzeichnis

- [Sz11] Szeliski, R.: *Computer Vision – Algorithms and Applications*. Springer, London, 2011.
- [WJ11] Won, K.H.; Jung, S.K.: hSGM: Hierarchical Pyramid Based Stereo Matching Algorithm. In (Blanc-Talon, J. Hrsg.): *Lecture Notes in Computer Science* 6915, 2011.
- [Me11] Mei, X. et al.: On Building an Accurate Stereo Matching System on Graphics Hardware. In *GPUCV'11: ICCV Workshop on GPU in Computer Vision Applications*, 2011.

⁴ Die Implementierung ist unter <https://bitbucket.org/ChristophDrexler/masterarbeit> frei verfügbar.

⁵ Die Bilddaten wurden von Dr. Klaus Häming erstellt. Sie berücksichtigen spezifische Besonderheiten der Freihandfassung von 3D-Objekten: Beleuchtungsdifferenzen, windschiefe Kameraachsen, Bildrauschen.

Entwicklung einer Kleidungssimulation für 3D-Charaktere auf Basis eines Masse-Feder-Systems

Jorin Guderian

4guderia@informatik.uni-hamburg.de

Art der Arbeit: Diplomarbeit

Betreuer/in: Dr. Werner Hansmann, Prof. Dr. Leonie Dreschler-Fischer

Abstract: Im Rahmen dieser Diplomarbeit ist ein in Echtzeit arbeitendes System zur Simulation des physikalischen Verhaltens von Textilien entwickelt worden. Textilien werden im Simulator durch ein Netz aus Masse-Feder-Systemen realisiert. Das Verhalten jeder Feder wird nach dem Hookeschen Gesetz berechnet und zur Zeitintegration wurde das Verlet-Verfahren implementiert. Die Applikation lädt Geometrien für einen virtuellen Catwalk (Bühne und Raum), einen animierten 3D-Charakter sowie die Geometrie eines Kleidungsstücks (Textil), dessen physikalische Eigenschaften in Echtzeit simuliert und dargestellt werden. Die Echtzeitfähigkeit des Simulators wird durch einen eigens entwickelten Preprocessingschritt, einen eigens entwickelten Kollisionserkennungsalgorithmus sowie durch die Implementation von Multithreading-Unterstützung ermöglicht. Zum Einbinden des 3D-Charakters wurde die Cal3D-Bibliothek und zum Laden sonstiger Geometrien die ColladaDom-Bibliothek verwendet [D⁺06, BF08]. Die Applikation ist in C++ und OpenGL geschrieben.

1 Einleitung

Eine effiziente Modellierung des physikalischen Verhaltens von Textil wird in der Computergrafik durch ein Netz von Masse-Feder-Systemen realisiert (Abb. 1) [Pro95]. Ein einzelnes Masse-Feder-System beschreibt das physikalische Verhalten einer realen mechanischen Feder. Die Berechnung des räumlichen Zustandes der Netzstruktur erfordert jedoch einen hohen Rechenaufwand. Der Berechnungsaufwand der Kollisionserkennung und der Kollisionsbehandlung eines bspw. auf der Haut eines 3D-Charakters aufliegenden Textils ist sogar noch größer. Aus diesem Grund wird z.B. in Computerspielen das Verhalten der Kleidung eines 3D-Charakters meistens vorberechnet und zur Laufzeit nur in Form einer Animation dargestellt. Eine physikalische Simulation von Textil kommt selten und dann auch nur in vereinfachter Form vor. Erst jetzt erreichen Computer die nötige Leistung, um die physikalische Simulation von Textil in Echtzeit zu realisieren. Deshalb ist sie ein aktuelles Forschungsgebiet der Informatik. Viele 3D-Bearbeitungs- und Animationswerkzeuge für Nicht-Echtzeitanimation bieten die Möglichkeit, in einer anzufertigenden Animationssequenz auch Textil physikalisch zu simulieren. Da das Verhalten des zu simulierenden Textils von vielen Parametern abhängt, ist es schwer, für diese und deren

Wirkung auf das Verhalten des Textils ein Gefühl zu bekommen. Um die Wirkung einer Parameteränderung sehen zu können, muss nach jeder Änderung eines Parameters auf die gerenderte Sequenz gewartet werden. Das Besondere des hier entwickelten Textilsimulators ist, dass seine Simulation in Echtzeit abläuft. So ist es möglich, durch das Bewegen von Schieberreglern die Auswirkungen auf das Verhalten des Textils direkt sehen zu können und so ein intuitives Verständnis für die Einflüsse der Parameter zu bekommen. Aufgrund des begrenzten Rahmens kann in diesem Paper nicht auf alle bedeutenden Teilaspekte der Arbeit bspw. das Multithreading eingegangen werden. Zur weiterführenden Lektüre wird daher auf die gleichnamige Diplomarbeit des Autors verwiesen.

2 Generierung der Textilstruktur

Der *Loader* des Simulators lädt zunächst eine aus Quads (Vierecken) aufgebaute Geometrie, welche die Form eines beliebigen Textils beschreibt. Dabei liegen im ersten von 2 Arrays (Vertex-Array) die Koordinaten der Vertices und im zweiten sind für jedes Quad vier Indizes auf dessen Punkte im ersten Array gespeichert (Quad-Array). Die Vertices der Geometrie dienen als Positionen für die Masseknoten des anzulegenden Netzes. Zur Realisierung der Textilstruktur mit Hilfe von Federn müssen alle Knoten mit ihren *direct neighbors*, den *shear neighbors* und den *bend neighbors* über jeweils eine Feder verbunden werden (Abb. 1). Der erarbeitete Algorithmus zur

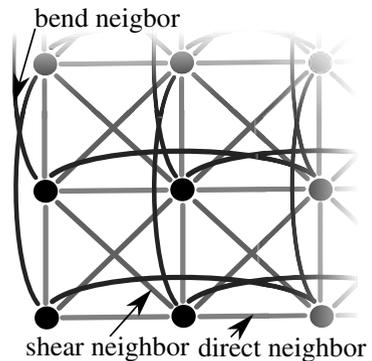


Abbildung 1: Netzstruktur des Textils

Findung der *direct neighbors* geht folgendermaßen vor: Es werden der erste Index des Vertex-Arrays ausgewählt und alle Quads vermerkt, in denen er vorkommt. Da in jedem Quad 4 Indizes vermerkt sind, die Nummerierung pro Quad also von 0 bis 3 geht, haben die *direct neighbors* eines Punktes in einem Quad die Indizes $(k + 1) \bmod 3$ und $(k - 1) \bmod 4$, wenn der Punkt im Quad den Index k hat. Für alle folgenden Punkte des Vertex-Array wird gleich verfahren und für die Punkte und deren Nachbarn werden entsprechend Datenstrukturen zur Speicherung der Zustände der Federn zwischen ihnen generiert. *Shear neighbors* sind in den ausgewählten Quads jeweils die Punkte mit dem Quad-index $(k + 2) \bmod 4$. Zur Findung der *bend neighbors* eines Punktes müssen wiederum Quads ausgewählt werden, die die Eigenschaft haben, dass sich in ihnen ein *direct neighbor* und ein *shear neighbor* befindet. In diesen Quads ist genau der Punkt der *bend neighbor*, der im Array den Index $(h + 1) \bmod 3$ oder $(h - 1) \bmod 4$ hat und kein *shear neighbor* des betrachteten Partikels ist, wobei h der dortige Index des *direct neighbors* zum betrachteten Partikel ist.

3 Erkennung und Behandlung von Kollisionen

Ist nun ein Kleidungsstück generiert und außerdem ein 3D-Charakter geladen, der das Kleidungsstück tragen soll, muss ein Algorithmus dafür sorgen, dass das Kleidungsstück

nicht durch den 3D-Charakter hindurchfällt. Hinzu kommt, dass das Textil nicht direkt aufliegen darf, da sonst die Oberfläche des 3D-Charakters durch die des Textils durchschimmert. Der entwickelte Algorithmus misst unter Zuhilfenahme des RayTriangleIntersection-Algorithmus von Möller und Trumbore den Abstand des Textils zur Oberfläche des 3D-Charakters und umgekehrt [MT97]. Kommt ein Vertex des Textils der Oberfläche des 3D-Charakters zu nahe, wird er auf einen festgelegten Mindestabstand gesetzt [MTTA00]. Der umgekehrte Fall ist schwieriger aufzulösen, da hier alle Vertizes eines Oberflächensegmentes des Textils unter Berücksichtigung der Hebelgesetze verschoben werden müssen. Diese Problemstellung lässt sich auf die Fragestellung reduzieren, wie ein Dreieck, das an einem einzigen Punkt aufliegt, sich bewegen wird, wenn es hinunterfällt (Abb. 2). Der entwickelte Algorithmus, der die Beträge $trans_a$, $trans_b$, $trans_c$ der Verschiebungen der Eckpunkte A , B , C entlang der Oberflächennormale des Dreiecks berechnet, wobei der Achspunkt in baryzentrischen Koordinaten u , v , w bezüglich der Eckpunkte gegeben ist und die Stärke der Kollision in Form von t , lautet in C++:

```
double trans_a=(t+(t*(u+v)*2))*w;
double trans_b=(t+(t*(w+v)*2))*u;
double trans_c=(t+(t*(u+w)*2))*v;
double trans_all=t-(((trans_a*w)
+(trans_b*u)+(trans_c*v)));
trans_a+=(trans_all);
trans_b+=(trans_all);
trans_c+=(trans_all);
```

Teil der Verschiebung, den alle Punkte gemeinsam haben ($trans_all$). Anschließend wird der gemeinsame Anteil auf die relativen Anteile aufaddiert, sodass diese dann die Beträge der absoluten Verschiebungen ergeben.

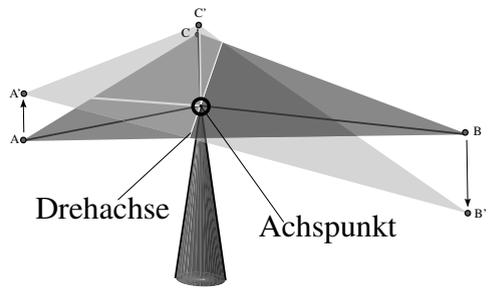


Abbildung 2: Kippverhalten eines Dreiecks

Die Berechnung des Rotationsanteils wird hier außer Acht gelassen, da dies bereits durch die Berechnung der Netzdynamik erreicht wird. Im ersten Teil des Algorithmus werden zunächst die relativen Anteile der Verschiebungen der Punkte berechnet (Stärke der Kippung nach oben/unten) und der

4 Preprocessing

Wenn alle Vertizes des Textils gegen alle Oberflächensegmente des 3D-Charakters geprüft werden und umgekehrt, ist die Komplexität der Kollisionserkennung $O(n^2)$. Zur Optimierung eignen sich hier keine klassischen Vorauswahlalgorithmen wie Boundingbox-Hierarchien, da sowohl die Geometrie des 3D-Charakters als auch die des Textils dynamisch sind und deshalb in jedem Frame die gesamte Hierarchie neu berechnet werden müsste. Der hier entwickelte Ansatz beruht auf dem Gedanken, dass nur Hautbereiche und Textilpartikel auf Kollision geprüft werden müssen, die einander bei der Initialisierung benachbart waren. In einem Preprocessing-Schritt werden für alle Partikel Arrays mit möglichen Kollisionspartnern angelegt. Diese werden dann zur Laufzeit in jeder Schleife einmal abgearbeitet. Auf diese Weise entfällt ein Großteil der Prüfungen. Des Weiter-

ren wurden die Programmteile, die in jeder Schleife aufgerufen werden, zur parallelen Ausführung als Threads implementiert, sodass die Last, verteilt auf mehrere Rechenkerne, parallel abgearbeitet werden kann.

5 Leistung und Ergebnisse

Auf einem System mit 4 Rechenkernen bei einer Taktfrequenz von 3.1 GHz konnte der Textilsimulator (Abb. 3) den physikalischen Zustand des Textils bei einer Bildrate von 20 FPS 40 mal pro Sekunde berechnen. Durch die parallele Ausführung konnte die Leistung bei vier Rechenkernen im Vergleich zur Ausführung auf einem einzigen Kern verdreifacht werden.

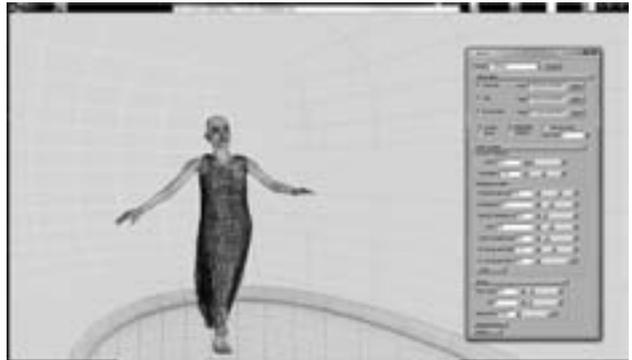


Abbildung 3: Ein Screenshot der GUI des Simulators

Das Preprocessing bewirkte eine zusätzliche Leistungssteigerung um das ca. 14,6 fache. Die Abb. 4 zeigt eine Tabelle mit Leistungsdaten. Der linke Teil der Tabelle zeigt einige gemeinsame Größen zweier Simulationsläufe. Der rechte Teil der Tabelle zeigt die Zahl der benötigten Kollisionsprüfungen und die jeweils erreichte Bildrate (FPS), einmal bei deaktiviertem und einmal bei aktiviertem Preprocessing.

Parameter	Value	Preprocessing disabled	
Character verteces	7337	Total checks per loop	61028707
Character triangles	9567	Average FPS on 4 cores at 3.1Ghz	1,33
Textile verteces	2598	Preprocessing enabled	
Textile quads	2464	Total checks per loop	2697488
Number of springs	7462	Average FPS on 4 cores at 3.1Ghz	19.5

Abbildung 4: Leistungsdaten im Vergleich

Literatur

- [BF08] M. Barnes und E. L. Finch. COLLADA Digital Asset Schema Release 1.5.0, 2008. http://www.khronos.org/files/collada_spec_1.5.pdf, 14.12.2012 zuletzt erreicht.
- [D⁺06] L. Desmecht et al. CAL3D 3D CHARACTER ANIMATION LIBRARY, 2006. <http://home.gna.org/cal3d/>, 14.12.2012 zuletzt erreicht.
- [MT97] T. Möller und B. Trumbore. Fast, minimum storage ray-triangle intersection. *Journal of graphics tools*, 2(1):21–28, 1997.
- [MTTA00] N. Magnenat-Thalmann, D. Thalmann und B. Arnaldi. *Computer Animation and Simulation 2000: Proceedings of the Eurographics Workshop in Interlaken, Switzerland*. Springer, 2000.
- [Pro95] X. Provot. Deformation constraints in a mass spring model to describe rigid cloth behavior. *Graphics Interface '95*, 1995.

Computational Fluid Dynamics within LibGeoDecomp

Stephan Helou

stephan.helou@ce.stud.uni-erlangen.de

Abstract: The major focus of this paper lies on the development and implementation of a lattice Boltzmann kernel for the LibGeoDecomp framework, an auto-parallelizing library for stencil codes. Different boundary conditions are derived, implemented and validated in two and three dimensions. The validation of the kernel and the boundary conditions is mainly done with the help of a Poiseuille flow. The latter is simulated in multiple ways, e.g. two different ways to incorporate force into the fluid simulation. In the three dimensional case additionally a shear flow simulation and a drag force simulation for a cylinder are performed and analyzed. Moreover, arbitrary shaped obstacles composed of basic geometries can be positioned in the flow.

1 Introduction and Motivation

The lattice Boltzmann method (LBM) is a widely used and approved method for solving problems in computational fluid dynamics. In contrast to the Navier Stokes approach the LBM solves the discretized lattice Boltzmann (LB) equation with a kinetic model. The aim of this paper is to implement a LB kernel for the LibGeoDecomp framework [1] with single-relaxation-time approximation, with different boundary conditions and fixed obstacle treatment. In the context of "Smart Life" it can be used as an analysis and verification tool for different systems and problems. Moreover the LBM can easily be parallelized and is therefore a good approach to model and simulate complex physical systems. For example the behavior of an aircraft wing before construction or the acoustics in a car can be simulated.

2 Lattice Boltzmann Method

One of the most popular approaches to solve the LB equation is done with the help of the so called Bhatnagar-Gross-Krook model [2]. The fully discretized version reads as follows

$$f_{\alpha}(\vec{x} + \vec{e}_{\alpha}\delta_t, t + \delta_t) - f_{\alpha}(\vec{x}, t) = -\frac{1}{\tau} \left[f_{\alpha}(\vec{x}, t) - f_{\alpha}^{(eq)}(\vec{x}, t) \right], \quad (1)$$

where f_{α} represents the particle distribution functions (PDF), \vec{e}_{α} the discrete velocity set, τ the lattice relaxation time and δ_t the time step size. For the D3Q19 model (3D, 19

PDFs) [2], the equilibrium distribution function can be calculated as

$$f_{\alpha}^{(eq)} = p \cdot \omega_{\alpha} \cdot \left[1 + \frac{3}{c^2} \vec{e}_{\alpha} \cdot \vec{u} + \frac{9}{2c^4} (\vec{e}_{\alpha} \cdot \vec{u})^2 - \frac{3}{2c^2} \vec{u} \cdot \vec{u} \right], \quad (2)$$

where $c = \delta_x / \delta_t$, with δ_x being the lattice space discretization and the weighting factor is w_{α} [2]. The lattice density p and the lattice velocity \vec{u} are given as

$$p = \sum_{\alpha} f_{\alpha}, \quad \vec{u} = \frac{1}{p} \sum_{\alpha} \vec{e}_{\alpha} f_{\alpha}. \quad (3)$$

Eq. (1) is usually solved in two steps,

$$\begin{aligned} \tilde{f}_{\alpha}(\vec{x}, t + \delta_t) &= f_{\alpha}(\vec{x}, t) - \frac{1}{\tau} \left[f_{\alpha}(\vec{x}, t) - f_{\alpha}^{(eq)}(\vec{x}, t) \right], \\ f_{\alpha}(\vec{x} + \vec{e}_{\alpha} \delta_t, t + \delta_t) &= \tilde{f}_{\alpha}(\vec{x}, t + \delta_t). \end{aligned} \quad (4)$$

the collision step and the streaming step, respectively. The post collision state is denoted by \tilde{f}_{α} . More on the LBM can be found for example in Sukop and Thorne [3].

2.1 Boundary Conditions

Different boundary conditions are implemented for the D2Q9 (2D, 9 PDFs) and D3Q19 model [3]. For solid walls one of the standard boundaries is the no-slip half way bounce-back scheme. Here, only the unknown populations are updated and reflected back into the cell they are coming from. This boundary is mainly used to simulate real walls. Another solid wall is the free-slip boundary condition. It is used to simulate smooth surfaces which do not decelerate the fluid. Here, the distribution functions are reflected only along the boundary normal. Furthermore the Zou He pressure and velocity boundary conditions are implemented in two [4] and three dimensions [5]. This method is based on the idea of bounceback of the non equilibrium part and only the unknown distributions are replaced. It is mainly used to simulate fluid inflow and/or outflow. In addition the velocity boundary condition can be used as a no-slip boundary for non moving walls if u_x , u_y and in the three dimensional case u_z is set to zero. When using the Zou He condition at perpendicular walls a special corner treatment has to be applied. More information on this type of boundary condition can be found in Zou and He [4] and in Hecht and Harting [5].

3 Implementation

The domain of the simulation is a grid of two (quadratic) or three (cubic) dimensional cells. In this `Cell` class all boundary conditions, the combined stream-collide step and additional helper functions are implemented. Furthermore an update function, which is called and executed during the simulation, is used as an interface to the `LibGeoDecomp`

solvers. Each cell is characterized by its own state e.g. fluid, obstacle or no-slip flag. Depending on this flag the cell gets updated. The flags at the boundaries, the obstacles and the LBM specific parameters are defined in a parameter file which is processed during the setup process. Fig. 1 gives a short overview of the implemented code.

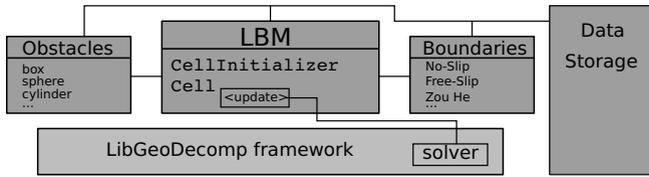


Figure 1: Overview of the implemented code.

4 Validation and Results

The LB kernel and especially the multiple boundary conditions are carefully validated and analyzed. For the two and three dimensional case a Poiseuille flow [3] is simulated in two different ways and verified. It is defined as

$$u_x = u_0 \left(1 - \frac{y^2}{a^2} \right); \quad u_y = 0; \quad (5)$$

where u_0 is the centerline velocity and a is the half width of the channel. In a first test scenario a Poiseuille flow is simulated in 16×32 cells. The x- boundaries are periodic and for the y- boundaries the Zou He velocity condition in their no-slip configuration. The flow is driven by a gravitational force. The analytic and simulated velocity profile are plotted in Fig. 3. Here the simulation results fit up to 14 digits with the analytic solutions and is exactly zero at the boundaries. Having velocity and/or pressure conditions at all boundaries, results up to machine accuracy are obtained when using the incompressible LB model [4]. A well known unsteady flow, the so called Kármán Vortex Street [3], is also simulated shown in Fig. 2. This turbulent flow can be achieved when an obstacle e.g. a cylinder is placed asymmetric in the fluid and a high Reynolds number is chosen.

In the three dimensional case also Poiseuille flow is validated. Using a 32^3 domain with

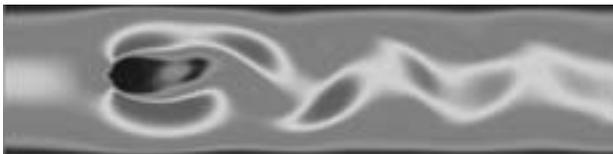


Figure 2: Karman Vortex Street.

periodic boundaries in x- and y- direction and the velocity no-slip condition on the top and

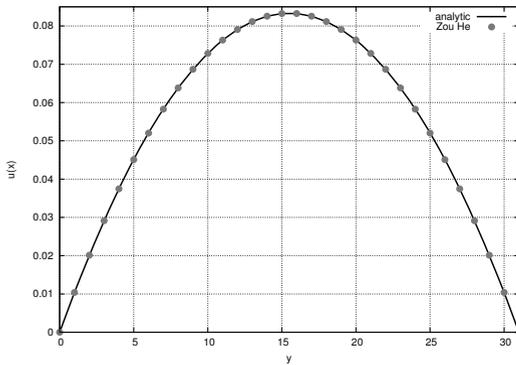


Figure 3: Poiseuille flow.

