

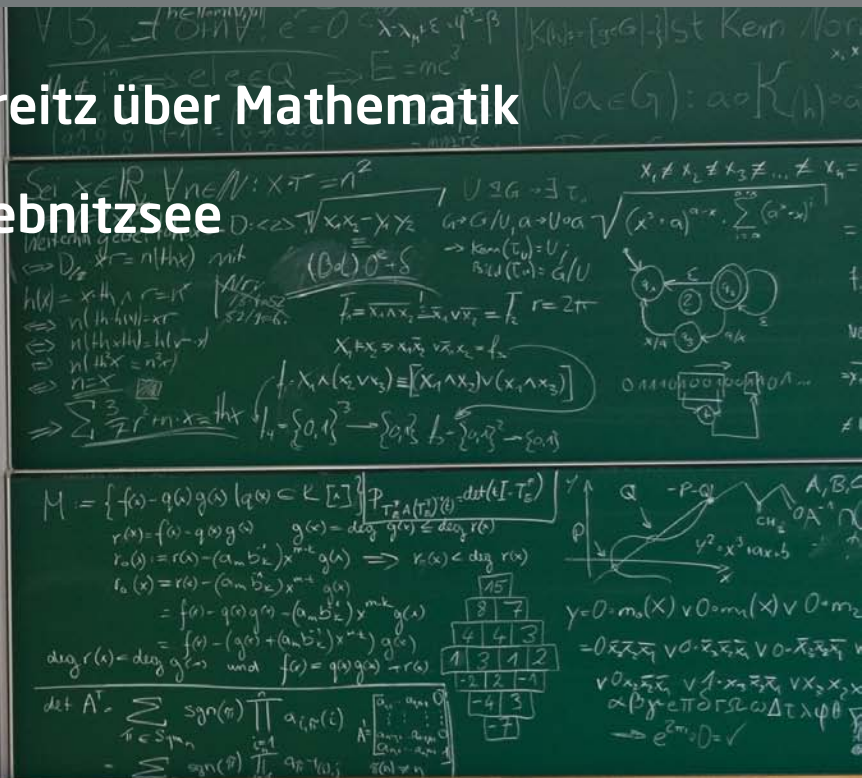


HPI mgzn

Hasso-Plattner-Institut

Ausgabe 9 - Winter 11

Herr Dr. Börner und Prof. Kreitz über Mathematik
Geschichte des Campus Griebnitzsee
Berichte von HPI Alumni





Editorial

28 Wochen und zwei Präsidentenbesuche sind seit dem vorherigen HPImgzn vergangen. Nun haltet ihr die neunte Ausgabe in Händen. In dieser Ausgabe klären wir, warum auch theoretische Grundlagen für IT-Systems Engineering von Bedeutung sind. Prof. Kreitz erklärt, warum wir die theoretische Informatik bei sicherheitsrelevanten Anwendungen brauchen. Dr. Börner erläutert in einem Gespräch, was uns die Mathematik bedeuten kann.

Darüber hinaus erzählt ein Alumnus aus seinem Berufsleben nach dem HPI und schildert seine heutige Sicht auf sein Studium. Wo es euch nach dem Studium hinzieht, habt ihr in der Umfrage selbst entschieden.

Nachdem wir viel über den Neubau berichtet haben, schauen wir nun einmal in einem Artikel in die Vergangenheit des Campus. Im Anschluss daran lassen wir es uns aber nicht nehmen, noch einmal über den Neubau zu schreiben.

Klubeigene Autoren dieser Ausgabe waren Maria Graber, Robin Schreiber, Leonid Berov, Andrina Mascher, Magdalena Noffke, Patrick Rein, Sebastian

Oergel, Stefan Schaefer, Franz Liedke, Jan Taske, Susanne Bülow und David Heller. Josefine Harzmann hat uns bei den Korrekturen unterstützt.

Grafisch umgesetzt haben die Zeitung Robin Schreiber, Magdalena Noffke, David Heller und Patrick Rein. Maria Graber hat die Leitung des Lektorats übernommen.

Wir danken den zahlreichen externen Autoren und wünschen nun viel Spaß beim Lesen der neunten Ausgabe des HPImgzns.

Magdalena Noffke & Patrick Rein

Kontakt

Redaktion:

klub-zeitung@hpi.uni-potsdam.de

Klubsprecher:

Magdalena.Noffke@student.hpi.uni-potsdam.de

Patrick.Rein@student.hpi.uni-potsdam.de

V.i.S.d.P.:

Magdalena Noffke

Patrick Rein



Intern

Alumni Erfahrungsbericht

Ein Weihnachtsbaum fürs HPI

HPI@SAP

Umfrage: Wo wollt ihr hin?



2

7

8

10

Forschung

Auf eine Zigarette mit Dr. Börner

Warum Grundlagen lernen?

Forschung am HPI

12

18

22

Leben

Vom Roten Kreuz zum roten Backstein	31
Das neue Gebäude: Mehr Platz für alle...	34
Wie gründet man ein HPI?	35

Verschiedenes

Sport am HPI	36
2. HPI Fotowettbewerb	37
Was den Teilnehmern der HPI-Sudoku-Meisterschaft erspart blieb...	40
Thorin und die zwölf Orks	43
Kurzgefasst	44

„Am meisten hat mich das Miteinander beeinflusst...“



Alumni-Erfahrungsbericht von Johannes Nicolai, CollabNet

Alumni-Club: Du arbeitest seit April 2009 bei CollabNet, Hauptsponsor von Subversion und einem der führenden Anbieter von verteilten Entwicklungsplattformen, Cloud Computing Lösungen und agilem Coaching. Wie hast Du den Einstieg bei diesem Unternehmen geschafft?

Nicolai: Nach meinem Bachelorabschluss am HPI habe ich zunächst im August 2007 ein Praktikum bei einer Münchner Partnerfirma von CollabNet begonnen. Meine Aufgabe war es, die SAP-basierte Transportmanagement-Software dieser Firma mit SourceForge Enterprise Edition (von CollabNet) abzugleichen. Nach zwei Monaten bin ich dann für weitere sieben Monate nach San Francisco gegangen, um das Integrationsprodukt in der CollabNet-Zentrale fertigzustellen. Von der technischen Konzeption über Product Marketing bis zum Vertrieb konnte ich damals fast alle Abteilungen durchlaufen. Außerdem war ich Teil des Subversion 1.5 Launch-Teams: Die HPI-Studenten, die im Wintersemester 2007/2008 eine Veranstaltung am Computergrafik-Lehrstuhl belegt haben, waren die ersten, welche die neuen Merge-Tracking-Features von Subversion ausprobieren konnten. Nach dem Praktikum habe ich mein Masterstudium am HPI fortgesetzt und nebenbei als Werksstudent bei CollabNet die Entwicklung einer neuen Open Source-Integrationsplattform geleitet. Meine Masterarbeit am Lehrstuhl für Business Process Technologies habe ich ebenfalls in Kollaboration mit CollabNet

geschrieben. Nach meinem Masterabschluss war es für mich keine Frage, eine Vollzeitstelle als Software Engineer bei CollabNet anzunehmen.

Alumni-Club: Welche Aufgaben hast Du bei CollabNet übernommen?

Nicolai: Wenn CollabNet TeamForge/Scrumworks in einem Unternehmen eingeführt wird, existieren dort typischerweise bereits Testplattformen wie HP Performance Center und Functional Testing, Requirements-Management Systeme wie DOORS und HP Quality Center, Bugtracker wie Bugzilla und Jira und ALM-Plattformen wie MS Team Foundation Server und Rational Team Concert. Die Daten dieser Bestandsysteme müssen mit unseren Lösungen bidirektional synchronisiert oder komplett migriert werden. Dafür nutzen wir eine Weiterentwicklung des Integrationsframeworks, das in meiner Werk-

studentenzeit entstand: Das CollabNet Connector Framework (CCF). Am CollabNet Connector Framework arbeiten derzeit sechs Entwickler/Tester, ein Technical Writer und ein Product Manager. Ich bin Development Manager/ Büroleiter Berlin für die technische Produktplanung und Koordination des global verteilten

als



Teams (Indien, USA und Deutschland) zuständig. Weiterhin bin ich der technische Ansprechpartner für strategische Partnerschaften mit Unternehmen wie HP, rekrutiere Softwareentwickler und Praktikanten in Deutschland und vertrete CollabNet auf Industriekonferenzen und Customer Advisory Boards.