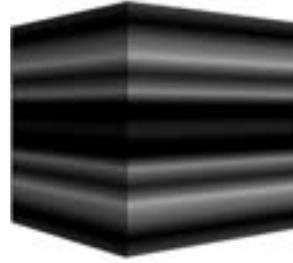


Figure 4: Shear flow simulation.

bottom of the channel, the same results as in the two dimensional example are achieved. Another test case is a shear flow simulation, which is described in Hecht and Harting[5]. In this scenario the x - and y - boundaries are periodic. At the top and bottom boundaries, the velocity conditions are imposed with a shear velocity $u_x = \pm 0.2$, while u_y and u_z are set to zero. As depicted in Fig.4, a linear flow profile with accuracy up to floating point precision is achieved. Furthermore a drag force simulation for a cylinder is performed and analyzed. Therefore the force on a sphere with radius r is calculated with Stokes law [6] and the corrected formula by Oseen [6]. These analytic results are compared with the forces gained by simulations, where the sphere is fixed in the center of the domain. Only the τ values and the domain size varies. Using a $\tau = 1.7$ the error ranges from 39.74% at a domain size of 40^3 cells and goes down to an error of $\sim 1\%$ at a domain size of 140^3 cells, compared to the Oseen solution.

References

- [1] Schäfer, Andreas, Fey, Dietmar: *LibGeoDecomp: A Grid-Enabled Library for Geometric Decomposition Codes*. Proceedings of the 15th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface, 2008.
- [2] Dazhi Yu, Renwei Mei, Li-Shi-Luo, Wei Shyy: *Viscous flow computations with the method of lattice Boltzmann equation*. Aerospace Sciences 39, 2003.
- [3] Michael.C. Sukop, Daniel T. Thorne, Jr.: *Lattice Boltzmann Modeling An Introduction for Geoscientists and Engineers*. Lawrence Livermore National Laboratory 1993.
- [4] Qisu Zou, Xiaoyi He: *On pressure and velocity boundary conditions for the lattice Boltzmann BGK model*. American Institute of Physics 1997.
- [5] Martin Hecht, Jens Harting: *Implementation of on-site velocity boundary conditions for D3Q19 lattice Boltzmann*. Journal of Statistical Mechanics 2010.
- [6] Jan Götz: *Massively Parallel Direct Numerical Simulation of Particulate Flows*. Phd Thesis, Universität Erlangen-Nürnberg 2012.

ARchäologie

Die Vermittlung von kulturellem Erbe mithilfe von Augmented Reality

David A. Plecher

plecher@in.tum.de

Fachgebiet Augmented Reality

Technische Universität München, Fakultät Informatik

1 Motivation

Augmented Reality (AR) ermöglicht die Einbettung virtueller Objekte in die Realität durch die Verwendung von Displays jeglicher Art und erfreut sich seit einigen Jahren großer Beliebtheit. Leistungsfähige Smartphones in Kombination mit ihren hochauflösenden Displays und Kameras ebnen den Weg, um AR in den Alltag des Menschen zu integrieren. Neben den technischen Wissenschaften zieht besonders die Archäologie großen Nutzen aus der AR, da antike Statuen, Gebäude und Tempel an Ort und Stelle rekonstruiert werden können. Das Archeoguide-Projekt [VIK⁺02] in Olympia und Life Plus in Pompeii [PSO⁺05] sind daher schon frühe Zeugnisse der AR in der Archäologie. Diese Projekte konnten damals allerdings noch nicht auf die heutigen leistungsfähigen Smartphones zurückgreifen.

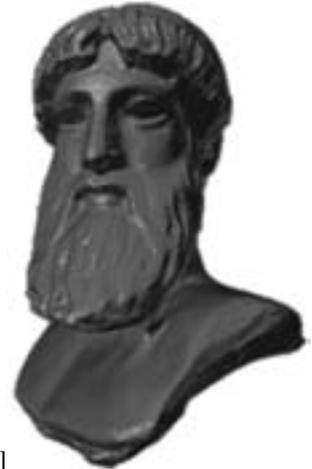
Diese Arbeit soll den Prozess einer realen Zeusstatue vom 3D-Scan bis zur Positionierung auf dem Marker am Ende dieses Beitrags - siehe Abb.5 - beschreiben.

2 Der 3D - Scan

Die Statue - Abb. 1a - wird durch zwei Bilderserien aufgenommen. Die erste Serie von ca. 20-30 Bildern zeigt die Statue von allen Seiten mit unterschiedlichen Details und auch die Position relativ zur Umgebung. Eine weitere Serie von 10-20 Bildern präsentiert den Blick aus der Vogelperspektive. Es ist insgesamt darauf zu achten, dass jedes Bild mindestens drei markante Punkte enthält, um die 3D-Rekonstruktion zu ermöglichen. Die Anwendung Autodesk 123Catch [Aut13] kommt hierbei zum Einsatz und berechnet aus dem gesammelten Material eine .obj - Datei inklusive der Texturen. Eine Nachbearbeitung in Blender entfernt Darstellungen der Umgebung und schließt Lücken in der virtuellen Statue. Das Ergebnis wird durch Abb. 1b dargestellt.



[a]



[b]

Abbildung 1: Reale und virtuelle Zeusstatue

3 AR Browser Junaio

Der AR Browser Junaio [Met13] von Metaio bringt AR auf das iPhone und iPad. Nach Download der App - siehe Abb. 2 - können durch Einscannen des entsprechenden QR-Codes vgl. Abb. 4 über den entsprechende Channel des Junaio Developers die Daten des Texts oder des 3D Objekts aus der Cloud oder vom FTP-Server geladen werden. Ein zugeordneter Marker wird von Junaio erkannt und bildet die Basis, um das 3D Objekt in die Realität einzubetten. Im Mittelpunkt des Markers entsteht sozusagen der Ursprung eines virtuellen Koordinatensystems, wodurch es möglich ist, das zugeordnete virtuelle Objekt durch Bewegung des Marker zu rotieren oder zu kippen. Der Benutzer sieht nun auf dem Display des iPhones/iPads seine Umgebung und die zusätzliche Information in 3D, sobald er die Kamera auf den Marker vgl. Abb. 5 richtet. Das Ergebnis lässt sich in Abb. 3 betrachten.



Abbildung 2: Junaio



Abbildung 3: Junaio at work

4 Ausblick

Die ARchäologie ist durch die reine Darstellung antiker Objekte noch nicht abgeschlossen. Zusatzinformationen wie Texte oder auch evtl. unterschiedliche Rekonstruktionen desselben Objekts können dargestellt werden. Es entsteht dadurch sowohl ein Mehrwert für den Archäologen an sich, als auch für den interessierten Laien. Eine Studie inwieweit sich insbesondere letztere Gruppe aufgrund der neuen Betrachtungs- und Informationsmöglichkeiten ihren Umgang damit verändert oder sich motiviert sieht sich genauer mit der Materie zu beschäftigen, soll sich dieser Arbeit anschließen. Dafür wird eine Zusammenarbeit mit einer Ausstellung oder Museum sowie einer geschichtlichen Zeitschrift angestrebt.

Literatur

- [Aut13] Autodesk. Turning Photos in 3D Modells, Jan 2013. <http://www.123dapp.com/catch> [Online; Letzter Zugriff 15-Jan-2013].
- [Met13] Metaio. Augmented Reality - Junaio, Jan 2013. <http://www.junaio.com> [Online; Letzter Zugriff 15-Jan-2013].
- [PSO⁺05] G. Papagiannakis, S. Schertenleib, B. O’Kennedy, M. Arevalo-Poizat, N. Magnenat-Thalmann, A. Stoddart und D. Thalmann. Mixing Virtual and Real scenes in the site of ancient Pompeii. *Computer Animation and Virtual Worlds*, 16(1):11–24, 2005.
- [VIK⁺02] V. Vlahakis, M. Ioannidis, J. Karigiannis, M. Tsotros, M. Gounaris, D. Stricker, T. Gleue, P. Daehne und L. Almeida. Archeoguide: An augmented reality guide for archaeological sites. *Computer Graphics and Applications, IEEE*, 22(5):52–60, 2002.

5 ARchäologie zum Ausprobieren



Abbildung 4: QR-Code zum Junaio-Channel

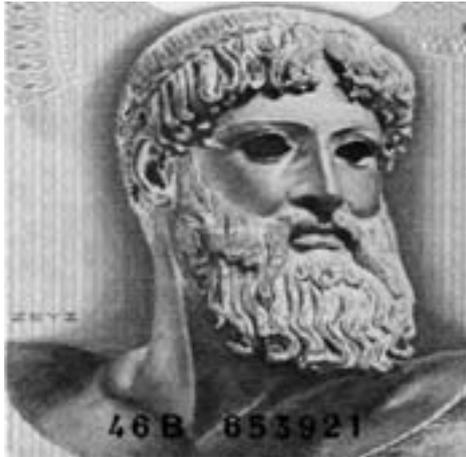


Abbildung 5: Marker: Ausschnitt eines 1000 Drachmen Scheins

Approximation von Reflektanzmessungen mit Vektorraummodellen und Latent Semantic Indexing

Ruben Reiser, B.Sc.
Fachhochschule Kaiserslautern
rure0001@stud.fh-kl.de

Abstract: In der Praxis werden Spektralfotometer eingesetzt, die mit einer kleinen Anzahl an Messwerten Reflektions-Merkmale von Materialien messen. In dieser Arbeit werden Verfahren des Information Retrieval verwendet, um Näherungen für solche Datensätze durch dicht vermessene Reflektanzfunktionen oder analytische Beleuchtungsgesetze zu bestimmen, die den Reflektions-Merkmalen des Materials weitestgehend entsprechen und für hochwertige Produktrenderings verwendbar sind.

1 Einleitung

In der Qualitätssicherung von Lacken und Farben werden in der Industrie Farbmessgeräte, sogenannte „Spektralfotometer“ verwendet, die schnell und einfach zu handhaben sind. Diese können die Reflektions-Merkmale von Materialien mit rund 20 Messpunkten erfassen. Die Daten können im Gegensatz zu den von MERL ([MPBM03]) erhobenen Daten nicht direkt für hochwertige Produktrenderings verwendet werden, da sie im Gegensatz zu den 1.458.000 Messpunkten der MERL-Daten nicht ausreichend viele Daten enthalten.

In dieser Arbeit wird der Ansatz verfolgt, Näherungen der „dünnen“ Messdaten mittels des „Latent Semantic Indexing“ (LSI) durch bekannte Reflektanz-Beschreibungen zu bestimmen. Diese Näherungen besitzen eine ausreichend hohe Qualität, um sie für Produktrenderings zu verwenden.

2 Grundlagen von Reflektanzfunktionen

Bidirektionale Reflektanzverteilungsfunktionen (BRDFs) beschreiben das Reflexionsverhalten von Oberflächen ([BB06]). In der Praxis werden für anisotrope Materialien vierdimensionale BRDFs und für isotrope Materialien dreidimensionale Funktionen verwendet. Die Eingabeparameter beschreiben eine Licht- und Betrachtungsrichtung, die mit dem Elevationwinkel θ_i und dem Azimutwinkel φ_i der Lichtrichtung und θ_o bzw. φ_o der Betrachtungsrichtung definiert werden. Diese Winkelpaare werden zu ω_i und ω_o zusammengefasst, wobei ω_i für die Lichtrichtung und ω_o für die Betrachtungsrichtung verwendet wird.

BRDFs beschreiben das Verhältnis von eingehender Strahlungsdichte E_i und ausgehender Strahlungsdichte L_o :

$$f_r(\theta_i, \varphi_i, \theta_o, \varphi_o) = f_r(\omega_i, \omega_o) = \frac{L_o(\omega_o)}{E_i(\omega_i) \cdot \cos(\theta_i)}$$

Damit physikalisch korrekte Werte generiert werden, müssen Reflektanzfunktionen zwei Eigenschaften erfüllen: Zum einen muss der Energieerhaltungssatz gelten. Das bedeutet, dass eine Oberfläche nicht mehr Licht reflektieren darf als von der Lichtquelle auf sie eingestrahlt wird. Zum anderen muss die Helmholtz-Reziprozität gegeben sein. Das bedeutet, dass das Vertauschen von Licht- und Betrachtungsrichtung keinen Unterschied macht:

$$f_r(\omega_i, \omega_o) = f_r(\omega_o, \omega_i)$$

BRDFs können in zwei Gruppen kategorisiert werden. Die „Samples“ sind vermessene BRDFs und die analytischen BRDFs, welche mit Beleuchtungsgesetzen versuchen, die physikalischen Reflektionsverhalten nachzuempfinden, wie Ward-Dür ([GD10]) und Ashikhmin ([AS00]). Die BRDFs können in Abhängigkeit der Eingabeparameter in verschiedenen Farbräumen Ergebnisse liefern. Geeignete Farbräume sind RGB, XYZ, $L^*a^*b^*$ oder spektrale Daten ([HP11]).

3 Motivation

In der Praxis werden Spektalfotometer verwendet, mit denen es möglich ist, die Reflektionsmerkmale von Materialien zu messen. Diese Geräte generieren Messungen in sehr kurzer Zeit und sind dadurch leicht zu handhaben. Aufgrund von wenigen Messpunkten ist der Vorgang schnell durchzuführen. Diese dünnen Messdaten reichen für die Qualitätssicherung vollkommen aus. Es kann beispielsweise überprüft werden, ob der Lack des Zulieferers die gewünschten Reflexionseigenschaften besitzt. Für die Produktvisualisierung im Marketing sind diese Daten nicht direkt einsetzbar. Die gängige Praxis ist, dass man in der Visualisierung für ein analytisches Beleuchtungsgesetz Parameter so einstellt, dass die gewünschten Reflexionseigenschaften des Materials vorliegen. Diese Anpassung der Parameter kann mithilfe eines nicht-linearen Optimierungs-Algorithmus rechnergestützt durchgeführt werden ([Häl11]). Ein Nachteil dieses Ansatzes ist, dass diese Algorithmen gute Startwerte benötigen, da sonst eine starke Abweichung des erzielten Materials entsteht. Darüber hinaus treten häufig Konvergenzprobleme auf, so dass diese Vorgehensweise nicht automatisiert durchgeführt werden kann.

Die dünnen Messdaten können analog zu einem Text-Retrieval als Suchanfrage verwendet werden: die Dokumentenmenge ist in diesem Fall ein Vektorraum-Modell für Reflektanzfunktionen ([ABL10]). Durch „Latent Semantic Indexing“ ([BBD99]) wird eine geordnete Liste von Materialien der Suchanfragen generiert.

4 Latent Semantic Indexing

In dieser Arbeit wird ein Vektorraum-basiertes „Information Retrieval“ verwendet ([LM06]). Es wird eine Dokument-Term-Matrix aufgebaut, in der die Spalten einzelne Dokumente sind und die Zeilen die Werte zu den Dokumenten darstellen. Dadurch wird eine $m \times n$ Matrix generiert. Die Anfrage wird als Vektor repräsentiert und mithilfe des Kosinus-Maßes oder andere geeignete Abstandsmaße wird die Ähnlichkeit des Vektors zu den Dokumenten berechnet ([BBD99]). Durch die Verwendung der Singulärwertzerlegung (SVD) wird die Matrix A zu einer Matrix A_k approximiert. Aufgrund dieser Approximation der Matrix erhält man semantische Zusammenhänge, die man durch einfache Abstandsberechnung nicht erhalten würde.

In der Arbeit repräsentiert, die Dokumente die BRDFs in der Matrix, und die Zeilen repräsentieren den Messwert zu den Winkeln. Durch die Verwendung unterschiedlicher Farbräume können die Ergebnisse zum Teil variieren. Der $L^*a^*b^*$ -Farbraum hat den Vorteil, dass die euklidischen Abstände mit den empfundenen Farbabständen übereinstimmen. Dies führt dazu, dass SVD und PCA zu visuell besseren Ergebnissen führen.

5 Ergebnisse

Die Ergebnisse zeigen, dass Materialien gefunden werden, die der Suchanfrage entsprechen. Selbst bei einer Datenbank mit wenigen BRDFs lassen sich gute Treffer erzielen. In Abbildung 1 wird auf der linken Seite ein hochauflösendes vermessenes MERL-Material dargestellt. Auf der rechten Seite wird ein analytisches BRDF vom Typ Lambert benutzt, welches die Suchanfrage als Ergebnis für das MERL Pink-Plastic darstellt. Dies ist auf der linken Seite veranschaulicht. Darüber hinaus zeigt sich, dass man durch Suchanfragen Materialien klassifizieren kann. Dies zeigt sich dadurch, dass Suchanfragen mit metallische Materialien überwiegend BRDFs, die Metallen nachempfunden sind, als Ergebnis anzeigen. Kunststoffe oder Holz liefern ebenfalls Ergebnisse, die zur Klassifizierung heran gezogen werden können.

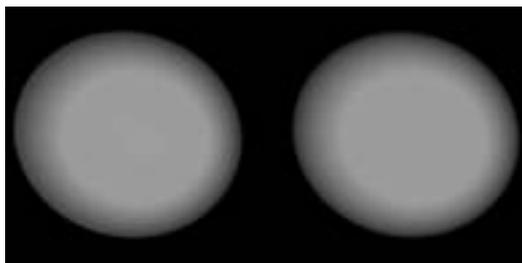


Abbildung 1: Pink-Plastic aus [MPBM03], links Originalmessdaten, rechts Lambert-BRDF. Als Renderer wird pbrt [PH10] verwendet, das um die OpenBRDF-Funktionalität erweitert ist.

6 Fazit und Ausblick

Die Qualität der Ergebnisse ist von der Quantität und Qualität der BRDF-Datenbank abhängig. Zu untersuchen ist, ob mithilfe eines iterativen Vorgehens die Größe der Datenmenge reduzierbar ist. Voraussichtlich können die nichtlinearen Parameterapproximations-Algorithmen kombiniert werden mit einer automatisierbaren Startwert-Erzeugung, die aus diesem vorgestellten Ansatz entnommen werden kann. Des Weiteren wird an einem intuitiven Interface-Konzept gearbeitet, um das betrachtete Vorgehen in den Praxis-Workflow zu integrieren.

7 Danksagung

Diese Arbeit wurde vom Bundesministerium für Bildung und Forschung im Rahmen von FHprofUnt im Projekt OpenBRDF unterstützt, Förderkennzeichnung „1733X09“. Ein besonderer Dank geht auch an Prof. Dr. Manfred Brill für die hervorragende Betreuung der Masterarbeit.

Literatur

- [ABL10] Sebastian Alberternst, Manfred Brill und Markus Laininger. OpenBRDF - An Open Toolkit for Reflection and Transmittance Distribution Functions. In *High Performance Graphics*, 2010.
- [AS00] Michael Ashikhmin und Peter Shirley. An anisotropic phong BRDF model. *J. Graph. Tools*, 5(2):25–32, 2000.
- [BB06] Manfred Brill und Michael Bender. *Computergrafik*. Hanser, 2006.
- [BBD99] Michael W. Berry, Murray Browne und Jack J. Dongarra. *Understanding Search Engines - Mathematical Modeling and Text Retrieval*. SIAM, 1999.
- [GD10] Geisler-Moroder, David und Dür, Arne. Bounding the Albedo of the Ward Reflectance Model. In *High Performance Graphics*, 2010.
- [HP11] R. Hung und M. Pointer. *Measuring Colour*. Imaging Science and Technology. Wiley, 2011.
- [Hä11] Stefan Häfner. Approximation vermessener Reflektanzfunktionen durch nichtlineare Parameterschätzung. *GI-Edition: Lecture Notes in Informatics*, Seiten 207–210, 2011.
- [LM06] Amy N. Langville und Carl D. Meyer. *Googles PageRank and Beyond: The Science of Search Engine Rankings*. Pinceton University Press, 2006.
- [MPBM03] Wojciech Matusik, Hanspeter Pfister, Matthew Brand und Leonard McMillan. Efficient Isotropic BRDF Measurement. In Daniel Cohen-Or Per Christensen, Hrsg., *Eurographics Symposium on Rendering*. Eurographics, 2003.
- [PH10] Matt Pharr und Greg Humphreys. *Physically Based Rendering: From Theory To Implementation*. Morgan Kaufmann, 2010.

Positionsabhängige perspektivische Darstellung von 3D-Inhalten auf mobilen Endgeräten

Kai Stenkamp

Hochschule Reutlingen
Deutsches Forschungszentrum für Künstliche Intelligenz GmbH
stenkamp@icloud.com

Art der Arbeit: Master-Thesis
Betreuer der Arbeit: Prof. Dr. rer. nat. Uwe Kloos (Hochschule Reutlingen),
Dr. sc. nat. Hilko Hoffmann (DFKI, Saarbrücken)

Abstract: Dieser Artikel beschreibt die Möglichkeit, eine dreidimensionale Illusion auf einem handelsüblichen zweidimensionalen Display zu erzeugen, ohne dafür auf Hilfsmittel wie beispielsweise eine Brille zurückzugreifen. Zur Erzeugung der Illusion findet die in mobilen Geräten verbaute Sensorik Verwendung. Defizite, die sich durch den eingeschränkten Blickwinkel oder Verdeckung der verbauten Kamera ergeben, werden hierbei durch den Lagesensor ausgeglichen.