Alumni-Club: Was macht den besonderen Reiz dieses Arbeitgebers aus? Und wo liegen die Herausforderungen?

Nicolai: Bereits vor meinem Studium war ich von Open Source und verteilter Softwareentwicklung fasziniert. Ich hatte mir es allerdings nie träumen lassen, einmal für die Firma arbeiten zu können, welche mit tigris.org, java.net und openoffice.org einige der größten Open Source Community-Plattformen weltweit betreut und über 5000 Kunden im Enterprise Bereich zu allen Fragen rund um verteilte Softwareentwicklung, Cloud Computing und agiles Projektmanagement berät sowie die notwendige Infrastruktur bereitstellt. Mit ungefähr 500 Mitarbeitern ist CollabNet groß genug, um die Softwareindustrie zu beeinflussen zu können und gleichzeitig noch klein genug, um als Individuum noch Veränderungen erzielen zu können. So ist die Umbenennung von SourceForge zu TeamForge auf eine Idee zurückzuführen, welche ich in meiner Zeit als Praktikant von CollabNet hatte. Die Herausforderungen liegen darin zu verstehen, wie quartalsgetrieben

eine amerikanische Softwarefirma tickt und Ideen so zu kommunizieren, dass sie genug Unterstützung finden, um nicht auf irgendeiner Hierarchiestufe versickern.

Alumni-Club: Wie würdest Du rückblickend Dein Studium beurteilen: Hat es Dich gut auf die jetzigen Aufgaben vorbereitet?

Nicolai: Ich denke, dass insbesondere die Softkill- und Englischkurse am HPI mir geholfen haben, Ideen wirkungsvoll zu kommunizieren. Während meiner Zeit im Fachschaftsrat habe ich gelernt, mit verschiedensten Interessengruppen zu verhandeln. Ohne mein Bachelorprojekt hätte ich wahrscheinlich nicht das Praktikum bei CollabNet's Partnerfirma bekommen. Am meisten hat mich aber wahrscheinlich das Miteinander mit meinen Kommilitonen beeinflusst. Die damalige Fokussierung des HPIs auf arbeitsteilige Entwicklung komplexer Systeme hat unser Verständnis von Software Engineering erheblich geprägt und oft den Unterschied zwischen hochspezialisierten Einzelkämpfern und einem wirkungsvollen Team gemacht.

Alumni-Club: Woran erinnerst Du Dich besonders gerne, wenn Du an die Zeit am HPI zurückdenkst?

Nicolai: Die hervorragende individuelle Betreuung, die sehr gute Unterstützung durch Frau Pamperin und Frau Seidler und das Leben im Studentendorf



werde ich immer positiv in Erinnerung halten. Die Zeit im Fachschaftsrat und im Bachelorprojekt sind wahrscheinlich meine persönlichen Highlights, da ich dadurch Freundschaften gefunden habe, die hoffentlich noch weit über das Studium bestanden haben.

Alumni-Club: Gibt es etwas, dass Du heute anders machen würdest, wenn Du nochmal als Erstsemester am HPI ins Studium starten könntest? (Was würdest Du Erstis empfehlen?)

Nicolai: Meine Empfehlung ist es, sich neben dem Studium genügend Freiräume zu schaffen, um sich

selbst zu verwirklichen. Ich habe den Eindruck, dass dies am HPI zunehmend schwieriger wird, da immer mehr Veranstaltungen ins Pflichtprogramm rutschen und weniger Zeit für Freizeit, eigene Projekte und Praktika bleibt. Meiner Meinung nach entwickelt man schnell selbst ein Gespür dafür, welche Veranstaltungen man wirklich braucht und hat bei weniger Pflichtveranstaltungen auch mehr Muße, nicht nur das allernotwendigste für das Studium zu tun. Studieren tut man meistens nur wenige Jahre, das Arbeitsleben ist fast zehnmal so lang, also genießt Eure Zeit.

– Das Interview führte der Alumni-Club



Ein Weihnachtsbaum fürs HPI!

Wer in den ersten Dezember-Tagen den Versuchsungen von Tele-Task getrotzt und sich trotz S-Bahn-Chaos, tiefem Schnee und eisigen Winden zum HPI-Hörsaalgebäude durchkämpfen konnte... wird dort keine Veränderungen bemerkt haben. Das ist eigentlich schade, dachte sich der Alumni-Klub und scheute keine Mühen, gemeinsam mit den HPI-Alumni einen Weihnachtsbaum zu schlagen, um das HSG in vorweihnachtliche Schwingungen zu versetzen. Eine kleine Gruppe Wagemutiger begab sich dazu in den tief verschneiten Potsdamer Wildpark und hatte inmitten unzähliger Märkischer Kiefern zunächst einige Mühe, den „richtigen“ Baum auszuwählen. Deshalb wurden es dann auch zwei. Das Fällen oder Schlagen gestaltete sich erstaunlich unkompliziert und hat dank optimaler Ausrüstung nur den beiden Kiefern richtig wehgetan.

Bei Lichte betrachtet sind auch die schönsten HPI-Alumni-Weihnachtsbäume nicht von perfektem Wuchs. Und doch entdeckten wir beim Aufstellen im HSG schnell die Schauseite wieder, die schon im Wald zu überzeugen wusste. Während es draußen langsam dunkel wurde, fanden drinnen Zweig um Zweig viele rote Kugeln und auch der gespendete Alumni-Baumschmuck ihren Platz am Weihnachtsbaum. Und schon kündeten ungezählte helle Lichter von der festlichen Pracht und tauchten das weite Hörsaal-Foyer in einen warmen Schein. Vom Glanze beseelt fand das Alumni-Event bei Glühwein und Chili einen würdigen Abschluss.

HPI Hasso Plattner Institut Alumni
www.hpi-alumni.de



HPI@SAP

Mit dem Seemann nach Walldorf

Kalt und dunkel war die Nacht ... und mit nur einer halben Stunde Verspätung ging es dann in der Nacht vom 24. zum 25. November 2010 vom HPI mit dem „Seemann“-Bus zu SAP nach Walldorf. Doch bevor wir dort ankamen, haben wir noch einen Abstecher nach Heidelberg gemacht, um uns die dortige traditionsreiche und sicherlich sehenswerte Altstadt anzusehen – wären wir dort nicht mit kaltem Wind und Schneeregen begrüßt worden. So wurde aus dem geplanten Stadtrundgang mehr ein „Von-einem-Dach-zum-nächsten-Huschen“. Etwa anderthalb Stunden später sind wir dann in aller Nässe nach Walldorf gefahren, um kurz unsere Sachen im Hotel abzugeben und uns dann gleich auf den Weg zu SAP zu machen. Bereits in Heidelberg wurden wir von Herrn Pleyer von SAP begrüßt und durch ein vielseitiges Programm geführt. Bevor wir uns in dem großen und umfangreich ausgestatteten Konferenzraum im SAP-Hauptgebäude einfanden, haben wir noch eine Rundfahrt durch die Gewerbegebiete in Walldorf und St. Leon-Rot gemacht, in denen die Firma in der Region angesiedelt ist. Dadurch bekamen wir einen ersten Eindruck von der Größe dieses Unternehmens und der Vielfalt der SAP-Welt.



Der SAP-Campus in Walldorf

Wieder zurück in Walldorf begann dann ein umfangreiches Vortragsprogramm. Wir bekamen einen Einblick in die verschiedenen Bereiche von SAP – ohne Anspruch auf Vollständigkeit oder Konsistenz. Aber das war auch gar nicht das, was uns geboten werden sollte, denn in zwei Tagen ist eine solch um-

fassende Gesamtübersicht über dieses riesige und global agierende Unternehmen gar nicht möglich. Die verschiedenen Vorträge boten Einblicke in aktuelle Entwicklungen. Die Themen reichten dabei von den neuesten „In-Memory Database“-Entwicklungen über die klassische SAP-Standardsoftware bis hin zu den neuesten Entwicklungen im mobilen Bereich, bei denen unter anderem Managern die Möglichkeit gegeben werden soll, Reports und Analysen unterwegs komfortabel und intuitiv auf dem iPad zu betrachten.