1 Einleitung

Dreidimensionale Abbildungen ermöglichen das Eintauchen in eine realistisch anmutende, virtuelle Welt. In den vergangenen Jahren öffneten sich mobile Endgeräte vom Businessmarkt gegenüber dem Endkundenmarkt. An aktuellen Beispielen wie den Smartphones und Tablet-PCs ist ein Trend im Endkundensegment hin zu mobilen Endgeräten erkennbar. Diese stellen heutzutage multimediale Funktionsgeräte dar. Laut dem Marktforschungsunternehmen Gartner Inc. wird die Anzahl der Tablet-PCs von heute ca. 47 Milliarden auf ca. 148 Milliarden im Jahr 2015 heranwachsen¹. In diesem Artikel wird eine Projektionsmethode gezeigt, die es einer einzelnen Person ermöglicht, eine dreidimensionale Szene zu betrachten. Die Projektion bzw. die Szene wird durch die Bewegung des Gerätes oder des Kopfes angepasst, wodurch der dreidimensionale Eindruck ohne das Tragen einer Brille oder ähnlichen Hilfsmitteln ermöglicht wird.

2 Forschungsstand

Ware et al. [Wa93] stellten 1993 die „Fish Tank Virtual Reality“ Technik vor, welche eine perspektivische Projektion einer dreidimensionalen Szene bezogen auf die Kopfposition des Betrachters vornimmt. Die mechanisch getrackte Kopfposition ergibt

¹ <http://www.gartner.com/it/page.jsp?id=1800514>

den Betrachtungswinkel, der zur Berechnung der Projektion verwendet wird und durch kontinuierliche Anpassung eine Immersion erzeugt. Lee [Le08] beschrieb die Möglichkeit, blickwinkelabhängige Ausgaben mit der Wii Konsole von Nintendo zu erzeugen. Mithilfe einer mit Sensoren bestückten Brille und der Kopfposition sowie der abgeschätzten Distanz der Brille zum Ausgabegerät wird die durch die Kopfbewegung ausgelöste Veränderung des Betrachtungsfeldes eine Illusion der räumlichen Tiefe erzeugt. Francone und Nigay [FN11] haben HCP (Head-Coupled Perspective) auf mobilen Endgeräten implementiert. Dabei nutzen sie für das Bestimmen der Kopfposition die in mobilen Endgeräten verbauten Frontkameras. Die Anwendung verschiebt bei Positionsveränderungen den Blickpunkt der 3D-Szene in die entgegengesetzte Richtung; daraus resultiert eine dreidimensionale Illusion. Der räumliche Eindruck entsteht nicht vor dem Gerät, sondern wird durch das Bild, welches das Gerät wiedergibt, selbst erzeugt.

3 Konzeption

Die Anwendung ist - um eine Plattformunabhängigkeit zu erreichen - modular aufgebaut und größtenteils in HTML5 realisiert. Sie gliedert sich in die Module Tracking und Repräsentation. Entwickelt wird auf dem Apple iPad 2 und dem Acer Iconia A500, welche mit den Betriebssystemen Apple iOS und Android arbeiten. Für das Tracking finden der Lagesensor und die in den Geräten verbaute Frontkamera Verwendung. Dabei erlaubt der Lagesensor die Messung der Kippbewegung des Gerätes über die drei Achsen. Dies geschieht durch das messbare Drehmoment des Kreiselkompasses, wenn dessen ursprüngliche Lage verändert wird. Aus dem Drehmoment kann der Winkel bestimmt werden, um den das Gyroskop gekippt wurde. Lagesensoren benötigen keinen festen Bezugspunkt und sind daher unabhängig nutzbar. Nachteilig ist jedoch, dass kein fester Ursprung, zu dem die Lage in Relation gesetzt werden kann, existiert. Mit der Frontkamera wird die Gesichtspose ermittelt. Unter perspektivischer Darstellung wird die Möglichkeit verstanden, dreidimensionale Objekte auf einer zweidimensionalen Fläche so abzubilden, dass ein räumlicher Eindruck entsteht. Die Projektion erfolgt Off-Axis mit einem asymmetrischen Frustum. Dabei trifft der Betrachtungswinkel nicht den Mittelpunkt des Displays, sondern ist entsprechend der Blickrichtung verschoben. Der Ursprung des Displays wird am neuen Schnittpunkt ausgerichtet. Bewegt der Benutzer seinen Kopf, so wird die Parallaxe der virtuellen Kamera einer 3D-Szene angepasst.

4 Umsetzung

Zum Einsatz kommt hierfür die echtzeitfähige Open Source Computer Vision Library (OpenCV). Die Gesichtsdetektion erfolgt nach dem Viola-Jones-Verfahren [VJ04] mit der auf Lienhard et al. [Li03] zurückzuführenden Weiterentwicklung. Für das Viola-Jones-Verfahren werden weiche, auf Schwellwerten basierende Klassifikatoren, die sogenannten „Haar-ähnlichen“ Merkmale verwendet. Mit OpenCV werden bereits Trainingsdaten, die mit mehreren hundert Gesichtern oder Augen trainierten Haar-Klassifikatoren bestückt sind, ausgeliefert. Für die Gesichtserkennung werden die Bild-

rohdaten in Grauwertdaten umgewandelt, in Portrait-Ausrichtung gebracht und gespiegelt. Die Klassifizierung wird durch bereitgestellte OpenCV Methoden ausgeführt, welche nach dem größten gefundenem Gesicht abbricht. Nach der Gesichtsdetektion kann die Augenposition in einem weiteren Iterationsschritt ermittelt werden. Zurückgeliefert wird ein Vektor, welcher die x- und y-Koordinaten wie auch die Breite und Höhe des gefundenen Gesichts enthält. Dieser Vektor wird an das JavaScript Tracking Modul übermittelt, welches die weiteren Berechnungsschritte ausführt. Durch die Bewegung des Geräts und den geringen Blickwinkel (FOV) der Frontkamera sowie auch durch die hierdurch wechselnde Beleuchtungssituation führt dies zu häufigem Positionsverlust. Daher wird auf den Lagesensor der Geräte zurückgegriffen. Wird in einem Zeitfenster von 350 Millisekunden kein Gesicht mehr detektiert, kommt der Lagesensor zum Einsatz. Der Lagesensor liefert für jede der drei Achsen einen Wert zwischen -10 und 10, welcher aus Gründen des mangelnden Bezugspunkts einen Unterschied zum Kamerawertebereich darstellt. Die aktuellen Daten des Lagesensors werden mit denen zum Zeitpunkt des Positionsverlusts des Gesichts subtrahiert und in den Koordinatenbereich der Szene transformiert. Das sich daraus ergebene Delta wird auf die letzte bekannte Blickrichtung, welche aus dem Kamera-Tracking resultierte, addiert. Dieses Tracking Modul liefert nach der ersten erfolgreichen Detektion durchgängig einen Vektor an das zur Anzeige bereitstehende Modul. Die abgeschätzte Kopfposition wird zum Bilden einer Transformationsmatrix benutzt. Um einen 3D-ähnlichen Effekt zu erzielen wird auf die Off-Axis Projektion zurückgegriffen. Zur grafischen Repräsentation der 3D-Szenen wird XML3D verwendet. Mit XML3D können dreidimensionale Inhalte durch W3C Standards beschrieben und in HTML5 ohne das Nutzen zusätzlicher Erweiterungen eingebunden und dargestellt werden [So10]. Die in XML3D generierte Szene wird zum Betrachtungspunkt des Benutzers verzogen. Für das Rendern kommt WebGL zum Einsatz. Weder WebGL noch XML3D liegt eine erweiterte Projektionsmatrix, welche die Verzerrung der Ausgabe ermöglicht, zugrunde. Diese wurde implementiert und durch die zusätzliche Anpassung des Kameraursprungs erweitert. Abbildung 1 zeigt die Darstellungsform zu verschiedenen Gerätepositionen.

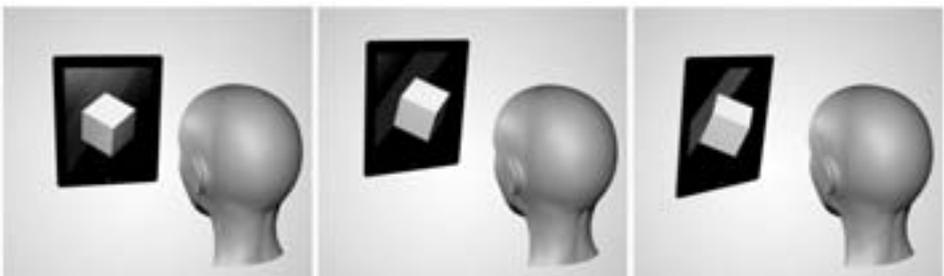


Abbildung 1: Angepasste Darstellung in Abhängigkeit des Blickwinkels zum Gerät

Im Zuge der Umsetzung zeigten sich deutliche Performanceunterschiede zu Lasten der Android Plattform. Zusätzlich ist die Position der Frontkamera je nach Android Gerät und Hersteller unterschiedlich. Dadurch wäre eine Anpassung für jedes unterschiedliche Gerät erforderlich, außerdem wird die Verdeckung des Kameramoduls begünstigt. Infolgedessen wurde die Anwendung für die Android Plattform nicht weiter betrachtet.

5 Evaluierung

Die realisierte Anwendung zeigt einen Campus Plan der Universität des Saarlandes, welcher durch die Veränderung des Blickfeldes auch verdeckte Gebäude offenbart. In der abschließenden Expertenbefragung zeigte sich, dass erfahrene Benutzer das Gerät bewegen, nicht den Kopf oder die Augen. Da das Augen-Tracking durch einen weiteren Iterationsschritt eine schlechtere Performance aufweist und sich zeigte, dass die Benutzer kaum Augen- oder Kopfbewegungen ausführen, wurde vom Augen-Tracking abgesehen.

5 Fazit und Ausblick

In diesem Artikel wurde gezeigt, dass ein dreidimensionaler Eindruck auch auf handelsüblichen Bildschirmen entstehen kann. Durch die Bewegung des mobilen Endgerätes wird beim Benutzer ein Zugehörigkeitsgefühl in der Szene vermittelt. In der abschließend erfolgten Expertenbefragung wurde der Mehrwert, welcher durch den zusätzlich zum Gesichts-Tracking genutzten Lagesensor entsteht, bestätigt. Diese Kombination ist im Rahmen der angefertigten Master-Thesis erstmalig betrachtet worden. Ohne den Einsatz des Lagesensors wäre die Anwendung kaum alltagstauglich, da der Blickwinkel der Kamera zu sehr eingeschränkt ist. Durch die Adaption neuer Anwendungsfälle wird die Anwendung weiter entwickelt und gilt als zukunftsweisend. Denkbar ist auch der Einsatz bei umgebungsintelligenten Systemen wie beispielsweise Smart Homes. Des Weiteren werden Variationen auch im Hinblick auf die Performance von komplexeren Anwendungsfällen evaluiert. Es besteht die Möglichkeit, das Gesichts-Tracking in bestimmten Intervallen auszuführen und fast völlig auf den Lagesensor zurückzugreifen, um eine weitere Performance Steigerung zu erzielen. Dies erfordert allerdings eine Anpassungsroutine, um einen flüssigen Ablauf zu garantieren.

Literaturverzeichnis

- [FN11] Francone, J. und Nigay, L. (2011): *Using the user's point of view for interaction on mobile devices*. In 23rd French Speaking Conference on Human-Computer Interaction (IHM '11), ACM, New York, NY, USA 2011.
- [Le08] Lee, J. C. (2008): *Hacking the Nintendo Wii Remote In*: IEEE Pervasive Computing, Band 7, Ausgabe 3, IEEE 2008, S. 39– 45.
- [Li03] Lienhart, R. Kuranov, A. und Pisarevsky, V. (2003): *Empirical analysis of detection cascades of boosted classifiers for rapid object detection*, in: Michaelis, Bernd und Krell, Gerald (Hrsg): Pattern Recognition, Springer Verlag 2003, S. 297– 304.
- [So10] Sons, K. et al. (2010): *XML3D: interactive 3D graphics for the web*. Proceedings of the 15th International Conference on Web 3D Technology, NY, ACM 2010, S. 175–184.
- [VJ04] Viola, P. und Jones, M.J. (2004): *Robust real-time face detection*, International Journal of Computer Vision, Band 57, Ausgabe 2, Springer Verlag 2004, S. 137–154.
- [Wa93] Ware C., Arthur K., und Booth K. S. (1993): *Fish tank virtual reality*, Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems, New York, NY, USA, ACM 1993, S. 37–42.

A Real-Time Visualization Infrastructure

Jan Sutter, M.Sc.

University of Applied Sciences Kaiserslautern

Abstract: Nowadays, Cloud Computing is getting more and more prominent and the browser has become the most ubiquitous application. With the introduction of the WebGL standard, real-time rendering has now found its way into the World Wide Web and, therefore, the browser. The transfer into the Web, however, introduces a new set of requirements and difficulties. Not only do we need to achieve interactive frame rates within the browser, but the change from a single user to a multi-user environment requires an infrastructure that is capable of serving multiple concurrent, data intensive requests without becoming unresponsive or unavailable. Furthermore, the infrastructure needs to be scalable and has to integrate into already existing content creation pipelines. This master's thesis presents a scalable infrastructure that is used as the foundation to successfully deploy browser-integrated real-time visualizations and accompanying back-end management web applications.

1 Introduction

WebGL leverages the hardware accelerated power of OpenGL to embed three dimensional graphics into the browser. It uses JavaScript to incorporate high fidelity 3D scenes into web applications. In order to provide real-time renderings like product visualizations at large scale, however, a few key problems have to be addressed. First, despite locally run applications, where the major part of the software is installed and all necessary assets are stored on the user's file system, real-time rendering within the browser inherently requires all scene data to be sent to the client on every request. This does not only create a high load on the server-side but also requires the client to wait until all data is transferred, before the application can finally be used. Second, web applications are used concurrently. This requires that the system is able to handle multiple users and high data loads without becoming unresponsive or unavailable. Careful examination of cloud applications with a large user base reveals that their success is based on a strong and scalable infrastructure that can handle those multiple accesses and high data loads.

2 Solution

In this thesis an infrastructure is developed that solves the before mentioned set of problems. This infrastructure is designed to be horizontally scalable, handle high loads, store and distribute scene data and provide functionality as asynchronous web services. However, it is not an isolated solution for WebGL based real-time visualizations, but a more



Figure 1: The different layers of the infrastructure

decentralized approach to visualization applications. As seen in Figure 1 the infrastructure consists of three layers and a set of applications on top of it. At the bottom of the infrastructure is the data-storage, a NoSQL database, mongoDB, responsible for storing all scene data. The central part comprises a RESTful service oriented interface based on NodeJS. On top are three applications to import scenes, embed real-time rendered visualizations within a website, and manage the back-end and interactively create GLSL based materials.

2.1 Data Format and Storage

The structure and format of data and their persistence is one of the first decisions to make when designing an application. For an infrastructure this decision becomes even more important as the data is used and shared by multiple applications. Anticipating the need of each application and predicting all data an application wants to store is impossible. Thus, a data format for the infrastructure needs to be extensible and easy to parse and to write in order to avoid inconsistencies in import and export routines. The JSON format satisfies these requirements. Moreover, it is JavaScript’s native serialization format and therefore does not require WebGL applications to transform the data upon usage.

Storing scene data and assets is typically done using the file system, which, however, is not a viable option as a centralized and scalable data-storage. Databases, like mongoDB, on the other hand, are designed as a central data storage facility with the ability to be scaled according to the required amount of data to store and the amount of queries to serve. MongoDB does not only use the JSON format internally but its schema-less design allows every application to add their own data to an entity without interfering with other datasets or applications. This approach is similar to the strategy of Doboš and Steed in [DS12].

However, the presented infrastructure solves the data validation problem by abstracting database access through a dedicated web service. Additionally, the infrastructure is able to handle scenes of size greater than 16 MB, the internal document limit of mongoDB. This is achieved by the importer web service that extracts the scene's components. It stores images using mongoDB's GridFS and divides large meshes into multiple smaller ones that are stored as separate documents. A scene's description then only contains references to all of its parts, thus, allowing every component of the scene to satisfy the 16 MB document limit.

2.2 RESTfull Services

Designing an infrastructure that is usable in different environments and that supports different applications raises the question how functionality is implemented regarding scalability and re-usability. Web services have proven well suited for maintainable and scalable software systems. The second layer of the infrastructure is therefore composed of web services designed according to Service Oriented Architecture methodology. All services are implemented in a Representational State Transfer approach using NodeJS. As each service is self-contained and stateless this layer can be scaled across multiple hosts. Additionally, the asynchronous nature of NodeJS allows for concurrent distribution of huge scene data to multiple clients without becoming unresponsive or unavailable.

3 Proof of Concept

Based on the infrastructure a real-time visualization of a shaver of the German company Braun is embedded within their website. This scenario shows the usage of the infrastructure's components and applications. Embedding a WebGL based rendering of a scene requires three steps: *a*) importing the scene using the importer service; *b*) designing and applying GLSL based materials using the back-end application (see Figure 2); and *c*) embedding the viewer application on the website. Since the material designer and the client viewer use the same visualization components, the final view of the scene is equivalent to the rendering at the bottom of Figure 2. Because of the asynchronous nature of the infrastructure's components the scene, consisting of approximately 10^6 triangles, can be distributed to the clients without degradation of the overall responsiveness of the website and clients can navigate around even if the scene is not yet loaded completely.

4 Conclusion and Future Work

Approaching a more decentralized setup for visualization applications is feasible and necessary in order to provide browser integrated real-time renderings at large scale. Because of the chosen components the developed infrastructure is scalable and can be used in any



Figure 2: Back-end application, visualizing an object with approximately 10^6 triangles

application that supports the JSON format and can issue HTTP requests. The infrastructure provides a great foundation for functionality, like revision control systems for 3D data [DS12]. Functionality like simple ASCII data compression algorithms or more sophisticated methods like *Progressive Meshes* [Hop96], however, are not yet implemented, but actively investigated. It also remains to be researched on how the infrastructure can build a decentralized and service oriented hub of functionality that supports new applications to progressively enhance their functionality as ubiquitous computing and permanent Internet connections become the standard.

5 Acknowledgments

This work was supported by the German Federal Ministry of Education and Research by the grant #1733X09.

References

- [DS12] J. Doboš and A. Steed. 3D revision control framework. In *Proceedings of the 17th International Conference on 3D Web Technology, Web3D '12*, 2012.
- [Hop96] H. Hoppe. Progressive meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, SIGGRAPH '96*, New York, NY, USA, 1996. ACM.

Methode zur Evaluierung von Public Cloud IaaS Providern unter Berücksichtigung individueller Anforderungen

Adnan Ademovic, Denise Uslu, Christian Kovac, Betlul Acekeza, Adil Sahiner

Hochschule Reutlingen
Semesterprojekt im Bachelorstudiengang Wirtschaftsinformatik

Vorname.Nachname@Student.Reutlingen-University.DE

Betreuer der Arbeit:
Herr Michael Falkenthal M.Sc.
Herr Prof. Dr. rer. nat. Alfred Zimmermann

Abstract:

Die Entwicklung von Cloud Diensten beschleunigt sich und deren Nutzung nimmt zu. Ein Vergleich potentieller Provider ist unerlässlich, um einen geeigneten Provider zu wählen, der den individuellen Ansprüchen entspricht. Durch neue Anbieter und deren heterogene Geschäftsmodelle gewinnt der Markt an Komplexität. Bis dato gibt es kein standardisiertes Verfahren um Cloud Computing Provider in Bezug auf Infrastructure-as-a-Service (IaaS) miteinander zu vergleichen. Das Ergebnis des Semesterprojekts ist eine Evaluierungsmethode, die die Bewertungskomplexität reduziert und den Entscheidungsträger bei der Auswahl eines geeigneten Anbieters entlastet. Die Methode umfasst einen praxisorientierten Qualitätsmerkmal-Katalog zur Planung und eine Benchmark-Matrix zur Durchführung und Auswertung der Evaluierung. Das Resultat ist eine wiederhol- und nachvollziehbare Verfahrensweise zur Bewertung der Qualität von Cloud Diensten, basierend auf individuellen Ansprüchen an Public Cloud IaaS Provider.

1 Problemstellung und Zielsetzung

Da die Nachfrage der Anwendungsunternehmen nach Cloud Computing wächst, weitet sich der Markt an Cloud Computing Angeboten rapide aus. Dadurch wächst die Provider Anzahl und wird zunehmend unübersichtlich. Die Produktvielfalt wird unüberschaubar und heterogen, hinzukommen komplexe Unterschiede in den Preismodellen der einzelnen Anbieter. ([RZ12], S. 1)

Um den Kunden die Auswahl der verschiedenen Public Cloud Anbietern auf Infrastructure-as-a-Service (IaaS) Ebene zu erleichtern, bedarf es verlässlicher Ansätze. Diese setzen standardisierte Bewertungen der Qualität und Passbarkeit von Cloud-Diensten auf die individuellen Anforderungen eines Anwendungs-unternehmens voraus. [RZ12] untersuchen in ihrer Arbeit erfolgskritische Faktoren bei der Auswahl eines Cloud Anbieters. Deren Arbeit enthält keine systematische Vorgehensweise zur Ermittlung eines geeigneten Cloud Providers, sie fokussiert lediglich auf eine Zusammenstellung relevanter Qualitätsmerkmale für Cloud Angebote.

In dieser Arbeit wird der Katalog an Qualitätskriterien um ein Vorgehensrahmenwerk erweitert, sodass eine systematische Evaluierungsmethode entsteht. Die Anforderungen an die Evaluierungsmethode sind es, eine transparente und strukturierte Planung, Durchführung und Auswertung zu gewährleisten, um eine Evaluierung jederzeit wiederholen und nachvollziehen zu können. Des Weiteren soll die Methode den Anwendern eine geeignete Auswahl von Qualitätsmerkmalen bieten, um sie bei der Entscheidungsfindung zu unterstützen und das gemeinsame Verständnis zu erleichtern. Zusätzlich soll ein Rahmenwerk den Anwendern bei der Formulierung von messbaren Szenarien Hilfestellung leisten. Die Evaluierungsmethode soll die Bewertungskomplexität reduzieren und dadurch die Entlastung des Entscheidungsträgers bewirken. Im weiteren Vorgehen sollen die Kategorien, in denen die zuvor erwähnten Qualitätsmerkmale unterteilt sind, gewichtet werden. Letztlich vereinigt man die bewertenden Szenarien und die gewichteten Kategorien, damit aussagekräftige Gesamtwerte zu den jeweiligen Providern geliefert werden, um aus dem Ranking den geeigneten Anbieter zu bestimmen.

2 Evaluierungsmethode

Die Methode umfasst sieben Schritte, die wiederum in drei Phasen (Planung, Durchführung und Auswertung) gegliedert sind. Die sieben Schritte sind teilweise an die *Architecture Tradeoff Analysis Method* (ATAM) [CKK11] vom Software Engineering Institute der Carnegie Mellon University angelehnt.

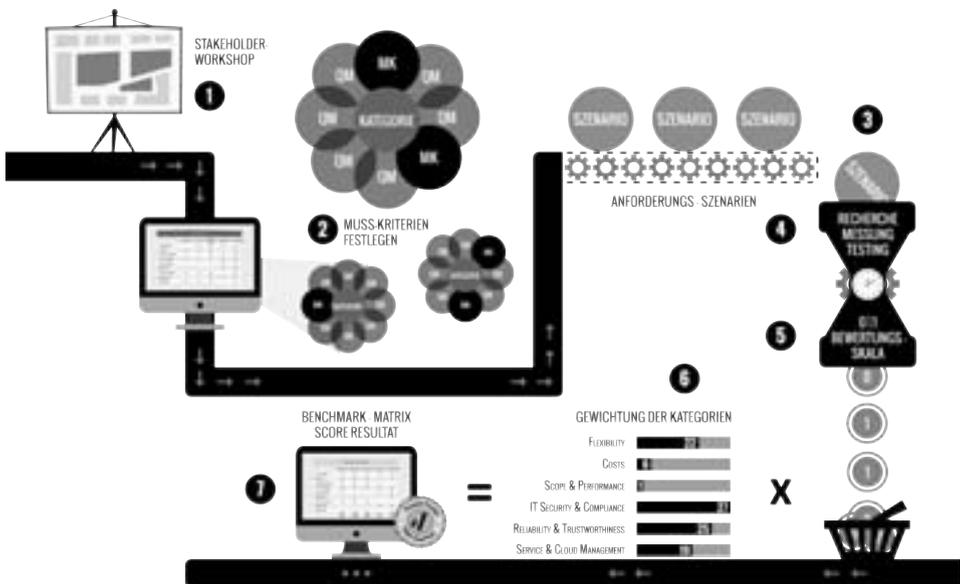


Abbildung 1: Methodischer Ablauf einer Evaluierung

Der erste Schritt der Evaluierungsmethode ist ein Stakeholder-Workshop. Dabei diskutiert man, anhand des Qualitätsmerkmal-Katalogs (QM-Katalog), in einem Brainstorming die individuellen Anforderungen an Public Cloud Provider.

Der Workshop dient als Ausgangsbasis für eine weitreichende Einarbeitung und Auseinandersetzung mit den Themen Cloud Computing und IT-Infrastruktur ([CKK11], S. 59). Der QM-Katalog setzt sich aus 30 Qualitätsmerkmalen zusammen, die wiederum in 6 Kategorien unterteilt sind. Diese Kategorien wurden im Rahmen unseres Semesterprojekts auf Gemeinsamkeiten überprüft. Dabei konnten wir feststellen, dass die QMs untereinander disjunkt sind. Folglich ist sichergestellt, dass die Kategorien voneinander unabhängig sind. Die Auswahl der QMs, samt der Kategorien, stammt aus einer wissenschaftlichen Arbeit des *Lehrstuhls für IKM der TU Berlin* [RZ12]. Die Autoren differenzieren zwischen „IaaS-spezifischen Qualitätsmerkmalen“ und „Service-unabhängigen Qualitätsmerkmalen“. Die Service-unabhängigen Qualitätsmerkmale können für die Evaluierung aller drei Cloud Servicemodelle (Infrastructure-, Plattform- und Software-as-a-Service) verwendet werden. IaaS-spezifische Qualitätsmerkmale werden hingegen ausschließlich für das Service Model Infrastructure in Betracht gezogen. ([RZ12], S. 3)

Der zweite Schritt besteht darin, anhand der QMs, die Muss-Kriterien festzulegen. Muss-Kriterien sind QMs, deren Nichteinhaltung so gravierend ist, dass der potentielle Provider sofort aus dem Auswahlverfahren ausgeschlossen wird.

Die Notwendigkeit des dritten Schrittes besteht darin, die unpräzisen Anforderungen aus dem Stakeholder-Workshop und den QMs in messbare Szenarien zu überführen. QMs alleine können ohne ein gemeinsames Verständnis aller Stakeholder unterschiedlich interpretiert werden. Die Überführung von QMs in Szenarien, stellt ein gemeinsames eindeutiges Verständnis der formulierten Anforderungen an einen IaaS Provider sicher. Da keine Schulungen und Zertifikate nötig sind, lassen sich Szenarien kostengünstig und effizient erstellen ([CKK11], S. 52 ff). Der QM-Katalog dient zusätzlich als Rahmenwerk für die Formulierung von Anforderungsszenarien. Dabei gibt er Aufschluss über die sechs Bestandteile von Szenarien, die sich wie folgt unterteilen: Stimulus (Auslöser), Source (Quelle), Environment (Umgebung), Artefakt (Systembestandteil), Response (Antwort) und Response Measure (Antwortmetrik). Darüber hinaus unterscheidet man drei Typen von Szenarien: Anwendungsszenarien, Änderungs-szenarien und Stress- und Grenzszenarien ([ST09], S. 63f). Letztlich operationalisieren Szenarien Qualitätsmerkmale und machen sie messbar ([ST09], S. 62). Die jeweiligen Szenarienbestandteile hält man im QM-Katalog fest. Sie ermöglichen die Formulierung eines vollständigen Szenarios. Zusätzlich extrahiert man den Referenzwert (Sollwert) aus den formulierten Szenarien. Anzumerken ist an dieser Stelle, dass es dem Anwender überlassen bleibt, wie viele Szenarien er zu einem QM beschreibt.

Anschließend beginnt im vierten Schritt der Methode die Durchführungsphase, bei der die Szenarien getestet, recherchiert und gemessen werden. Der dabei entstehende Aufwand variiert je nach Anforderung und kann deshalb nicht pauschal angegeben werden.

Die notierten Messergebnisse werden im fünften Schritt anhand einer Ordinalskala mit (0) oder (1) bewertet, da lediglich die vollständige Erfüllung der Mindestanforderung in Betracht gezogen wird. Die Bewertung (0) steht für *nicht erfüllt* und die (1) für *erfüllt*, d.h. der Ist-Wert entspricht dem Soll-Wert.

Im sechsten Schritt multipliziert man die bewerteten Messergebnisse in jeder Kategorie mit einer entsprechenden Gewichtung. Unsere Methode gibt vor, insgesamt 100 Punkte auf die 6 Kategorien zu verteilen. Die exemplarisch angewandte Gewichtung in der Benchmark-Matrix orientiert sich ansatzweise an der wissenschaftlichen Arbeit *Developing a Cloud Provider Selection Model* ([RWZT11], S. 168), der TU Berlin.

Abschließend werden in der Auswertungsphase, im siebten Schritt der Methode, die zuvor multiplizierten Werte summiert. Die resultierende Gesamtsumme aus den 6 Kategorien bzw. den 30 QMs liefert in der Benchmark-Matrix das maßgebende Score-Resultat. Die nach individuellen Anforderungen untersuchten Public Cloud Provider werden in einem Ranking aufgeführt. Dieses ermöglicht den Anwendern die verschiedenen Cloud Provider auf ihre getesteten Anforderungen zu vergleichen und somit auszuwählen, welcher Cloud Anbieter den Ansprüchen am besten entspricht.

3 Zusammenfassung und Ausblick

Das Resultat unseres Semesterprojekts ist eine aussagekräftige Nutzwertanalyse in Form einer Methode zur Evaluierung von Public Cloud IaaS Provider unter Berücksichtigung individueller Anforderungen. Die Methode umfasst einen praxisorientierten QM-Katalog zur Planung und eine Benchmark-Matrix zur Durchführung und Auswertung der Evaluierung. Ein Rahmenwerk hilft Anforderungsszenarien zu beschreiben und schafft damit gleiche Testing-Voraussetzungen. Letztlich werden mit Hilfe eines Bewertungssystems die Szenarien-Messwerte bewertet und diese Werte mit einer praxisrelevanten Gewichtung der 6 QM-Kategorien multipliziert. Eine Score Grafik veranschaulicht das Ranking, der evaluierten Provider, um daraus den geeigneten Anbieter auszuwerten.

Die Methode dient als Grundlage für ein anknüpfendes Projekt, in dem die *IBM Smart Cloud Enterprise* evaluiert wird.

- [RZ12] Repschläger, J., Zarnekow, R.: *Erfolgskritische Faktoren und Kundensegmente im Cloud Computing*, Technische Universität Berlin, IKM Lehrstuhl, 2012
- [CKK11] Clements, P., Kazmann, R., Klein, M.: *Evaluating Software Architectures – Methods and Case Studies*, 9th Printing, Addison-Wesley, 2011
- [ST09] Starke, G.: *Effektive Software-Architekturen*, 4. Aufl., München: Carl Hanser Verlag, 2009
- [RWZT11] Repschläger, J., Wind, S., Zarnekow, R., Turowski, K.: *Developing a Cloud Provider Selection Model*, In Lecture Notes in Informatics (LNI) –Proceedings. Series of the Gesellschaft für Informatik (GI). Volume P-190. (S. 163-175). Bonn: Gesellschaft für Informatik, 2011

Service Technologien

SOA Integrationsmodell

Anders, Däubler, Hick, Krajcevic, Maier, Stiefel

Fakultät Informatik
Hochschule Reutlingen
Studiengänge Wirtschaftsinformatik
Alteburgstraße 150
72762 Reutlingen

Franziska.Anders@Student.Reutlingen-University.DE
Michael.Daeubler@Student.Reutlingen-University.DE
Andreas_Kolja.Hick@Student.Reutlingen-University.DE
Denis.Krajcevic@Student.Reutlingen-University.DE
Philipp.Maier@Student.Reutlingen-University.DE
Daniel.Stiefel@Student.Reutlingen-University.DE

Abstract: Diese Ausarbeitung beschäftigt sich mit der Frage, wie ein Integrationsmodell aus den Referenzarchitekturen von OpenGroup und OASIS, sowie dem OASIS Referenzmodell erstellt werden kann. Nach einer Problembeschreibung und Definitionen, werden die Quellen betrachtet und für eine Integration analysiert. Nach der eigentlichen Umsetzung der Integration, gibt ein kurzer Ausblick die nächsten Arbeitsschritte vor und zeigt weitere Möglichkeiten auf.

1 Problem

Viele Ansätze der letzten Jahre zum Thema Servicetechnologien waren sehr Programm-bezogen und basierten auf fragmentiertem Wissen über Technologien und dem Modellieren von Architekturen. Diese Ausarbeitung beschäftigt sich mit der Frage, wie die besten Elemente der unterschiedlichen Vorgehensweisen miteinander integriert werden könnten und wie ein konkretes Integrationsmodell aussehen könnte. Als Basis dafür dienen die Referenzarchitekturen von OpenGroup und OASIS, sowie das OASIS Referenzmodell.

2 Anforderungen

2.1 OASIS

2.1.1 OASIS Referenzmodell

Das OASIS RM definiert das Vokabular von Elementen einer serviceorientierten Architektur (SOA) und deren Beziehungen. Das Modell besteht aus einem Glossar für Elemente und mehreren Beziehungs-Diagrammen. Es hat zum Ziel die Semantik der Terminologie zu standardisieren, geht aber über das Definieren nicht hinaus.

Durch Konzepte wie beispielweise der Service-Beschreibung, Sichtbarkeit und Interaktion von Services legt das Modell den Schwerpunkt mehr auf die IT als auf die Business-Architektur.

2.1.2 OASIS Referenzarchitektur

Die OASIS RA basiert auf den Konzepten und Beziehungen des hauseigenen Referenzmodells. Der Schwerpunkt dieser Architektur liegt darauf, Geschäftslogik mit der IT zu integrieren, die benötigt wird, um die Logik umzusetzen.

Die Architektur wird nach der Vorgehensweise aufgebaut, die in ANSI/IEEE 1471 Std. empfohlen wird. Das heißt sie wird in Modelle, Ansichten (Views), und Standpunkte (Viewpoints) unterteilt. Es existieren darin drei Hauptsichten: die „Business via Service“-Sicht als Grundlage für Geschäftslogik im Kontext der SOA; die „Realizing Services“-Sicht, die die Voraussetzungen für das Konstruieren einer SOA adressiert und die „Owning SOA“-Sicht, die sich auf Governance und Management fokussiert.

2.2 OpenGroup Referenzarchitektur

Die Referenzarchitektur von OpenGroup stellt eine SOA als Schichtarchitektur, bestehend aus neun Schichten (Layers), dar. Der Schwerpunkt bei OpenGroup liegt vollständig auf den technischen Aspekten – Geschäftslogik ist nicht Bestandteil dieser Architektur. Dafür wird auf IT-Aspekte sehr detailliert eingegangen, sodass der komplette Lebenszyklus eines Services inklusive Entwicklung, Verwendung, Bereitstellung und Governance abgedeckt wird.

3 Integrationsmodell

3.1 Analyse

Zunächst gilt es das Integrationspotential der Quellen anhand einer Korrelationsmatrix zu analysieren. Aufgrund der hohen Komplexität und des enormen Umfangs befasst sich diese Arbeit nur mit einem Ausschnitt und legt die Grundlage für ein Integrationsmodell.

Die OpenGroup Referenzarchitektur hat technische Aspekte als Schwerpunkt, während bei OASIS die Interaktion von IT und Geschäftslogik im Vordergrund steht. Aus diesem Grund ergeben sich sehr wenige Schnittpunkte, dafür ergänzen sich beide Architekturen sehr gut. Das Referenzmodell befindet sich in einer höheren Abstraktionsebene und dient als Basis.

In der Analyse wurden Schnittpunkte zwischen dem „Integration Layer“ von OpenGroup RA und dem „Visibility Model“ und „Interacting with Services Model“ von OASIS RA erkannt. Zusätzlich wurden der „Operational System Layer“ und das „Stakeholder and Participants Model“ als Repräsentanten der disjunkten Menge betrachtet

Bei OpenGroup ist der „Integration Layer“ das Kommunikationssystem der SOA, der dafür sorgt, dass alle Services von allen Layern erreicht werden können. Er steuert die Interaktion und Integration von Services, verarbeitet alle Nachrichten im System und beinhaltet Quality of Service, Sicherheitsprüfungen, das Management und Monitoring der gesamten Kommunikation. Bei OASIS ist das „Service Visibility Model“ dafür zuständig, dass sich Services untereinander sehen. Nach den Prinzipien „Awareness“, „Willingness“ und „Reachability“ wird sichergestellt, dass Service Provider und Konsument kommunizieren können. Ein Mediator kann dabei als Schnittstelle dienen. Ergänzend stellt das „Interacting with Services Model“ ein Prinzip zum Nachrichtenaustausch bereit. Bei der Gegenüberstellung fällt sofort auf, dass das Prinzip des Mediators in beiden Architekturen behandelt wird. Hier kann also eine Integration ansetzen. Der „Integration Layer“ wird um die OASIS-Modelle ergänzt, wodurch der Mediator noch mehr in den Mittelpunkt rückt und die Prinzipien der Sichtbarkeit (Visibility) ergänzt werden.

Im Gegensatz dazu hat der „Operational System Layer“ keine Entsprechung in OASIS. Er beinhaltet die Software und Hardware Infrastruktur, die für eine SOA notwendig ist und alle Komponenten der Entwurfs- und Laufzeit.

Umgekehrt findet sich das „Stakeholder and Participants Model“ von OASIS nicht in OpenGroup wider. Es beschäftigt sich mit der Beziehung zwischen den Stakeholdern, Teilnehmern und den Services einer SOA.

3.2 Umsetzung

Nun gilt es ein Integrationsmodell aus den Ergebnissen der Analyse zu entwickeln. Als Basis wird die Schichtenarchitektur von OpenGroup verwendet, da diese Darstellungsform sehr verbreitet ist und sich damit komplexe Zusammenhänge gut gliedern lassen. Außerdem sollen die technischen Aspekte vollständig abgebildet werden und um Geschäftslogik und Modelle der OASIS RA erweitert werden. Das OASIS Referenzmodell kann bildlich als Dach betrachtet werden, das auf höchster Abstraktionsebene die Semantik der Terminologie standardisiert. Zudem ist es die generische Grundlage, aus der sich die Referenzarchitekturen ableiten lassen. Die Modelle der Referenzarchitektur von OASIS, die nicht in das technische Schichtmodell einzuordnen sind, weil sie allgemeingültig formuliert sind und die Architektur in ihrer Gesamtheit betreffen, werden in eine Zwischenschicht „Generals“ eingeordnet. Abbildung 1 zeigt wie die Struktur des Integrationsmodells aussehen soll.

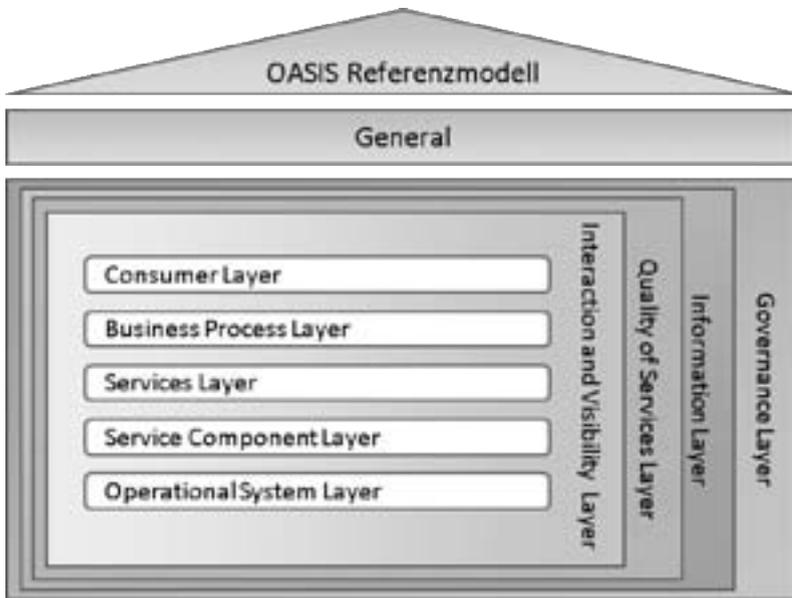


Abbildung 1: Struktur des Integrationsmodells

4 Ausblick

Wie zuvor erwähnt legt diese Arbeit nur den Grundstein für ein Integrationsmodell. Das Vorgehen ist bei den folgenden Schritten aber identisch: Integrationspotentiale werden mit der Korrelationsmatrix gesucht. An Schnittstellen muss entweder eine Entscheidung für eine Auslegung getroffen werden oder es wird eine Schnittmenge gebildet. Die disjunkte Menge wird weiter ausgearbeitet und in das Modell eingefügt. Natürlich muss auch immer die Frage nach der Relevanz gestellt werden, eventuell erweisen sich Teile als nicht notwendig.

Des Weiteren besteht die Möglichkeit zusätzliche Modelle oder Architekturen zu integrieren oder Bestandteile zu ersetzen. Das Integrationsmodell könnte beispielsweise zu einem Würfel mit Elementen des Enterprise Software Architecture Reference Cube (ESARC) erweitert werden. [AZ12]

Literaturverzeichnis

- [OA06] OASIS Standard: Reference Model for Service Oriented Architecture 1.0. 2006.
- [OA08] OASIS Standard: Reference Architecture for Service Oriented Architecture 1.0. 2006.
- [OG11] The OpenGroup: SOA Reference Architecture. 2011.
- [AZ12] A. Zimmermann, H. Buckow, H.-J. Groß, O.F. Nandico, G. Piller, and K. Prött: "Capability Diagnostics of Enterprise Service Architectures using a dedicated Software Architecture Reference Model", IEEE-SCC2011: Washington DC – July 5-10, 2011, pp. 592-599, 2011.

Effektivität von Multiagenten-Organisationstrukturen am Beispiel der Störgeräuscherkennung in Fahrzeugen

B.Sc. Christopher Jud

Universität Hohenheim
Wirtschaftsinformatik 2
cjud@uni-hohenheim.de

Art der Arbeit: Studienarbeit

Betreuer der Arbeit: Prof. Dr. Stefan Kirn

Abstract: Das Identifizieren von Störgeräuschen in Fahrzeugen im Rahmen von Wartung und Reparatur ist zeit- und kostenaufwändig. In dieser Arbeit wird daher die laufende Erkennung und Identifikation von Störgeräuschen durch ein on-board Multi-Agenten-System (MAS) behandelt. Es wird der Frage nachgegangen, welche Organisationsstruktur innerhalb eines MAS eine effektive Störgeräuscherkennung erlaubt. Die vorliegende research in progress-Arbeit definiert ein Forschungsprogramm, welches durch Prototyp und Experiment die Beantwortung der Forschungsfrage erlaubt.

1 Einführung

Es ist zu erwarten, dass die Intelligenz in Fahrzeugen immer weiter steigt. Schon heute sind durch Infotainment, Navigationssystem und dem mobilen Arbeitsplatz viele Computersysteme in einem Fahrzeug verbaut. Für die Zukunft verspricht die Entwicklung einer selbstständigen Fehlererkennung in Automobilen große wirtschaftliche Potenziale. In diesem Szenario identifiziert die im Fahrzeug verbaute Intelligenz über entsprechende Sensorik Störgeräusche im Fahrzeug. Das System überwacht den Fahrzeuginnenraum und analysiert selbstständig die auftretenden Geräusche im Hinblick auf die Identifikation von Störgeräuschemustern. Mehrere solcher Systeme können sich bei der Analyse selbstständig organisieren, um gemeinsam an der Identifikation zu arbeiten [Ki96].

In der Literatur wird die Organisation von Multi-Agenten-Systemen in verschiedenen Ausprägungen behandelt. [HL04] stellen in ihrer Arbeit verschiedene konkrete Organisationsformen für Multi-Agenten-Systeme vor. Die Arbeit stellt jedoch lediglich eine Aufzählung und Beschreibung prinzipieller Strukturen dar, sie enthält keine Aussagen über die Eignung der Strukturen für bestimmte Zwecke. [SS10] betrachten ein MAS zur Fehlererkennung und -identifikation in der chemischen Industrie, gehen dabei aber nur von einer Organisationsstruktur aus. Die Analyse des Einflusses unterschiedlicher Organisationsstrukturen ist nicht Gegenstand dieser Arbeit. [RTF06] beschreiben ein MAS für die verteilte Fehlerdiagnose und simulieren den Einsatz an einer Gasturbine.