Am Abend trafen wir uns dann alle zum gemütlichen „Get together“ in der Walldorfer „Marktstube“, um den ersten Tag ausklingen zu lassen. Dank der Gastfreundschaft von SAP wurden wir dabei in jeglicher Hinsicht gut versorgt – nicht nur die Verpflegung, auch das Hotel ließ keine Wünsche offen.

Am nächsten Tag ging es dann in die nächste Vortragsrunde. Detailliertere Erklärungen zu Themen vom Tag zuvor und weitere Einblicke in das Unternehmen standen auf dem Programm. Bei einer Führung durch die Walldorfer SAP-Welt konnten wir uns dann auch anschauen, womit das Unternehmen sich schon am Vortag als attraktiver Arbeitgeber präsentiert hat. Für die Mitarbeiter wird in vielerlei Hinsicht gesorgt: man hat die Möglichkeit, die SAP-Fitnessräume, die Physiotherapie, Ärzte und sogar einen Friseur, Bankfilialen und eine Apotheke in Anspruch zu nehmen. Auch für die Kleinen wird gesorgt: So kann man bei einem unerwarteten Ausfall der Kinderbetreuung sein Kind mit zu SAP bringen, wo es umfassend versorgt und beschäftigt wird, während die Eltern im Raum nebenan arbeiten.

Nach dem Rundgang wurden wir in die Mensa eingeladen, wo uns eine große Auswahl an verschiedenen Gerichten inklusive Vor- und Nachspeisen erwartete. Nach den letzten Vorträgen stiegen wir dann sogleich in den Bus und machten uns erschöpft, aber sicherlich beeindruckt, auf den Rückweg nach Potsdam.

Ein herzliches Dankeschön auch an dieser Stelle noch einmal an Mychajlo Wolowyk, der die Fahrt seitens des HPI-Connect-Klubs organisiert hat, und vor allem an Adolf Pleyer und seine Kollegen von SAP, die uns durch das zweitägige Programm begleitet und uns tolle Präsentationen geboten haben. Vielen Dank natürlich auch an das Unternehmen SAP, das uns diese Fahrt und den Aufenthalt finanziell ermöglicht hat.

– Sebastian Oergel



Die Teilnehmer der SAP-Fahrt

IT TOOK A LOT OF WORK, BUT THIS LATEST LINUX PATCH ENABLES SUPPORT FOR MACHINES WITH 4,096 CPUs, UP FROM THE OLD LIMIT OF 1,024.

DO YOU HAVE SUPPORT FOR SMOOTH FULL-SCREEN FLASH VIDEO YET?