Zum Einsatz kommt eine, von den Autoren entwickelte, Organisationsstruktur für die Zusammenführung verteilt erhobener Diagnosen. Einflüsse unterschiedlicher Organisationsstrukturen auf die Gesamtdiagnose werden nicht untersucht.

2 Forschungsansatz

Es wird ein Modell betrachtet, das eine Menge von einzelnen Agenten (Sensoren) über ein Fahrzeug verteilt (vgl. Abbildung 1). Im stilisierten Fahrzeug sind verschiedene Agenten (schwarz dargestellt) verbaut. In die Organisationsstrukturen sind jeweils nur Teilmengen der Agenten eingebunden, nicht eingebundene Agenten sind in grau dargestellt.

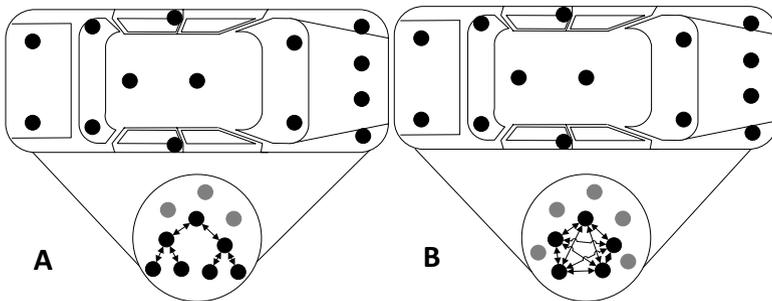


Abbildung 1 Agenten in Fahrzeugen: hierarchische und pluralistische Organisation der Agenten.

Die Agenten erfassen Geräusche, die von den Funktionsgeräuschen des Fahrzeugs abweichen. Erkennen sie die Geräusche als Störgeräusch, interagieren sie. Die Agenten beginnen, das Geräusch zu identifizieren. Zur Identifikation gehören u.a. die Ermittlung der Herkunft des Störgeräusches, die Kategorisierung des Störgeräusches und die Häufigkeit des Auftretens. Die Aufgabe der Identifikation wird damit im MAS verteilt und arbeitsteilig gelöst. Die Form der Arbeitsteilung wird im Rahmen einer Organisationsstruktur geregelt. Hierbei werden u.a. Rollen, Regeln zur Aufgabenerlegung, -zuteilung und -zusammenführung, Kommunikationswege, Hierarchien etc. festgeschrieben. In der Arbeit geht es um die Frage, welche Konfiguration der Agenten-Organisationsstruktur zur effektiven Erkennung geeignet ist.

Die betriebswirtschaftliche Organisationslehre bietet hierzu Erklärungsmodelle, die bei der Gestaltung von Organisationsstrukturen in MAS genutzt werden können. Als Beispiel seien hier die Diskrepanz- und Wirkungszusammenhangsmodelle von Hill, Fehlbaum & Ulrich genannt [HFU98]. Dabei muss berücksichtigt werden, dass die Organisationstheorie Konzepte betrachtet, die sich an menschliche Organisationen richten. Für die Anwendung auf technische Organisationen müssen die Konzepte auf die Adaptierbarkeit auf technische Systeme hin untersucht und angepasst werden. So können in Bezug auf Abbildung 1 bspw. folgende Fragen gestellt werden: Welche Auswirkung (auf die Effektivität der Erkennung von Störgeräuschen) hat das Hinzutreten oder Ausscheiden eines Agenten aus der Organisation? Welche Auswirkungen haben Veränderungen der organisatorischen Beziehungen? Welche Auswirkungen haben

veränderte Interaktionsstrukturen? Mit der Implementierung eines Prototyps, in dem die Strukturen der Organisation verändert werden können und der Durchführung unterschiedlicher Laborexperimente am physischen Modell Fahrzeug, lassen sich diese Fragen beantworten und Aussagen über probate Gestaltungsmittel treffen. Die mit diesem Ansatz erzielten Ergebnisse werden verglichen mit erfahrenen KFZ-Mechanikern. Das mittelfristige Ziel ist eine besser Identifikation von Störgeräuschen durch das MAS im Vergleich zu den KFZ-Mechanikern.

3 Forschungsprogramm

Zunächst werden aus der Organisationstheorie Modelle für die Strukturen von MAS-Organisationen abgeleitet (Artefakt-Typ: Modell). Anschließend wird auf dieser Basis ein MAS für die Identifikation von Störgeräuschen in Fahrzeugen prototypisch implementiert (Artefakt-Typ: Instanz). Durch den Prototyp sollen in Laborexperimenten effektive¹ Organisationsstrukturen für MAS evaluiert werden (Evaluationsmethode: Kontrolliertes Experiment). Der Prototyp wird anhand des GAIA-Modells [WJK00] realisiert.

3.1 Entwicklung und Realisierung eines Prototyps

Im ersten Schritt für die Umsetzung des Prototyps werden die Anforderungen ermittelt, die an das System gestellt werden. Die Phase umfasst explizit die Ableitung von Konstrukten aus der Organisationstheorie Modellierung von Organisationsstrukturen; so sollen unterschiedliche Rollen-, Interaktions und Acquaintance-Modelle erstellt werden. In der Analysephase wird das System mit Hilfe generischer Rollen-, sowie Interaktionsmodelle konzipiert. So sollen die Attribute Verantwortlichkeiten, Befugnisse, Aktivitäten und Protokolle zu den verschiedenen Rollen im System abgebildet werden. Die in der Analyse-Phase entworfenen Modelle werden in der Design Phase mit den Agenten-, Service- und Acquaintance-Modellen unterlegt. Das Modell der Services beinhaltet die notwendigen Services u.a. zur Erfassung von Störgeräuschen und das generische Acquaintance-Modell beinhaltet die Bekanntschaftsverhältnisse der Agenten. Die zu untersuchenden Organisationsformen werden mittels Parametrierung der Modelle instanziiert und bilden so die späteren Experiment-Settings.

3.2 Kontrolliertes Experiment

Im Experiment werden unterschiedliche Organisationsstrukturen in MAS (Explanans) im Hinblick auf ihre Effektivität bei der Erkennung von Störgeräuschen (Explanandum) untersucht. Die Evaluationsmetrik „Effektivität“ wird dabei in dreierlei Hinsicht erhoben: (1) als Verhältnis der erkannten Störgeräusche in Bezug zu den insgesamt im Experiment eingespielten Störgeräuschen (Recall), (2) als Verhältnis der richtig erkannten Störgeräusche in Bezug zu den vom MAS als Störgeräusch identifizierten

¹ „search for an effective artifact“, [He04]

Geräuschen (Precision) und (3) als Vergleich von MAS und Mensch (mehrere, voneinander unabhängige, Kfz-Mechaniker) in Bezug zu den beiden Größen Precision und Recall.

3.4 Erwartete Ergebnisse

Unterschiedliche Organisationsstrukturen sind unterschiedlich für die Erkennung von Störgeräuschen geeignet. Zu ermitteln und bewerten sind die Faktoren, von welchen die Eignung einer Organisationsstruktur abhängt. Als Beispiel hierfür sei der Vergleich einer pluralistischen und einer hierarchischen Organisationsstruktur genannt. Für die hierarchische Organisationsstruktur wird angenommen, dass die Aufgabenverteilung über „vorgesetzte“ Agenten abgebildet wird. In der pluralistischen Organisation müssen die Agenten sich erst auf eine Aufgabenteilung „einigen“, bevor der Problembearbeitungsprozess begonnen werden kann. Aus diesem Beispiel lassen sich beispielhaft zwei Faktoren für die Bewertung der Organisationsstruktur ableiten: der Aufwand für die Abstimmung in der Organisation und die Aufgabenteilung innerhalb der Organisation. Zu erwarten ist, dass Veränderungen in Organisationsstrukturen im MAS zu unterschiedlichen Ergebnissen bzgl. der Effektivität bei der Erkennung von Störgeräuschen führen. In Übereinstimmung mit den aus den Organisationstheorien entnommenen Erkenntnissen ist zu erwarten, dass eine effektive Organisationsform gefunden werden kann. Die Ergebnisse sollen in Form von Gestaltungsempfehlungen (Mittel-Zweck-Relationen) generalisierbar gemacht werden und so für andere MAS-Anwendungen nutzbar sein.

Literaturverzeichnis

- [He04] Alan R. Hevner et al. (2004): Design Science in Information Systems Research. In: MIS Quarterly, Vol. 28, No. 1, pp. 75-105.
- [HFU98] Hill, Wilhelm; Fehlbaum, Raymond; Ulrich, Peter (1998): Theoretische Ansätze und praktische Methoden der Organisation sozialer Systeme.
- [HL04] HORLING, BRYAN; LESSER, VICTOR (2004): A survey of multi-agent organizational paradigms. In: Knowl. Eng. Rev. 19 (04).
- [Ki96] Kirn, St. (1996): Gestaltung von Multiagenten-Systemen: Ein organisationszentrierter Ansatz. Westfälische Wilhelms-Universität Münster
- [RTF06] X. Ren, H.A. Thompson und P.J. Fleming (2006): An agent-based system for distributed fault diagnosis. In: Int. Journ. of Knowledge-based and Intelligent Engineering Systems 10 Issue 5, 319–335.
- [SS10] Seng Ng, Yew; Srinivasan, Rajagopalan (2010): Multi-agent based collaborative fault detection and identification in chemical processes. In: Engineering Applications of AI 23 (6), S. 934–949.
- [WJK00] Wooldridge, Michael; Jennings, Nicholas R.; Kinny, David (2000): The Gaia Methodology for Agent-Oriented Analysis and Design. In AAMAS, September 2000, Volume 3, Issue 3, pp 285-312.

Wissensbasiertes automatisiertes IT-Management

Doktorand	Erstbetreuer	Zweitbetreuer
Fabian Meyer	Prof. Dr. Ernst Althaus	Prof. Dr. Reinhold Kröger
Hochschule RheinMain Labor für Verteilte Systeme 65195 Wiesbaden fabian.meyer@hs-rm.de	Johannes Gutenberg-Universität Institut für Informatik 55099 Mainz ernst.althaus@uni-mainz.de	Hochschule RheinMain Labor für Verteilte Systeme 65195 Wiesbaden reinhold.kroeger@hs-rm.de

1 Motivation

IT-Anwendungen und die darunterliegende Infrastruktur sind in den letzten Jahren für Unternehmen in nahezu allen Branchen existentiell geworden und leisten einen signifikanten Beitrag bei der Erfüllung von Unternehmenszielen. In der Regel wachsen die dabei eingesetzten Systeme über Jahre heran und bestehen aus einer Menge heterogener Komponenten. Dennoch wird von den offerierten Diensten meist ein Höchstmaß an Zuverlässigkeit abverlangt, wodurch das IT Management zu einem immer wichtigeren Beitrag für den Betrieb des gesamten Unternehmens geworden ist. Betrachtet man jedoch die ständig wachsende Komplexität solcher Systeme, ist deren Management zu einer fast unlösbaren Aufgabe geworden, sodass der Bedarf an intelligenten, automatisierten Management-Werkzeugen stark gewachsen ist. Zwar existieren solche Werkzeuge bereits, in der Regel sind sie jedoch spezifisch für ein Produkt oder eine Komponente und ermöglichen nicht das ganzheitliche Management von großen, heterogenen Systemen. Ein Grund dafür ist die Diversität der unterschiedlichen Management-Modelle und -Schnittstellen, die eine Integration in bestehende Frameworks extrem aufwändig macht.

Im Zuge der Entwicklungen im Bereich von Semantic Web Technologien haben in den letzten Jahren Ontologien aus dem Gebiet der künstlichen Intelligenz einen neuen Aufschwung als Modelle zur domänenübergreifende Wissensrepräsentation erfahren und viele neue Standards wie die Web Ontology Language (OWL), die Semantic Web Rule Language (SWRL) und die SPARQL Protocol and RDF Query Language (SPARQL) sind entstanden. Dabei sind Ontologien nicht nur eine syntaktische Beschreibung von Wissen, sie definieren auch eine auf der Beschreibungslogik basierende Semantik, sodass damit entworfene Modelle von Reasonern validiert und neue Fakten aus der bestehenden Wissensbasis abgeleitet werden können. Diese Charakteristiken machen Ontologien zu einem sehr mächtigen Hilfsmittel bei der Kombination von unterschiedlichsten Domänenmodellen und somit bei einem Einsatz für das übergreifende, wissensbasierte Management von heterogenen IT-Systemen. Es gibt bereits einige Arbeiten, die sich mit dem Einsatz von Ontologien für das IT-Management beschäftigt haben (siehe [VVB04] [DGVB09] [VV06] [TMK11]). Dabei zeigte sich, dass Ontologien sehr gut zum Management von kleinen Systemen ohne Betrachtung von Zeit geeignet sind, echte Systemmodelle jedoch sehr viel komplexer und dynamischer sind, wodurch sich beim Einsatz von Ontologien die folgen-

den Probleme ergeben: (1) Die Reasoning-Komplexität von OWL2 Ontologien ist NP-vollständig [MGH⁺12], was sie für das Management von großen, dynamischen Systemen schwer handhabbar macht. (2) In Ontologien existiert kein Zeitkonzept, sie reflektieren nur den Zustand für einen konkreten Zeitpunkt, sodass eine interne Versionierung für zeitbasiertes Reasoning benötigt wird [WFM⁺, BDM], was wiederum die Ontologie aufbläht und die Performance negativ beeinflusst.

Daher ist es das Ziel dieser Arbeit, ein Management-Framework zu entwickeln, das Ontologien mit Verfahren aus den Bereichen des Complex Event Processings (CEP) und der Optimierung verknüpft, um die Stärken der semantischen Beschreibung von domänenübergreifenden Modellen als Wissensbasis mit den Stärken der Echtzeitdatenstromanalyse von zeitbehafteten Informationen und mit Optimierungsalgorithmen zu kombinieren. Die gesamte Konfiguration für die Laufzeitanalyse soll dabei aus dem Modell abgeleitet werden, sodass ein ganzheitliches Management von heterogenen IT-Systemen auf Basis einer homogenen Modellbeschreibung erreicht wird.

2 Methodik

In der Arbeit wird ein Framework zum automatisierten Management von IT-Systemen entwickelt. Dazu wird als Grundlage die von IBM in [IBM06] vorgestellte Monitor Analyze Plan Execute Knowledge (MAPE-K) Loop eingesetzt, die den Selbst-Management-Zyklus auf fünf zentrale Komponenten herunterbricht. Im Folgenden wird die Realisierung der einzelnen Komponenten im vorgestellten Ansatz beschrieben, die in Abbildung 1 dargestellt sind.

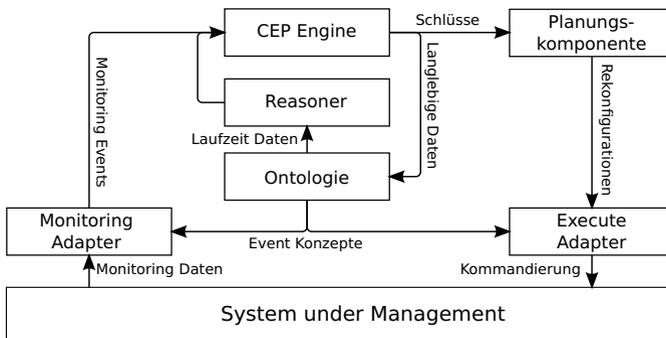


Abbildung 1: Architektur der zentralen Komponenten des Selbst-Management-Zyklus.

Als zentrale Komponente fungiert die Knowledge-Komponente, in der Wissen über das System unter Management (SUM) hinterlegt ist. Diese Komponente wird durch eine Menge von Ontologien unterschiedlicher Domänen realisiert, die alle Systeminformationen widerspiegeln. Dazu gehört ein systemspezifisches Modell der Systemtopologie, ein Modell der im System auftretenden Events, eine Abbildung der Events auf Topologieelemente und ein Service-Level-Management-Modell, auf dessen Basis Service Level Agreements über Service Level Indikatoren des Topologiemodells definiert werden können.

Die Analyse-Komponente nutzt die Monitoring-Daten des überwachten Systems in Kombination mit den Informationen aus der Wissensbasis, um neue Schlüsse über den Zustand des Systems abzuleiten. Dazu wird in dieser Arbeit ein neuartiges, kombiniertes Auswertungsmodell verfolgt, das Ontologie-Reasoning mit Verfahren aus dem Bereich des Complex Event Processings verknüpft. Dabei werden aus in der Ontologie modellierten Aggregationsvorschriften Regeln für eine Event Processing Engine generiert. Als Eingabe für die Auswertung der Regeln werden zur Laufzeit sowohl die Monitoring-Daten als auch das von einem Reasoner aus der Wissensbasis geschlossene Wissen genutzt. Verwandte Ansätze zur Einbeziehung von semantischem Wissen in CEP wurden bereits in [SVL03] und [TMT⁺12] beschrieben, dabei ist die Beschreibung der Events und deren Aggregationsvorschriften jedoch nicht Teil des Modells.

In der Planungskomponente wird das Ontologiemodell, welches um das abgeleitete Wissen aus der Analysephase erweitert ist, dazu genutzt, um Rekonfigurationen für das System zu planen. Basierend auf einer Menge von im Modell definierten Einzelrekonfigurationen wird mit dem Einsatz von Optimierungsalgorithmen wie Tabu Search [GTT93] eine Rekonfigurationskette bestimmt, die das System anhand der definierten SLAs optimiert.

Die Monitor- und Execute-Komponenten sind einfache Technologieadapter, die als Schnittstelle zwischen dem SUM und dem hier vorgestellten Management-Framework dienen.

3 Anwendungsfall

Die Arbeit ist im Kontext einer Forschungskooperation zwischen der Hochschule Rhein-Main und der Deutschen Flugsicherung GmbH angesiedelt. Die Flugsicherung entwickelt das Air Traffic Control (ATC)-System PHOENIX [Hei09], dessen Aufgabe die Verarbeitung von aus externen Datenquellen bezogenen Informationen wie Radarmessungen, Wetterdaten, Flugplänen, etc., über deren Korrelation und Situation Assessment bis hin zur Visualisierung der Lagedarstellung für die Lotsen ist. Der dabei unterlagerte Verarbeitungsprozess wird durch die Interaktion einer Vielzahl lose gekoppelter Prozesse realisiert. Da im Flugsicherungskontext die Sicherheit höchste Priorität hat, muss das fehlertolerant angelegte Gesamtsystem bestehend aus den externen Schnittstellen, der eingesetzten Hardware und den Prozessen durchgängig von einer Gruppe von Administratoren überwacht werden, was sehr zeit- und kostenaufwendig ist. Daher soll durch die Integration der entwickelten Methoden in Form eines Assistenzsystems, das bereits frühzeitig eigenständig Gefahrensituationen erkennt, analysiert und Rekonfigurationsvorschläge erzeugt, eine Entlastung der Administratoren erzielt werden. Insbesondere wird dabei die Überwachung der Radaranlagen und des daraus resultierenden Trackings (Korrelation von Radarmessungen zu Bewegungsmustern) betrachtet.

4 Stand

Die für die Methoden eingesetzten Ontologien wurden bereits in ersten Versionen umgesetzt und durch deren Einsatz in einer prototypischen Implementierung unter Nutzung von

synthetischen Eingabedaten und Management-Regeln ein Proof of Concept für die kombinierte Analyse aus Ontologien und CEP gegeben, die Optimierung wurde dabei noch nicht betrachtet. In den verbleibenden zwei Jahren sollen die Umsetzung der Komponenten des generischen Management-Frameworks inklusive der Integration der beschriebenen Verfahren, die Integration der Optimierungsverfahren, die Finalisierung der Ontologien und die Anbindung des echten Flugsicherungssystems durchgeführt werden.

Literatur

- [BDM] Jie Bao, Li Ding und Deborah L. McGuinness. Semantic History: Towards Modeling and Publishing Changes of Online Semantic Data.
- [DGVB09] Jorge E. López De Vergara, Antonio Guerrero, Víctor A. Villagrà und Julio Berrocal. Ontology-Based Network Management: Study Cases and Lessons Learned. *Journal of Network and Systems Management*, 17(3):234–254, September 2009.
- [GTT93] F Glover, E Taillard und E Taillard. A user’s guide to tabu search. *Annals of operations research*, 1993.
- [Hei09] Ralf Heidger. The PHOENIX White Paper V2.0. publicly available on request, 2009.
- [IBM06] IBM Corporation. An Architectural Blueprint for Autonomic Computing, Technical Whitepaper (Fourth Edition), June 2006.
- [MGH⁺12] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu Achille Fokoue und Carsten Lutz. OWL 2 Web Ontology Language Profiles (Second Edition), 2012.
- [SVL03] Sachin Singh, Pravin Vajirkar und Yugyung Lee. Context-Based Data Mining Using Ontologies. In Il-Yeol Song, StephenW. Liddle, Tok-Wang Ling und Peter Scheuermann, Hrsg., *Conceptual Modeling - ER 2003*, Jgg. 2813 of *Lecture Notes in Computer Science*, Seiten 405–418. Springer Berlin Heidelberg, 2003.
- [TMK11] Andreas Textor, Fabian Meyer und Reinhold Kroege. Semantic Processing in IT Management. In Pascal Lorenz und Eckhard Ammann, Hrsg., *Proceedings of the Fifth International Conference on Advances in Semantic Processing (SEMAPRO)*, Lisbon, Portugal, November 2011.
- [TMT⁺12] Andreas Textor, Fabian Meyer, Marcus Thoss, Jan Schaefer, Reinhold Kroege und Michael Frey. An Architecture for Semantically Enriched Data Stream Mining. In Sandjai Bhulai, Joseph Zernik und Petre Dini, Hrsg., *Proceedings of the First International Conference on Data Analytics*, Barcelona, Spain, September 2012.
- [VV06] Víctor A. Villagrà und Jorge E. López De Vergara. Ontologybased Policy Refinement Using SWRL Rules for Management Information Definitions. In *in OWL. In: Proc. 17th IFIP/IEEE International Workshop on Distributed Systems, Operations and Management (DSOM)*, Seiten 227–232, 2006.
- [VVB04] Jorge E. López De Vergara, Víctor A. Villagrà und Julio Berrocal. Applying the Web ontology language to management information definitions. *IEEE Communications Magazine*, 42:68–74, 2004.
- [WFM⁺] Christopher Welty, Richard Fikes, Selene Makarios, Christopher Welty, Richard Fikes und Selene Makarios. 2006. A reusable ontology for fluents in OWL. In *In Proceedings of FOIS*, Seiten 226–236.

Informationslogistisches Rahmenwerk für bedarfsgerechte Informationsversorgung in Unternehmen

Dirk Stamer

Universität Rostock

Fakultät für Informatik und Elektrotechnik
Lehrstuhl für Wirtschaftsinformatik
dirk.stamer@uni-rostock.de

Art der Arbeit: Dissertation

Betreuer der Arbeit: Prof. Dr. Kurt Sandkuhl

Abstract: Bedarfsgerechte und zeitnahe Informationsversorgung kann in Unternehmen als Erfolgsfaktor angesehen werden. Trotz des Angebots von diversen etablierten Rahmenwerken wie ITIL oder COBIT wird in Unternehmen viel Zeit für das Suchen der richtigen Information aufgewendet. Gerade bei unstrukturierten oder plötzlich auftretenden Entscheidungsproblemen mangelt es an der richtigen Information trotz einer Implementierung eines bekannten Rahmenwerks im Unternehmen. Daher wird in dieser Arbeit ein Rahmenwerk basierend auf der Informationslogistik angeboten, welches das bestehende Konzept der Informationsbedarfsmuster anwendet, um den Informationsfluss in Arbeitsgruppen von Organisationen zu verbessern. Die Arbeit folgt der Forschungsfrage, wie organisatorisches Wissen um den Informationsbedarf von Gruppen nutzbringend wiederverwendbar gemacht werden kann.

1 Motivation

Die rechtzeitige Verfügbarkeit von Informationen in der notwendigen Qualität ist gerade für Entscheidungsaufgaben, wissensintensive Aufgaben und bei der Problemlösung von entscheidender Bedeutung. Dem entgegen steht die Entwicklung in Unternehmen der Industrie, dass die Mitarbeiter einer Überfrachtung mit Informationen ausgesetzt sind. Es kann angenommen werden, dass eine Verbesserung des Informationsflusses Auswirkungen auf die Effizienz der Organisation und die Güte der Arbeitsergebnisse hat [1].

Die Informationslogistik als Forschungsgebiet beschäftigt sich mit der Verbesserung der Informationsbereitstellung und des Informationsflusses insbesondere bei Entscheidungen unter Unsicherheit sowie spontan auftretenden Aufgaben. Eine bedarfsgerechte Informationsversorgung ist in diesem Zusammenhang definiert als die richtige Information, zur richtigen Zeit, in der richtigen Qualität, in der richtigen Form und am richtigen Ort für die suchende Person [2].

2 Informationslogistikgrundlagen

Vorangegangene Arbeiten haben gezeigt, dass der Informationsbedarf einer Person in einem Unternehmen in Abhängigkeit zu ihrer Rolle steht [3]. Ausgehend von dieser Prämisse wurden in anderen Arbeiten Informationsbedarfsmuster entwickelt und das iterative Vorgehen zu Entwicklung solcher Muster validiert [4]. Informationsbedarfsmuster haben das Ziel den Informationsbedarf der Rolle zu kapseln und in ein Muster zu überführen. Das entstandene Muster kann dann auf ähnliche Rollen in unterschiedlichen Domänen oder auf identische Rollen in anderen Unternehmen angewendet werden und führt somit zu einer Wiederverwendung des erfassten Informationsbedarfs. Ein Informationsbedarfsmuster greift also ein wiederkehrendes Informationsflussproblem auf, das sich für eine bestimmte Rolle und eine Arbeitssituation in einem Unternehmen ergibt und präsentiert dafür eine konzeptuelle Lösung [5].

Ein Informationsbedarfsmuster setzt sich aus den folgenden Bestandteilen zusammen: *Name des Musters*, *Organisationskontext*, *Probleme*, *konzeptuelle Lösung* und *Effekte*. Der *Name des Musters* beschreibt üblicherweise die Bezeichnung der Rolle, die das Muster betrifft. Der *Organisationskontext* stellt den Anwendungsfall des Musters dar, z.B. Funktion, Abteilung oder Domäne. *Probleme* beschreibt die Schwierigkeiten, die die jeweilige Rolle hat und umfasst ebenfalls ihre Aufgaben und Verantwortlichkeiten. Wie das beschriebene Problem gelöst werden soll, thematisiert die *konzeptuelle Lösung*. Dimensionen wie Informationsbedarf, Qualitätskriterien und Zeitachse werden hier berücksichtigt. Die *Effekte* beschreiben die negativen Auswirkungen bei fehlenden oder unvollständigen Informationen [4].

3 Forschungsvorgehen

Die der Arbeit zugrundeliegende Forschungsmethodik ist die Methode „Design-Science“. Dabei werden die folgenden von Hevner et.al. [6] vorgeschlagenen Richtlinien des Rahmenwerks durchlaufen: *Artefaktentwurf*, *Problemrelevanz*, *Entwurfevaluation*, *Forschungsbeitrag* und *Forschungsgenauigkeit*. Aktuell wird ein Informationsbedarfsmuster für eine Gruppe entwickelt, also die Übertragung des bekannten Konzepts der Informationsbedarfsmuster auf Arbeitsgruppen untersucht. Dies geschieht im Kontext der Universität anhand des Lenkungsgremiums eines internen Projekts. Das entstehende Informationsbedarfsmuster für diese Gruppe ist das initiale Artefakt analog zur eingesetzten Forschungsmethode. Es wird untersucht, ob der Informationsbedarf einer Arbeitsgruppe mehr ist als die Summe der Informationsbedarfe der einzelnen Rollen. Weiterhin wird aus dem Entwicklungsprozess des Musters ein generelles Vorgehen zur Entwicklung von Mustern für Gruppen abgeleitet.

Dieses isolierte generelle Vorgehen zur Erstellung von Informationsbedarfsmustern für Gruppen kann als weiteres Artefakt im Sinne der Methode „Design Science“ betrachtet werden.

Aus der bereits beschriebenen Forschungsfrage - wie kann organisatorisches Wissen um den Informationsbedarf von Gruppen nutzbringend wiederverwendbar gemacht werden - ergibt sich unter anderem die Notwendigkeit den Mehrwert des entwickelten Gruppenmusters für Unternehmen in diesem Kontext zu evaluieren. Hierfür soll ein Messansatz entwickelt werden, um den erwarteten Mehrwert des vorgeschlagenen Rahmenwerks validieren zu können.

In Unternehmen sind in der Regel eine Vielzahl von Informationen in diversen Informationssystemen vorhanden. Es wird weiterhin vorgeschlagen eine Verknüpfung zwischen Informationsbedarfsmustern und Informationssystemen dergestalt herzustellen, dass zu einem gegebenen Informationsbedarfsmuster die benötigten Informationen teil- bzw. vollautomatisiert aus den vorhandenen Informationssystemen beschafft werden. Aktuell wird ein Prototyp entwickelt, der auf Basis von Ontologien eine Verbindung zwischen Informationsbedarfsmustern und Informationssystemen herstellt. Anhand des Prototyps soll die Entwicklung von der Informationsbedarfsanalyse über die Nutzung von Informationsbedarfsmustern hin zu einer in Unternehmen anwendbaren Lösung gezeigt werden.

4 Zusammenfassung und Ausblick

Im Vorfeld der Arbeit wurde bereits die Anwendbarkeit von Informationsbedarfsmustern validiert sowie Verbesserungsvorschläge erarbeitet [7]. Die Möglichkeit der Übertragung von Informationsbedarfsmustern von einzelnen Rollen auf Gruppen wird aktuell untersucht. Ebenso wird die Frage untersucht, ob ein Informationsbedarfsmuster für Gruppen aus mehr besteht als aus der Summe der Informationsbedarfe der beteiligten Einzelmuster. Das generelle Vorgehen zur Gewinnung von Gruppenmustern wird extrahiert und validiert. Aktuell wird ebenfalls die Weiterentwicklung von Informationsbedarfsmustern zu einer in Unternehmen anwendbaren informationslogistischen Lösung geprüft.

Das Ziel der Arbeit ist die Entwicklung eines Rahmenwerkes, das mit dem Einsatz von Informationsbedarfsmustern für Arbeitsgruppen einen Beitrag zur Steigerung der Effizienz in Unternehmen liefern soll.

Literaturverzeichnis

1. Öhgren, A., Sandkuhl, K.: Information Overload in Industrial Enterprises - Results of an Empirical Investigation. Presented at the Proceedings ECIME 2008, London (2008).
2. Sandkuhl, K.: Information Logistics in Networked Organizations: Selected Concepts and Applications. Springer (2008).
3. Lundqvist, M.: Information Demand and Use: Improving Information Flow within Small-scale Business Contexts (2007).
4. Sandkuhl, K.: Information Demand Patterns. Presented at the PATTERNS 2011, The Third International Conferences on Pervasive Patterns and Applications (2011).
5. Sandkuhl, K.: Validation and Use of Information Demand Patterns in Higher Education. Presented at the Business Information Processing, Berlin, Heidelberg (2010).
6. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS quarterly. (2004).
7. Sandkuhl, K., Stamer, D.: Towards Improving Structure and Content of Information Demand Patterns. Presented at the Perspectives in Business Informatics Research September 24 (2012).

Campus2Go

Content Enrichment Using Enterprise Integration Patterns - Master's Project -

Eberhard Heber, Daniel Stiefel

Wirtschaftsinformatik Master
Hochschule Reutlingen
Alteburgstraße 150
72764 Reutlingen
surname.lastname@student.reutlingen-university.de

Abstract: In their daily academic life, students, professors, and staff need to access multiple systems to get and exchange information. Since these systems are usually not integrated, the situation is rather unsatisfying. The goal of this project is to build a mobile application including the middleware -Campus2Go (C2Go)- that allows accessing these systems' data in an integrated manner. The major challenge that needs to be tackled is the integration of the underlying systems. In particular to link their information such that the result is a network of enriched information. This work presents how three Enterprise Integration Patterns (EIP) using the Apache Camel platform enabling the enrichment of content.

Problem

In C2Go data is classified as master or content, whereas master data itself can be content, but not vice versa. Master data represent the structure of a campus. The underlying idea is to use master data to enrich content data as well as to navigate through the content. The problem with master data is that it needs to be put in relation with other master data. Like content master data needs to be extracted from existing systems first before it can be used to enrich content and to navigate through the content.

For example, CMSs (Content Management System, see [WG12]) provide their news via RSS feeds (Really Simple Syndication, see [RSS]). A feed entry is published by a certain person, who belongs to an organization (e.g., faculty), has a room in a building, an e-mail address, a phone number, consultation-hours, and so forth. The feed entry does not provide all this information. Therefore, other systems, for example, the university's LDAP (Lightweight Directory Access Protocol) system has to be integrated. It provides the person's e-mail, phone number, and other information. To link the person's details with the feed entry is called **enrichment**. Information about the person, his organization, room, building, etc. is master data. The news feed entry, however, is content. Once the feed entry is enriched, it is possible to navigate to the person's details such as his organization, and his room. If available, it is possible to view the room's bookings. The result is a network of information and no longer just a feed entry.

Solution

As an initial step to allow the enrichment, an appropriate class structure was derived from the results of a survey about the functionalities students would like to have in C2Go. With this information in hand, it was possible to easier identify that parts of the campus which should be master data. Figure 1 illustrates this class structure (note that the diagram just shows class names and relationships).

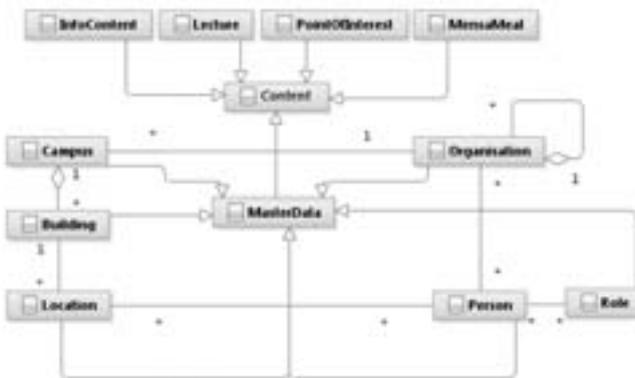


Figure 1: Simplified UML class diagram.

To actually enrich content three strategies have been identified:

- 1) *By-id*: For some new content a static and manual classification is possible. For example, a feed explicitly names a person and it is certain that this person is known by the system. I.e., the association with a person will finally exist. In this case the CMS uses the LDAP system already.
- 2) *By-information*: Other new content provides information in a different format only. For example, in the timetable application a lecturer is just represented by an abbreviation of the name and an additional step is needed to transform the timetable's representation of a person into a representation that allows creating the association. In this case the timetable system does not use the LDAP system.
- 3) *By-best-match*: Yet other content might even need text mining [HQW] approaches, because it is not sure if the new content provides any information at all, which can be exploited to conduct the enrichment and create associations.

To model and implement the enrichment according to these three strategies C2Go uses Enterprise Integration Patterns (EIP). As a platform to implement the EIP, Apache Camel was chosen. There was no product evaluation and the decision to use this platform was because Camel is based on known EIPs.

Figure 2 now illustrates the solution. Via an adapter data is imported into C2Go, for example, the adapter queries an RSS feed. Each adapter is instrumented by a requester and creates a Camel specific exchange-object (*SyncFeed* in case of RSS) that contains some content. The exchange object is transformed into an unenriched C2Go specific content-object, such as a lecture. Unenriched means the associations to the master objects are missing. Since different content might require a different degree of enrichment. The succeeding router, which is based on the **Recipient List Pattern**, delivers the content-object only to those processors actually needed for enrichment. For example, content that needs to be enriched only by a person is therefore routed only to the *PersEnrichment* processor.

According to the Camel documentation a “Recipient List from the EIP patterns allows you to route messages to a number of dynamically specified recipients” [AC, IA11].

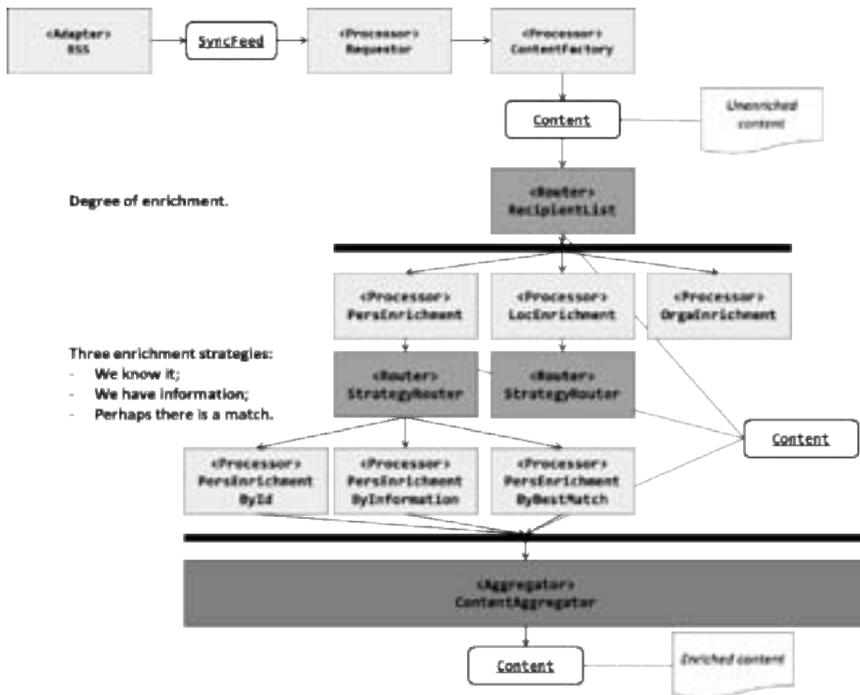


Figure 2: Solution using EIPs

As already said, there are three possible enrichment strategies: “*by-id*”, “*by-information*” or “*by-best-match*”. Therefore, a *StrategyRouter* is used for each master class (Person, Location, Organisation, etc.). The *StrategyRouter* is a “**Content Based Router**”, which “allows to route messages to the correct destination based on the contents of the message exchanges” [AC, IA11].

The destination of a *StrategyRouter* is exactly one processor that finally enriches the given content-object. The concrete implementation is use-case specific, and for example,

content from the timetable system requires a different implementation of the enrichment than an RSS entry does. In case of the *by-id* strategy, the implementation takes usually the id of the master object. In case of the *by-information* strategy, a key-value list where keys are possible attributes is passed to the implementation. In case of the *by-best-match* strategy, text mining or other information retrieval approaches are needed. Note that concrete implementations are beyond the scope of this paper.

The **Aggregator** pattern is finally used “to combine a number of messages together into a single message” [AC, IA11]. The aggregator finally enriches content. For each content class a different aggregation is needed.

Conclusion

In fact, the C2Go team is not able to precisely judge the value of this solution using EIP. C2Go is still in its initial phase. However, the loosely coupled architecture based on EIP as provided by Apache Camel can scale with future requirements and significantly increases the re-usability as experienced already. Moreover, data is provided to mobile clients also using Camel. So far, most of the problems were due to the integration of Hibernate, Camel, and Tomcat using Spring as glue. Most notably, reliable transaction and session handling across these different systems is cumbersome. Of course, Apache Camel is just one possible framework to implement EIP, but C2Go’s intention was not to evaluate products or solutions.

Bibliography

- [AC] Apache Camel Product Website. Jan. 2013.
<https://camel.apache.org/enterprise-integration-patterns.html>
- [EIP03] Hohpe, G.; Woolf B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 2003.
- [HQP08] Heyer, G.; Quasthoff, U.; Wittig, T Text Mining: Wissensrohstoff Text: Konzepte, Algorithmen, Ergebnisse. Herdecke. ISBN 978-3-937137-30-8. 2008.
- [IA11] Ibsen, C.; Anstey, J.: Camel in Action. Manning Publications Co., Stamford, USA. 2011.
- [RSS] RSS 2.0 Specification,
<http://www.rssboard.org/rss-specification>, Jan. 2013
- [WG12] Witt, A.; Göbe, T.: webEdition: CMS, eCommerce, Online-Marketing. München, Open Source Press. ISBN 978-3-941841-79-6. 2012.

Einführung in Business Process Management und die Entwicklung von Business Process Model & Notation Patterns

Mehtap Yilmaz | Agim Imeraj | Benjamin Hasimov | Dino Delic |
Markus Riegger | Usama Mir

University of Applied Science Reutlingen
Master - Wirtschaftsinformatik
Alteburgstraße 150
72762 Reutlingen

Mehtap.Yilmaz@student.reutlingen-university.de
Agim.Imeraj@student.reutlingen-university.de
Benjamin.Hasimov@student.reutlingen-university.de
Dino.Delic@student.reutlingen-university.de
Markus.Riegger@student.reutlingen-university.de
Usama.Mir@student.reutlingen-university.de

Abstract: In der Vergangenheit versuchten die Unternehmen Ihre gesamten Geschäftsprozesse mit einem System abzudecken und alle anfallenden Daten und Informationen in einer einzelnen Datenbank zu speichern. Dies führt mit sich, dass die einzelnen Prozessschritte applikationsabhängig sind und eine Prozessänderung dadurch enorm erschwert wurde. Mit Business Process Management (BPM) kann diese Schwerfälligkeit verbessert und eine zeitnahe Anpassung an die Marktdynamik ermöglicht werden.

1 Einführung

BPM besteht im Wesentlichen aus den Punkten Erfassung, Analyse, Optimierung, Kontrolle und Dokumentation von Geschäftsprozessen eines Unternehmens. Diese Managementdisziplin lässt sich in weitere Teildisziplinen unterteilen, wie beispielsweise Business Process Model & Notation (BPMN). Die bisherigen verbreiteten Notationen (EPK und UML) haben entweder die Sicht aus dem Fachbereich oder der IT dargestellt. Welche Notation beschreibt die Modelle, die für alle Beteiligten zu verstehen sind – ohne dass dabei Informationsverluste in Kauf genommen werden? BPMN eignet sich

sowohl für den Fachbereich als auch für die IT-Abteilung. Die Geschäftsprozesse können einfach und verständlich erstellt werden, jedoch besteht die Möglichkeit die GPs in voller Komplexität zu modellieren. Die Möglichkeit die Prozesse mit einer gemeinsamen Notation zu Modellieren ist ein entscheidender Vorteil für jedes Unternehmen. Dadurch kann ein unternehmensweites und einheitliches Verständnis der Prozesse erreicht werden.

Das Ziel dieser Arbeit ist es, Möglichkeiten aufzuzeigen wie BPMN verstanden und eingesetzt werden.

2 BPMN Patterns

BPMN besteht aus diversen Elementen und Funktionen. Aus diesen werden passende Patterns abgeleitet. Einen Überblick zu den Elementen und Funktionen, sowie die entwickelte Kategorie für die entworfenen Patterns kann der *Abbildung 1* entnommen werden.

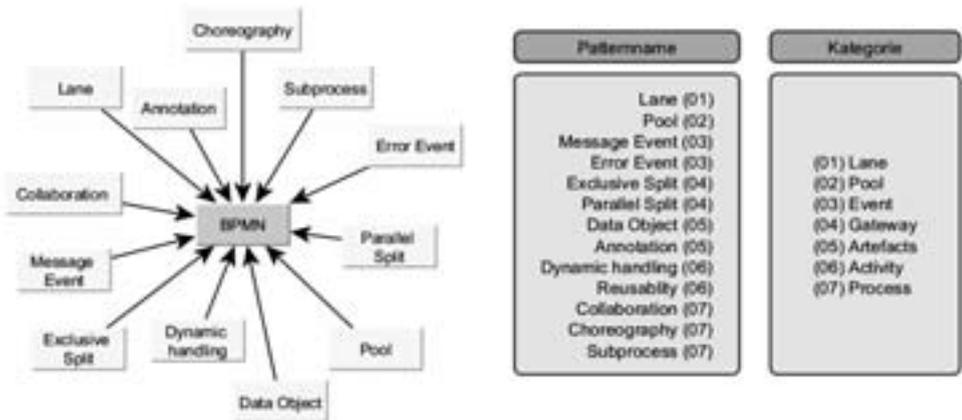


Abbildung 1 - Übersicht der BPMN Elemente, Patterns und Kategorien

Die nachfolgenden Unterkapitel zeigen einen repräsentativen Ausschnitt der entworfenen Patterns.