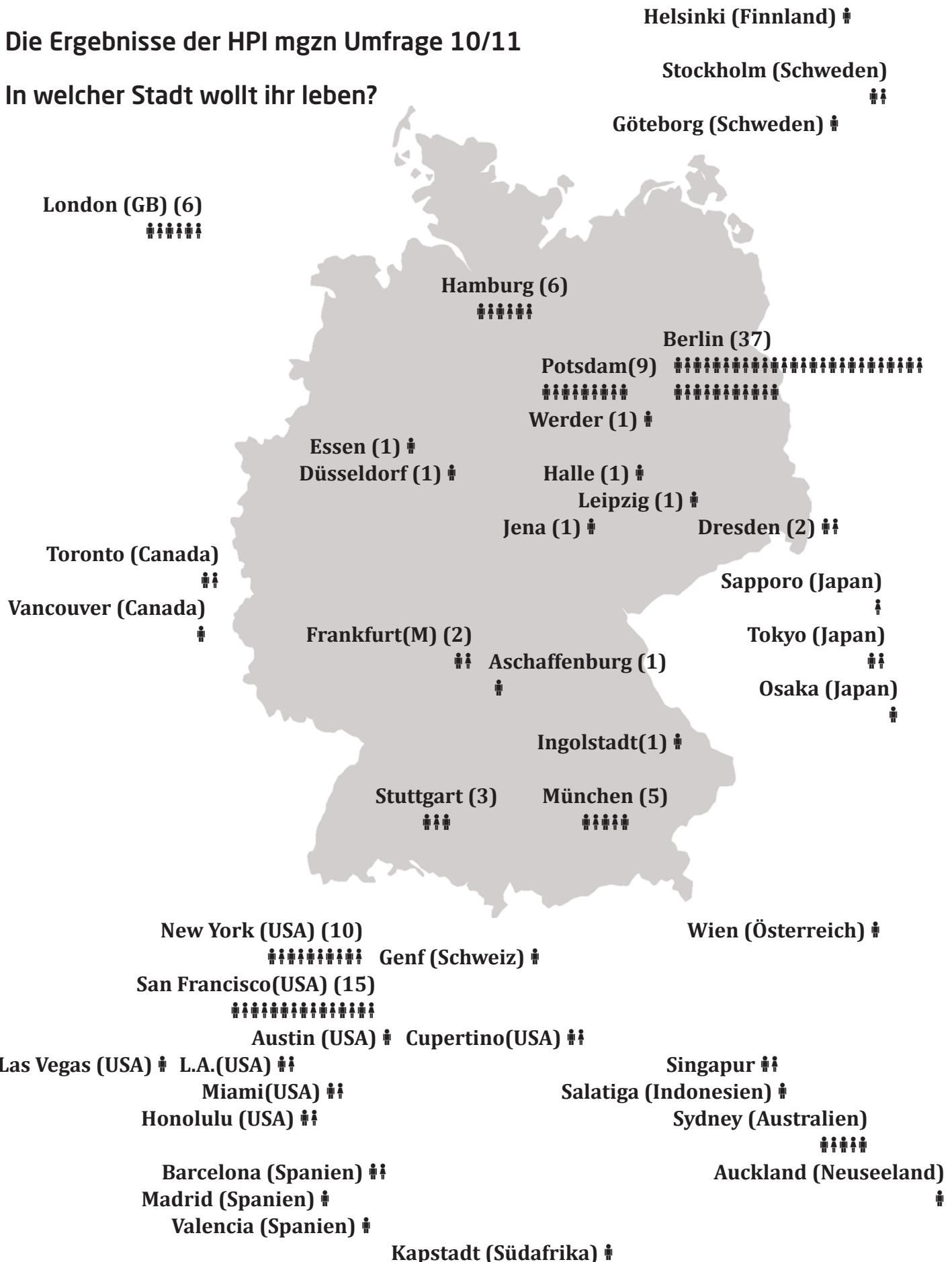
NO, BUT WHO USES THAT?



Wo wollt ihr hin?

Die Ergebnisse der HPI mgzn Umfrage 10/11

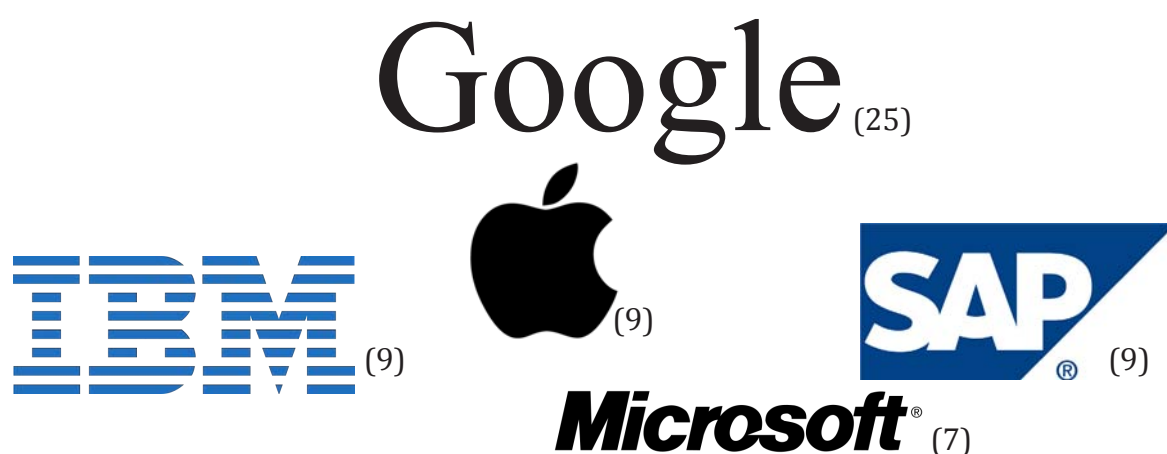
In welcher Stadt wollt ihr leben?