2.1 Lane Pattern

In einem Unternehmen erstrecken sich immer mehr Prozesse übergreifend über unterschiedliche Abteilungen und Rollen. Oft sind unklare Verantwortung und Zuständigkeit eine Folge davon. Wie kann mit dieser Herausforderung umgegangen werden, um eine klare Zuständigkeit zu definieren und um Verantwortung innerhalb eines Unternehmens klar zu trennen? Für eine klare Abgrenzung der Verantwortung innerhalb eines Unternehmens kommt in BPMN die Funktion der Lanes zum Einsatz. Mithilfe von Lanes können die Bestandteile wie Tasks und Subprozesse in einem

Prozessdiagramm auf die jeweiligen Abteilungen in einem Unternehmen verteilt werden. Dies wird in der Abbildung 2 nochmal verdeutlicht. Im Prozessmodell wird das Unternehmen anhand von drei Lanes dargestellt. Diese stellen die Vertriebs-, Lager-, und IT-Abteilung dar. Innerhalb des

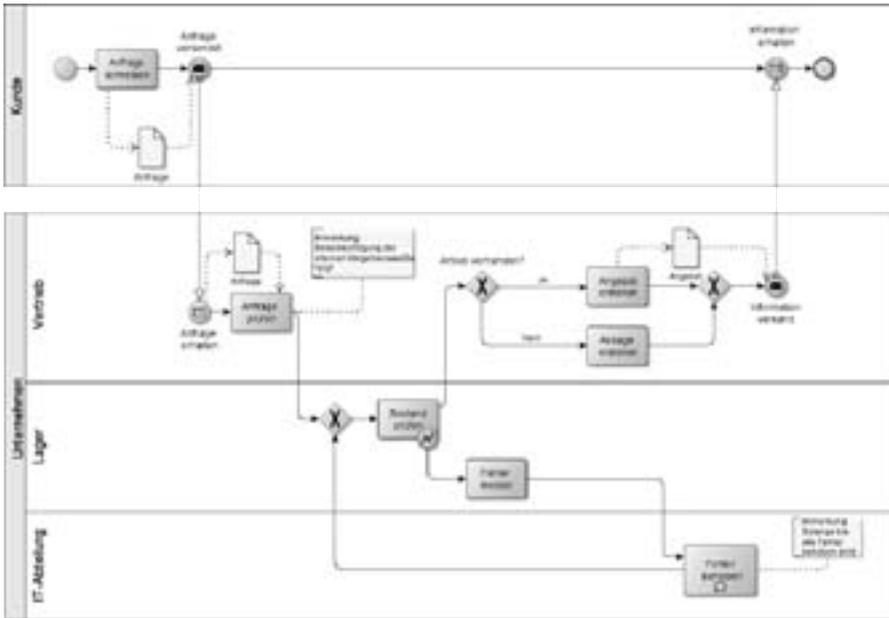


Abbildung 2: Prozessbeispiel mit BPMN

Prozesses ist die Lagerabteilung verantwortlich für die Bestandsüberprüfung und bei Fehler in den Beständen diese zu melden. Gleichmaßen kommt die Funktion der Lanes für die anderen Abteilungen zum Einsatz, indem klare Verantwortung und Zuständigkeit innerhalb der Lanes festgelegt werden.

2.2 Data Object Pattern

Bei einem Geschäftsprozess werden diverse Dokumente erstellt und bearbeitet. Diese Dokumente dienen als Träger von Informationen. Wie können bei einer Aktivität zugehörige Dokumente hinzugefügt werden um den Geschäftsprozess verständlich und vollständig mit eingehenden und ausgehenden Informationen darzustellen? Das Data Object ist ein Element aus der Kategorie Artefakte und kann elektronische und physische Objekte beschreiben. Ebenfalls ist eine Unterscheidung von eingehenden und ausgehenden Informationen möglich. Die Abbildung 2 zeigt auf welche Rolle das Data Object *Anfrage* im Prozessbeispiel spielt. Bei der Tätigkeit *Anfrage schreiben* wird ein physisches Dokument erstellt in welcher sich Informationen zu der Anfrage befinden. Dieses Dokument wird anschließend vom Kunden zum Vertrieb versendet und stößt gleichzeitig die weiteren Prozesse im Unternehmen an.

2.3 Dynamic Handling Pattern

Während der Modellierungsphase weiß der Modellierer nicht, wie die Situation bei der Ausführung an bestimmten Punkten eines Prozesses sein wird und wie viele Instanzen einer bestimmten Aktivität an diesen Punkten benötigt werden. Wie kann schon während der Modellierung eine Handhabe für solche Situationen sichergestellt werden? In einer solchen Situation kommt in BPMN die Schleife bzw. loop zum Einsatz. Die Schleife wird so lange durchlaufen, bis die notwendige Bedingung erfüllt ist. Nach jedem Durchlauf wird die Bedingung erneut überprüft. Dabei ist zu beachten, dass die Durchläufe sequenziell ablaufen.

Die nächste Iteration kann erst beginnen, wenn die vorherige beendet worden ist. In der *Abbildung 2* wird mit der Aktivität "Fehler beheben" genau so eine Schleife sichtbar gemacht. Während der Modellierung weiß der Modellierer nicht, wie viele Fehler während der Ausführung behoben werden müssen. Aus diesem Grund wird in diesem Prozessschritt eine sequenziell ablaufende Schleife eingebaut. Diese läuft solange bis alle Fehler behoben worden sind. BPMN bietet auch die Möglichkeit Schleifen einzusetzen, die parallel ablaufen. Die Beschreibung des Patterns kann in der Dokumentation nachgeschlagen werden.

3 Zusammenfassung und Ausblick

Auf der Grundlage BPMN 2.0 by Example und BPMN der Open Group, wurden Patterns definiert die das Verständnis und den Einsatz von BPMN fördern sollen. Dadurch lassen sich die definierten Fragen "Was ist die Essenz von BPMN?" und "Wie können die entwickelten BPMN Patterns klassifiziert werden?" beantworten und für spätere Forschungszwecke einsetzen. Zu beachten ist, dass in diesem Paper lediglich ein Auszug über Einsatz und Funktion der entwickelten Patterns gegeben wurde. Die ausführliche Beschreibung der weiteren Patterns kann in der Dokumentation nachgeschlagen werden. Die Einführung in BPM und die Beschreibung der entwickelten BPMN Patterns zeigen die zukünftige Wichtigkeit der Thematik um dadurch eine einfache und qualitative Nutzung von BPMN zu unterstützen.

Literaturverzeichnis

- [MW08] Miers, D.; White, S.A.: BPMN - Modeling and Reference Guide. Future Strategies Inc., Lighthouse Point, Florida, 2008
- [GR11] o.V. OMG (2011) o.V., Object Management Group - OMG (Hrsg.): Business Process Modeling Notation (BPMN), Version 2, Object Management Group (OMG) 2011, <http://www.omg.org/spec/BPMN/2.0/PDF> (20.11.2012 12:30)
- [GR10]]BPMN 2.0 by Example, Version 1.0, Object Management Group (OMG) Juni 2010 <http://www.omg.org/spec/BPMN/20100601/10-06-02.pdf> (20.11.2012 12:40)

Improving Performance and Precision of UniLoG's Load Generation Process by Utilizing Realtime Enhancing Features of the Operating System

Alexander Beifuss 7beifuss@informatik.uni-hamburg.de

University of Hamburg
Department of Computer Science
Telecommunications and Computer Networks (TKRN) Research Group

Form of work: Bachelor Thesis

Advisor: Prof. Dr. rer. nat. Bernd E. Wolfinger

1 Introduction

During the last years communication systems have become more and more complex. In order to describe and predict the behavior of such systems, simulation as well as analytical models are useful techniques. However, simulation and analytical models could require significant efforts in their construction and validation phase. Beside both of these methods there is another more practical approach, namely measurement. One of the challenges of measurements is the requirement to observe proper measurement points during intervals when a specific load is generated. A tool that allows the experimenter to produce a specific load is a load generator. However, if the produced load deviates from its specification, the experimenter might deduce wrong conclusions. While hardware load generators provide high precision, software based tools are more flexible but quite often less precise [DBP07]. Due to the flexibility of software and the high costs of hardware based solutions, the research community usually applies software variants. Therefore, it seems suitable to make an investigation on how software based load generators could be improved.

Most of the related work concerning the architecture of the generators itself, utilizes special hardware [APF⁺08] or distributes the load generators [AEPV04], but seldom covers the aspects of the operating system. For that reason the bachelor thesis summarized here, has been dedicated to this last mentioned topic.

This article is structured as follows: After providing the reader briefly with fundamentals on generating load in the second section, the third section illustrates why the operating system potentially has considerable influence on the load generator's performance and precision. In the fourth section the conducted measurements are introduced, followed by the presentation of the obtained results. Finally, the results are discussed and a short prospect for further work is given in the fifth section.

2 Basics

The UniLoG (Unified Load Generator), that is utilized in this case study, follows the principal of precise distinction between the specification, the generation and the execution process. This is achieved by applying a formal description technique based on state machines [KK10]. The specification process happens offline and results in a PUBA (Parameterized User Behavior Automaton). At runtime a GAR (Generator of Abstract Requests) component produces a sequence of abstract requests according to the PUBA, which in turn are converted to executable requests (w.r.t. interface IF) and executed by an interface specific adapter (ADAPT).

The current implementation of UniLoG is a single thread solution. As mentioned, at least two components are involved in the load generation process (GAR and ADAPT). To avoid waste of *CPU* time by possibly long busy waits for request execution time, a light weight scheduler (SCHED) is introduced as a third component. This scheduler is non pre-emptive (components have to return control by themselves) and decides control assignment dependent on the current state, in particular, based on internal queue states and statistics on component's time consumption. UniLoG's architecture is depicted in figure 1.

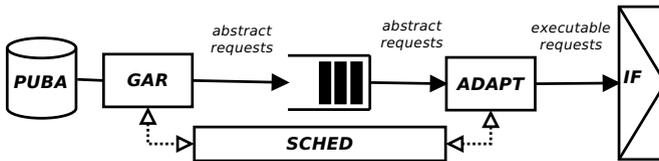


Figure 1: UniLoG's main components.

3 Influences of the Operating System on the Load Generating Process

As mentioned, the load generator manages component's control flow by itself. Therefore, it is desirable to have exclusive access to the *CPU* over the whole period of execution. Using a general purpose operating system, this demand seems unrealistic since many concurrent processes compete for computing time and it is in operating system's responsibility to assign the computation time equitably. Usually this is done by a preemptive scheduler, granting a quantum (time slice of computing time) to the next process with the highest priority. Since such a quantum is in the order of several milliseconds, it could be fatal for a load generator not to gain access to the *CPU* (especially when generating high-frequency packet rates).

Another effect coming along with the dispatched process, is that pages in cache might be replaced, which in turn will lead to page misses and may result in extra time required to refill the cache on reactivation. Eliminating as many concurrent processes as possible and assigning a high priority to the load generator are options to decrease the chance of being interrupted.

4 Measurements and Results

In order to assess the load generator’s performance as well as its precision, two kinds of measurements were performed on a reference machine with a 2.6 GHz CPU and 4 GB RAM. The first measurement focuses on the maximum packet rate the load generator can produce for requests with specific packet lengths and infinitesimal small *iat* (interarrival-time), since it is a typical characteristic for the performance of load generators [DBP07]. By a second measurement the mean missing time of request execution is determined to characterize the precision. Both measurements rely on the *UDP* Interface.

The described measurements were performed on two different operating systems. The first one is Microsoft Windows 7 (professional) and the second one is OnTime RTOS-32 [On], a proprietary realtime operating system with a micro kernel based on a Windows kernel. Windows 7’s process scheduler is preemptive but offers process prioritization and options for setting the CPU affinity of processes. Thus, the measurements were repeated with two configurations: c1, *normal* priority (which is the default for processes) and without setting the affinity; c2, *high* priority and exclusive bound CPU (both can be set by the task manager), also unneeded Windows services were disabled. In contrast to Windows 7, RTOS-32 offers the ability to manage the scheduling manually. However, since the load generator is the only present process, sharing the *CPU* is not an issue.

The following presented graphs are the result of a statistical evaluation of measured samples. Each measurement was performed ten times, generating CBR traffic (fixed *iat* and packet length) for 20 seconds as described above. The traffic was sent to a load sink, directly connected to the load generator by a 1 GBit Ethernet link.

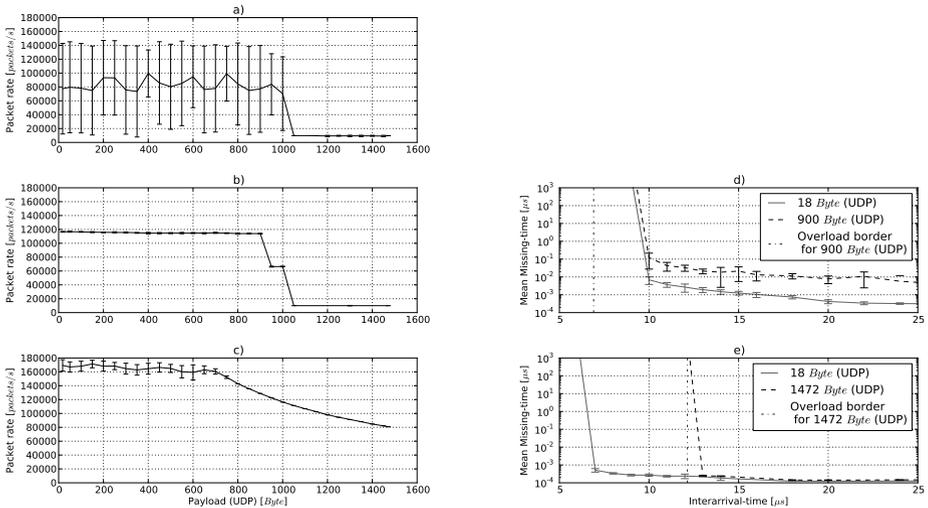


Figure 2: left: Achieved packet rate (avg., 95% confidence interval) plotted against packet length, top to bottom: a) Windows 7 c1, b) Windows 7 c2, c) RTOS-32; right: Missing time (average, minimum and maximum) plotted against specified interarrival-time, top to bottom: d) Windows 7 c2, e) RTOS-32.

5 Conclusion

Windows 7 (c1) is not a reliable environment to execute a load generator that should produce high-frequency packet rates, since fluctuations in packet rates are immense. However, making some efforts to increase the chance that *CPU* is continuously available for the load generator, the packet rate is stable for packet length lower than 900 Byte, as Fig. 2 a) shows. Recent investigations unveiled that the sudden drop on packet rate results from using a blocking socket with a low capacity buffer. A full buffer will block the whole process for at least one quantum and therefore prevent the load generator from generating and executing requests within these periods. Normally, this phenomenon is not a problem for common software, since no high-frequency packet rates are generated. Comparing the packet rate generated on RTOS-32 to that one on Windows 7 (c2), we see that using RTOS-32 the average packet rate is round about 33% higher. Moreover, the packet rate behaves as expected and decreases according to the network's capacity. Looking to the precision [Fig. 2 c) and d)] it is to mention, that under Windows 7 (c2) packets with a length up to 900 Byte and an *iat* larger than $10\mu s$ can be generated and executed with a miss smaller than $1\mu s$. On contrary, RTOS-32 allows to generate packets with *iat* of $7\mu s$ at very high precision. For packets with the length of MTU the results behave as expected and converge against the calculated value such a packet would need to be put on the link.

It has been shown, that the operating system and its adjustment definitely have an impact on the load generator's performance as well as its precision. Especially, the naive use of a general purpose operating system, without doing optimization efforts, could lead to faulty experiments if the experimenter relies upon the load specification. Future investigations could include a multi-threaded solution of UniLoG for realtime operating systems, where it could be ensured that each thread continuously gets computing time.

References

- [AEPV04] S. Avallone, D. Emma, A. Pescapè, and G. Ventre. A Distributed Multiplatform Architecture for Traffic Generation. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, 2004.
- [APF⁺08] G. Antichi, A. Di Pietro, D. Ficara, S. Giordano, G. Procissi, and F. Vitucci. Design of a High Performance Traffic Generator on Network Processor. In *DSD'08*, pages 438–441, 2008.
- [DBP07] A. Dainotti, A. Botta, and A. Pescapè. Do you know what you are generating? In *Proceedings of the 2007 ACM CoNEXT conference*, CoNEXT '07, pages 32:1–32:2, New York, NY, USA, 2007.
- [KK10] A. Kolesnikov and M. Kulas. Load Modeling and Generation for IP-Based Networks: A Unified Approach and Tool Support. In *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, volume 5987 of *Lecture Notes in Computer Science*, pages 91–106. Springer Berlin / Heidelberg, 2010.
- [On] On Time Informatik GmbH. Win32 API Compatible RTOS for 32/64-bit x86 Embedded Systems. <http://www.on-time.com/rtos-32.htm>, last checked: 29.12.2012.

Verschlüsselte Sprachkommunikation zwischen Android-Smartphones

André Gimbut

Martin-Luther-Universität Halle-Wittenberg
Institut für Informatik
andre.gimbut@digitrade.de

Art der Arbeit: Bachelorarbeit
Betreuer/in der Arbeit: Dr. Sandro Wefel

Abstract: Die mobile Kommunikation ist für die heutige Gesellschaft alltäglich. Bekannte Angriffe auf aktuelle Mobilfunkstandards wie GSM und UMTS beeinträchtigen die Vertraulichkeit der Gespräche. Die Bachelorarbeit beschreibt Ansätze zur Absicherung von Voice-over-IP-(VoIP-)Sprachverbindung durch Verschlüsselung und geeignete Authentifizierungsmechanismen.



1 Einleitung

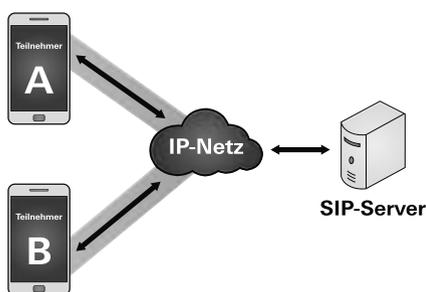
Heutzutage sind wir überall und jederzeit erreichbar und können mit anderen Personen in Kontakt treten. Die ständige telefonische Erreichbarkeit ist für die Geschäftswelt mittlerweile eine notwendige Voraussetzung, damit Entscheidungen schnell getroffen werden können. Dabei achten die wenigsten Teilnehmer in besonderer Weise auf die Sicherheit ihrer Telefonate.

Es gibt viele Gründe, weshalb eine Sprachverbindung abgehört wird. Das größte Interesse liegt sicherlich in den Bereichen Wirtschafts-, Regierungs- oder Militärsplionage. Ein Großteil der Mobilfunk-Teilnehmer misst der Gefahr des Abhörens einer Sprachverbindung keine ausreichende Bedeutung bei. Sie nehmen an, dass ein Abhören nur durch Staatsorgane ausgeführt werden kann, vor denen sie in der Regel nichts zu verbergen haben. Dies wird weitgehend darin begründet sein, dass es im Allgemeinen nicht bekannt ist, wie leicht normale GSM-Verbindungen entschlüsselt werden können. Laut einem Bericht des Heise-Verlags veröffentlichte der CCC (Chaos Computer Club) bereits 2010 ein Verfahren zum Brechen der bei GSM verwendeten A5/1-Verschlüsselung. „Mit einem gängigen Laptop und der selbstgebaute Abhöreinrichtung könne die Verschlüsselung des GSM-Funkverkehrs »in rund 20 Sekunden« geknackt werden.“ [1]

Daraus lässt sich schlussfolgern, dass der Aufwand und die Kosten zum Abhören von GSM-Verbindungen mittlerweile sehr marginal geworden sind. UMTS-Verbindungen sind zwar besser gesichert, doch haben Mobiltelefone in der Regel eine Abwärtskompatibilität zu GSM. So können auch UMTS-Verbindungen durch das Imitieren einer GSM-Funkzelle mit besserem Empfang abgehört werden.

Neben den bereits genannten Gefahren muss auch die Authentizität der Teilnehmer verifiziert werden. Hier ist es nicht besonders schwer eine bestimmte Nummer vorzutäuschen und durch angepasste Sprache sich als eine Vertrauensperson auszugeben, mit der sensible Informationen geteilt werden können. Viele dieser Probleme können durch Anwendung kryptografischer Verfahren behoben werden.

2 Lösungsansatz



Das allgemeine Konzept für VoIP zwischen Smartphones wird in der nebenstehenden Grafik gezeigt. Das wesentliche Sicherheitskriterium und der entscheidende Vorteil gegenüber der Standardverschlüsselung von GSM und UMTS ist die Ende-zu-Ende-Verschlüsselung, die direkt zwischen den beiden Teilnehmern und nicht über den Server aufgebaut wird. So ist es ausgeschlossen, dass Provider Zugriff auf die Verschlüsselung und somit auch auf die

übertragenen Informationen haben. Zusätzlich sollte eine Benutzerauthentifizierung mittels Zertifikaten erfolgen. Dafür ist eine „Certificate Authority“ (CA) notwendig, welche im Voraus die Authentizität der Teilnehmer überprüft und bestätigt. Außerdem ist es sehr wichtig, dass die privaten Schlüssel der Benutzerzertifikate so sicher abgelegt sind, dass kein Angreifer die Möglichkeit hat an diese zu gelangen. Dies könnte z.B. innerhalb einer Smartcard oder eines in dem Smartphone befindlichen TPM (Trusted Platform Module) sein.

Gute kryptografische Systeme halten sich an die Kerckhoffs'sche Maxime, welche besagt: Die Sicherheit eines Verschlüsselungsverfahrens beruht auf der Geheimhaltung des Schlüssels und nicht auf der Geheimhaltung des Verschlüsselungsalgorithmus [2]. Dies bedeutet, dass es von enormer Wichtigkeit ist die Schlüssel für die Verschlüsselung der Sprache gut zu wählen, geheim zu halten und am besten auch regelmäßig zu ändern. Optimal wird dies in einem hybriden Verschlüsselungssystem umgesetzt. Dabei werden die Schlüssel aus echten Zufallszahlen generiert und über ein asymmetrisches Verfahren, z.B. RSA, ausgetauscht. Die zertifikatbasierte Teilnehmerauthentifizierung hilft bei diesen Verfahren einen Man-in-the-middle-Angriff abzuwehren.

3 Umsetzung

Session Initiation Protocol (SIP) stellt die meist verwendete Möglichkeit dar, eine Verbindung zwischen VoIP-Teilnehmern herzustellen. Die Übertragung der eigentlichen Sprachinformationen erfolgt mittels Real-Time Transport Protocol-(RTP-)Paketen. Jedes RTP-Paket besteht aus einem 12-Byte großem Header und einem bis zu 160-Byte langem Content. Verschlüsselt wird nur der Content-Anteil in dem Paket. Dies gewährleistet die Kompatibilität mit nahezu allen SIP-Providern.

Als Basis wurde der SIP-Client Sipedroid ausgewählt, da er eine große Verbreitung hat und er nicht auf die SIP-API von Android aufbaut, sondern die notwendigen Protokolle selbst implementiert. Zu einer Verschlüsselung gehört auch ein geeigneter Authentifizierungsmechanismus. Bei dieser Anwendung wird eine 1-Faktor Authentifizierung mittels Passwort genutzt. Beide Teilnehmer verwenden das gleiche Passwort. Es muss hinreichend lang und zufällig gewählt sein. Wird das Passwort einem Dritten bekannt, so ist die Kommunikation nicht mehr vertraulich.

Als heutiger Industriestandard für Verschlüsselung gilt die Blockchiffre Advanced Encryption Standard (AES). Diese zeichnet sich durch eine hohe Sicherheit und Performance aus [3]. Der AES verschlüsselt nach NIST-Vorgaben immer 128-Bit große Datenblöcke [4]. Da die übertragenen Informationen bereits innerhalb eines RTP-Pakets zu groß für einen Block sind, muss eine Möglichkeit geschaffen werden, mehrere Blöcke zu verschlüsseln. Dafür gibt es mehrere standardisierte Blockmodi, die beschreiben, wie eine Information, aufgeteilt in mehrere Blöcke, mittels einer Blockchiffre verschlüsselt wird.

Es gibt sicherheitsrelevante Unterschiede zwischen verketteten und nicht verketteten Blockmodi. So sind bei dem ECB-Modus (nicht verkettet) auch nach der Verschlüsselung noch Strukturen erkennbar, welche bei einem verketteten Blockmodus (z.B. CBC) verschleiert werden. Für diese Arbeit wurde der CBC-Modus verwendet, da er neben einer einfachen und effizienten Implementierung auch eine akzeptable Fehlertoleranz aufweist. Außerdem eignet er sich für die RTP-Übertragung, bei der Pakete verloren gehen können. Ein weiteres Kriterium ist, dass dieser Modus die Grundlage für die CBC-MAC Integritätsprüfung ist, welche bei einer Weiterentwicklung angedacht ist.

Wichtig für eine sichere Verschlüsselung sind regelmäßig wechselnde Schlüssel. In dieser Implementierung wird jedes Mal zu Beginn eines neuen Telefonats bei jedem Teilnehmer ein zufälliger Session-Key erzeugt. Hierbei wird die SecureRandom Klasse aus „java.security“ verwendet. Der enthaltene RNG (random number generator) erzeugt Zufallszahlen, welche die Grundlage für die Session-Keys bilden. Hierzu wird zunächst aus dem hinterlegten Passwort ein SHA256-Hash gebildet. Dessen erste 128-Bit dienen als Verschlüsselungsschlüssel zur Ver- und Entschlüsselung der Session-Keys. Durch mehrfaches Durchführen des Hashing wird der Schutz vor Brute-Force-Attacks verbessert. Die Anzahl der Hash-Runden sollte möglichst hoch sein, aber für den Nutzer keine spürbaren Performance-Einschränkungen hervorrufen.

Der Austausch der Schlüssel ist etwas aufwändiger, da nicht angenommen werden kann, dass ein RTP-Paket tatsächlich beim Empfänger ankommt und verarbeitet werden kann. Da RTP ein Echtzeit Protokoll ist, werden verlorene Pakete nur protokolliert, nicht aber neu angefordert und sind somit für den Empfänger nicht verfügbar. Die Informationen in den Schlüsselpaketen sind allerdings für eine gesicherte Kommunikation eine notwendige Bedingung. Beim Start der Verbindung müssen beide Smartphones eine Reihe von Aufgaben abarbeiten, die je nach Rechenleistung und Auslastung unterschiedliche Zeit in Anspruch nehmen. Wird ein Schlüsselpaket an ein Smartphone geschickt, das dieses noch nicht verarbeiten kann, geht das Paket verloren. Um den Schlüssel zu erhalten, muss ein neues RTP-Paket mit dem Session-Key gesendet werden. Dazu wurde ein spezielles Verfahren implementiert. Beide Teilnehmer versenden zunächst RTP-Pakete, die einen speziellen Identifier und den verschlüsselten Session-Key beinhalten. Sobald ein Teilnehmer ein Schlüsselpaket erhalten hat, entschlüsselt er den Sitzungsschlüssel und behält ihn für dieses Gespräch. Des Weiteren erhalten alle Schlüsselpakete, die dieser Teilnehmer nun versendet, ein Acknowledge (ACK). Bekommt einer der Teilnehmer anschließend eines dieser Schlüsselpakete mit ACK, kann er davon ausgehen, dass beide Teilnehmer über beide Session-Keys verfügen und beginnt mit dem Senden der Sprachpakete. Wenn der andere Teilnehmer eines dieser Sprachpakete erhalten hat, beginnt dieser ebenfalls mit dem Senden von Sprachpaketen und das Gespräch kann begonnen werden.

4 Ergebnisse

Mit der im Rahmen dieser Bachelorarbeit entwickelten Erweiterung der VoIP-Software Sipdroid, können Anwender gut geschützt bzw. unter entsprechender Abgrenzung „sicher“ über VoIP-Verbindungen telefonieren. Damit zwei Teilnehmer sicher kommunizieren können, benötigen beide ein Android-Smartphone mit der entsprechenden App und einen SIP-Account bei einem (beliebigen) SIP-Provider. Zusätzlich müssen sich beide vorher auf ein gemeinsames Geheimnis (Passwort) einigen, das sie bei anschließenden Gesprächen authentifiziert. Die Verschlüsselung erfolgt mittels 128-Bit AES-CBC und für jede Verbindung werden am Anfang zwei zufällige Schlüssel erstellt.

Literaturverzeichnis

- [1] Stefan Krempf, „27C3: Abhören von GSM-Handys weiter erleichtert“, <http://www.heise.de/security/meldung/27C3-Abhoeren-von-GSM-Handys-weiter-erleichtert-1159542.html>, 28.08.2012
- [2] Auguste Kerckhoffs, „La cryptographie militaire.“, In: Journal des sciences militaires. Bd. 9, S. 5–38 (Jan. 1883), S. 161–191 (Feb. 1883)
- [3] National Institute of Standards and Technology (NIST), „Report on the Development of the Advanced Encryption Standard (AES)“, 02.10.2000
- [4] NIST, „FIPS 197, Advanced Encryption Standard (AES)“, 26.11.2001

Abstract zu „Analyse und Vergleich von Autorensystemen für ein WBT zu Vorlesungsinhalten“

Michael Gruben

michael.gruben@gmx.net

George Bernhard Shaw sagte einmal: „Hohe Bildung kann man dadurch beweisen, dass man die kompliziertesten Dinge auf einfache Art zu erläutern versteht“¹. Mit dieser Aussage bewegte er sich keineswegs auf einem Holzweg, denn der Grad der Bildung zeigt, wie tiefgründig eine Person über ein Thema bescheid weiß und wie gut diese das Thema selbst verstanden hat. Erklärt uns eine solche Person einen Sachverhalt, so bezeichnen wir sie ohne Zweifel als schlau. Wie kann die Informatik uns nun auf dem Weg zur Schlauheit begleiten und unterstützen und uns somit ein „Smart Life“ garantieren?

Die Studienarbeit mit dem Titel „Analyse und Vergleich von Autorensystemen für ein WBT² zu Vorlesungsinhalten“ greift grundlegende Fragestellungen auf, in wie weit Computer als Lernmittel eingesetzt werden können und was dabei beachtet werden sollte. Zu diesem Zweck ist die Studienarbeit zielorientiert aufgebaut und beginnt bei lerntheoretischen Grundlagen, um ein WBT zu definieren. Dieses WBT soll der Übung von Vorlesungsinhalten dienen und Tests anbieten, die dem Studenten ein Feedback über dessen Lernfortschritt aufzeigen. Ziel der Arbeit ist das Auffinden eines geeigneten Autorenwerkzeugs zum Erstellen des definierten WBTs.

Bedeutende Vorarbeiten zur Entscheidung, ob ein WBT qualitativ gut und zweckmäßig ist haben Dorothea Thomé ([Tho89]) und Stephan Benkert ([Ben08]) geleistet. Thomé erstellte bereits im Jahr 1989 eine „Prüfliste für Lernsoftware“ nach damaligen Maßstäben. Von der Qualität der Verpackung, der Dokumentation und des Datenträgers bis zur Zweckmäßigkeit der Software selbst werden alle wesentlichen Kriterien berücksichtigt. Im Jahr 2008 griff Benkert diese Liste auf und passte diese an den modernen Stand der Technik an.

Um fundierte Aussagen über die Qualität eines Lernangebots im Hinblick auf das studentische Umfeld zu treffen, werden im ersten Komplex der Arbeit theoretische Grundlagen für diese Studienarbeit erarbeitet. Es wird, im Wesentlichen basierend auf den Erläuterungen aus [Spi12], darüber aufgeklärt, was Lernen ist und was im Gehirn passiert, wenn Inhalte gelernt oder abgerufen werden. Damit einhergehend spielt die Motivation eine wesentliche Rolle, die mit deren zwei Kategorien eine Erklärung findet. Auch das sich über dem Lernen befindliche Wissen wird dahingehend untersucht, ob es durch einen Computer überhaupt vermittelt werden kann und in wie weit der Begriff des eLearnings als fragwürdig erscheint. Abgeschlossen werden die theoretischen Grundlagen anhand der Definitionen der

¹George Bernhard Shaw, Irischer Dramatiker, Schriftsteller und Nobelpreisträger, 1856-1950

²Web Based Training

technischen Begriffe eLearning, Gamification, Learning Management System, Autorensystem, SCORM und WBT.

Resultierend aus den Erkenntnissen der theoretischen Grundlagen und Definitionen wird im zweiten Komplex die Wahl einer geeigneten Software diskutiert. Dazu erfolgt zunächst das Erarbeiten von Anforderungen an Lernsoftware im Allgemeinen, an WBTs und schließlich auch an Autorensysteme. Die Kriterien werden gängiger Literatur, wie [Bau97] und [Wen03], entnommen. Mithilfe der erarbeiteten Kriterien wird eine fundierte Kurzbetrachtung von Autorensystemen möglich, die zusätzlich in einem gewichtetem Bewertungsschema gegenübergestellt werden können. Dieser Vergleich unterstützt eine Kategorisierung und Eingrenzung auf die Lösungen, die für eine weitere Betrachtung relevant sind, welche anschließend auch mit den Anforderungen aus dem theoretischen Teil abgeglichen werden. Dieses Vorgehen führt letztlich auch zur Wahl eines geeigneten Autorensystems für das Erstellen eines prototypischen WBTs, dessen bebilderte Dokumentation Inhalt des dritten Teils der Arbeit ist.

Zweck des vierten Komplexes ist eine allumfassende Auswertung der Ergebnisse der Arbeit. Es wird das entstandene WBT mit den zuvor gestellten Erwartungen verglichen. Hinzu kommt eine Erklärung entstandener Probleme und Chancen. Abgeschlossen wird die Studienarbeit mit einer Zukunftsbetrachtung, in der Vorschläge im weiteren Umgang mit den Ressourcen, für Evaluationen entstandener WBTs und zwei strategische Konzepte beschrieben werden, die den qualitativen und nutzenbringenden Einsatz der elektronischen Lernhilfe im studentischen Umfeld auch längerfristig unterstützen.

Zusammenfassend ist ein Werk entstanden in dem lerntheoretische Grundlagen umfangreich erklärt sind. Hinzu kommt eine Übersicht einiger Autorenwerkzeuge für die Erstellung von WBTs. Passend zum Auftrag der Studienarbeit fiel die Wahl auf eines der analysierten Werkzeuge, mit dem ein prototypisches WBT erstellt wurde. Anhand dessen sind die Gestaltungsmöglichkeiten eines WBTs ersichtlich. Am Ende steht ein hinweisgebendes Kapitel für die Einbindung von WBTs in den studentischen Unterricht.

Literatur

- [Bau97] Peter Baumgartner. *Didaktische Anforderungen an (multimediale) Lernsoftware*. Psychologie-Verl.-Union, Weinheim, 1997.
- [Ben08] Stephan Benkert. *Erweiterte Prüfliste für Lernsysteme – Kriterienkatalog zur (vergleichenden) Beurteilung multimedialer Lernsysteme*. Universitätsverlag Siegen, Siegen, 2008.
- [Spi12] Manfred Spitzer. *Digitale Demenz*. Droemer, München, 2012.
- [Tho89] Dorothea Thomé. *Kriterien zur Bewertung von Lernsoftware*. Dr. Alfred Hüthig Verlag GmbH, Heidelberg, 1989.
- [Wen03] Matthias Wendt. *CBT und WBT - konzipieren, entwickeln, gestalten*. Carl Hanser Verlag, München, Wien, 2003.

Design and Implementation of the UULM Network Lab

Michael Müller
Ulm University
michael-4.mueller@uni-ulm.de

Abstract: We describe the foundation of a network laboratory for educational purposes in which it is possible to set up and manage different network scenarios. Students should be able to gather a better understanding of the concepts, technical challenges and design decisions behind the Internet. For this purpose we develop an educational concept for such a laboratory, identify necessary components and compose an encompassing technical architecture. Furthermore a modular concept for a management system to administrate the laboratory and the devices is introduced.

1 Introduction

About forty years after the research into a global, distributed network first began, knowledge of networking technologies has now become an essential part of any computer science education. The Internet with its enormous complexity and various facilities is however often hard to understand. Lectures aim to provide a theoretical background and try to further practical insights by offering accompanying assignments. However, these assignments often only cover a small part of the subject and usually do not provide hands-on experience with hardware which is actually used in existing working environments. Hence, students do not generally have the possibility to gain practical knowledge on how to build network scenarios with hardware from a productive environment.

We therefore aim to build the foundation for a network laboratory in which it is possible to set up and manage different network scenarios. The laboratory's purpose should be to serve as an educational environment. Students should be able to gather a better understanding of the concepts, technical challenges and design decisions behind the Internet. Over the period of one semester, students should set up core technologies within the laboratory, thus creating a macroscopic version of the Internet as part of a practical course.

2 Educational Concept

We aim for students to gain further insights into the technologies and inner workings of the Internet by gaining practical experiences and observing in further detail which tasks different operators have to fulfill. For this purpose students will have to complete assignments from the perspective of a variety of the typical roles which can be found in the Internet. For example, the assignments will require students to take on tasks which an ISP typically

performs: to administrate an enclosed *Autonomous System* (AS), which coordinates with other ASs and offers services to customers (e.g. server/web/mail hosting). On a more abstract level students will be required to coordinate the IP address space with other groups, a task for which the ICANN is responsible.

We found that consecutive Bottom-Top assignments best fulfill our educational requirements. In order to provide students with the freedom to choose for themselves on how to solve assignments we provide as little pre-configuration as possible. For example, we will not preconfigure routing or provide fixed cabling. Students will observe their own network set-up growing from a very basic set-up to a complex network scenario. During the course the groups will at first configure the devices within their group and later configure them to coordinate with those of other groups. We think this gives more insight into the concepts behind the Internet and rather represents its spirit—many independent organizations coordinating and communicating.

There are several projects at other universities which can be considered related to ours [LWMF10, MHL⁺10, wai]. From our understanding, there are several key differences to our concept. We aim to encourage students to gain hands-on experience with physical hardware, while abstaining from hardware emulations and instead offer physical access to the devices.

3 Technical Architecture

The laboratory consists of a variety of components. Racks are used to physically group different devices into logical units. From an educational and technical perspective, a set-up consisting of a number of equally equipped racks is most suitable, thereby allowing for various equal systems which can be flexibly interconnected. For example, each rack could represent an AS and thus be used to interconnect with the other racks. At least three racks are necessary to build interesting scenarios: for instance, failover problems only make sense with at least three different entities; if the direct route between two entities fails, the system should be able to adapt and use the third entity to balance the loss. Equipping the racks with identical equipment can be justified by an educational advantage; enabling three different groups to each work on a rack. Similarly to the racks, at least three routers and switches for each rack are essential for interesting scenarios such as failover or load balancing problems. Each rack will be equipped with exactly three routers and switches in order to keep the laboratory simple whilst still having a broad variety of options for scenarios.

The students need to have access to a server in order to configure a web server or use different network tools. We decided to equip each rack with one server with multiple Ethernet interfaces.

A separate management server is used to administrate the network laboratory. Serial connections connect the management server to each switch, router and server in the laboratory. Its purpose is to configure the laboratory and flexibly deploy configurations to the different devices. Student groups work on the management server within their own virtual environ-

ment. The virtual machines restrain the groups' access to the devices which are assigned to a specific group.

We have additionally decided to set-up two separate VLANs in the laboratory. The *Student VLAN* is used by students to connect to different devices. The *Management VLAN* is used to deploy configurations for which the limited bandwidth of the serial lines is too small. This is for example necessary for the Virtual Machine images.

4 Management software

The network lab is set up in such a way that Out-of-Band management is supported: it is always possible to return to a working state. This is the single most highest priority of the management system, as otherwise a multitude of problems could occur. For this purpose, students working on devices do not have the necessary privilege levels to permanently change the device configuration. As a consequence, the system must administrate the necessary privilege levels for different assignments. Before an assignment starts, the privileges needed to complete the assignment have to be deployed to the devices.

Groups are able to save the state of their work at any time within a version control system. This provides a safe way of exploring the lab; by way of encouraging experimentation rather than constraining it. Thus we want to provide a safe way of being able to work with the lab. Flexibly deploying device configuration files also enables the students to enhance a specific network scenario without having to first build the entire scenario. For example, students may wish to enhance a company network e.g. with various security features. In this case, a company network scenario could be loaded and all devices configured accordingly.

We have implemented the management system within a first prototype. Our implementation hereby follows the Unix philosophy: Do one thing and do it right [Gan94, KP84]. We use different scripts for dedicated tasks, each fulfilling one function, remaining simple and small. A wrapper script is then used to compose the various scripts. Our initial set-up does not need any kind of database system. Instead, data is stored within flat ASCII files; a simple solution which has many additional advantages. It enables the use of tools from the Unix toolchest to rapidly build up the system logic. The project can easily be understood, maintained and enhanced using standard Unix tools (sed, grep, awk, etc.).

5 Conclusion

We have described the foundation of a network laboratory for educational purposes. Besides the educational concept, we have described a technical architecture and a management system to flexibly handle assignments. The concept has been implemented as a first prototype with real hardware and, as a next step, will be evaluated with student groups.

Literatur

- [Gan94] Mike Gancarz. *The UNIX Philosophy*. Digital Press, 1994.
- [KP84] B. W. Kernighan und R. Pike. *The Unix Programming Environment*. Prentice-Hall, Inc., 1984.
- [LWMF10] Dan Levin, Andreas Wundsam, Muhammad Amir Mehmood und Anja Feldmann. Berlin: The Berlin Experimental Router Laboratory for Innovative Networking. In Thomas Magedanz, Anastasius Gavras, Huu-Thanh Nguyen und Jeffrey S. Chase, Hrsg., *TRIDENTCOM*, Jgg. 46 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Seiten 602–604. Springer, 2010.
- [MHL⁺10] Amir Mehmood, Oliver Hohlfeld, Dan Levin, Andreas Wundsam, Florin Ciucu, Fabian Schneider, Anja Feldmann und Ralf-Peter Braun. The Routerlab à Emulating Internet Characteristics in a Room. In *Proceedings of 11th ITG Conference on Photonic Networks (11. ITG-Fachtagung Photonische Netze)*, Seite 201â208, Berlin / Offenbach, Germany, May 2010. VDE-Verlag. Poster Session.
- [wai] The Wisconsin Advanced Internet Laboratory (WAIL) is part of the Computer Science Department at the University of Wisconsin - Madison. WAIL is home to the network research group led by Professor Paul Barford, Professor Suman Banerjee and Professor Cristian Estan.

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 Language to Information Systems, NLDB'2001.
- P-4 H. Wörn, J. Mühlhng, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensor-gestützte Chirurgie; Workshop des SFB 414.
- P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.
- P-6 Hans-Jürgen Appelpath, 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 – Erfahrungen und Visionen
- P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems
- P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications
- P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: 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 Rannenber, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)
- P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenber, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)
- P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenber (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.): Persistence, Scalability, Transactions – Database Mechanisms for Mobile Applications
- P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning, E-Services
- P-45 Bernhard Rumpe, Wolfgang Hesse (Hrsg.): Modellierung 2004
- P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment
- P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society
- P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications
- P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und 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 Ad-hoc-Netze Medienintegration
- P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government
- P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen
- P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality
- P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Rannenber, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments
- P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für informatische 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 its Applications

- P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005
- P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolfrid 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 Kowalczyk, 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. Morgens-tern, 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 Magenheimer, 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 Walthert (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 management 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 Special 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 Kommunikationstechnologien 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 und 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 Informationssysteme (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
I²CS 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. Claudepein, 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 (I²CS) – 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 (I²CS)
- P-187 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
4. DFN-Forum Kommunikationstechnologien, 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 Landwirtschaft 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 Kommunikationstechnologien
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-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

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

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