In welchem Land wollt ihr leben?

australien (8) belgien (2) canada (8) china (2)
 deutschland (50) england (7) finnland (3) frankreich (4)
 gb (7) indien (2) indonesien (1) irland (1) island (1) israel (1) japan (3) mexiko (1) neuseeland (2)
 niederlande (2) norwegen (3) osterreich (1) polen (1) schweden (5) schweiz (7)
 singapur (2) spanien (7) sudafrika (1) usa (37)

Bei welchem Unternehmen wollt ihr arbeiten?



Eigene Firma 8

Weiß nicht 6

Start-Up 5

HPI 5

Facebook 3

Twitter 2

Blizzard 2

MS Research 2

Selbstständig

Ubisoft

Piranha Bytes

Github

Heroku

Finn

Universität

Forschungseinrichtung

MIT (Research User Interface)

CERN

Fraunhofer Gesellschaft

Airbus

Boeing

Daimler Chrysler

Bombardier

Audi

Porsche

VW

Uni Karlsruhe

Omikron

Kiva

Open Source Firma

TrueCryptFoundation

Apache Foundation

SUN

IDEO

Finnlabs

SoundCloud

WG Systemhaus für

Informationstechnologie

SD&M

Apoxo GmbH

Lufthansa

Teradata

Nicht SAP

Schering

Kleines Unternehmen für

Schülersoftware

Mittelstand

Maxon

Carl Zeiss

DeutscheBank

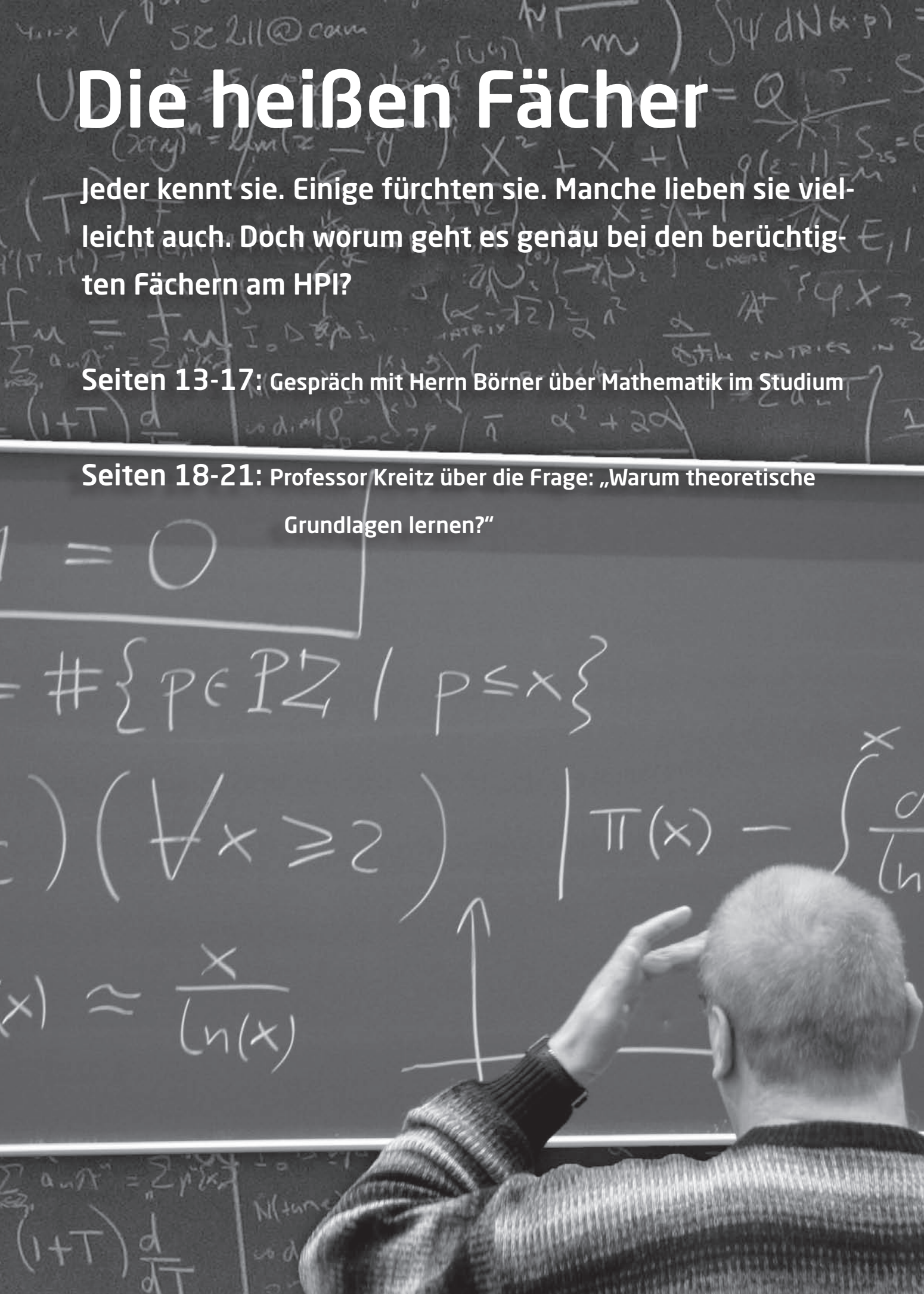
Geldspeicher von Dagobert

Die heißen Fächer

Jeder kennt sie. Einige fürchten sie. Manche lieben sie vielleicht auch. Doch worum geht es genau bei den berühmtesten Fächern am HPI?

Seiten 13-17: Gespräch mit Herrn Börner über Mathematik im Studium

Seiten 18-21: Professor Kreitz über die Frage: „Warum theoretische Grundlagen lernen?“



Auf eine Zigarette mit Dr. Börner

Ein Gespräch mit Herrn Börner über die Bedeutung der Mathematik im Studium und warum NP-Vollständigkeit nicht immer Unlösbarkeit bedeutet.

HPI-mgzn: Herr Dr. Börner, einem Studenten am HPI laufen im Studium hier oft solche mathematischen Teilbereiche über den Weg, wo einem nicht sofort klar ist wofür man sie braucht. Die meisten hören dann zum ersten mal von irgendwelchen Ringen und Körpern, haben das noch nie gehört und fragen sich jetzt: Warum braucht man das?

Dr. Börner: Ringe, Körper und vor allem Vektorräume gehören nun einmal zu den mathematischen Grundstrukturen. Vielleicht ist manchmal der spezielle Stoff gar nicht so wichtig. Wie auch? Man kann doch nicht vorhersagen, mit welchen speziellen Themen die Studenten später im Berufsleben beschäftigt sein werden. Daher muß hier am HPI in der Mathematik ein allgemeines Bild vermittelt werden. Aber mit irgendwelchen Strukturen werden sie zu tun haben, und dann ist es gut, wenn sie wissen was z.B. ein Isomorphismus ist, oder eine Unterstruktur. Ich persönlich halte das „Denken in Strukturen“ für Informatiker für extrem wichtig.

Stimmt. Gerade beim Programmieren entdeckt man das auch des Öfteren wieder. Es sind ja aber nicht nur die Probleme sondern auch die Definitionen, die man in Mathematik dann aufstellt, die ebenfalls eindeutig und genau sind.

Ja, was genau kann denn die Mathematik? Zu allererst liefert sie einmal eine Sprache. Eine Fachsprache, in der man Probleme formulieren kann, und zwar ganz exakt und ganz genau. Und oft hat man mit solch einer mathematischen Formulierung auch schon einen ersten Ansatz zur Lösung.

Zudem liefert die Mathematik ja auch ein breites Instrumentarium. Es gibt die Grundstrukturen, die im ersten Semester unterrichtet werden: Mengen, Relationen, Funktionen, Graphen, die absoluten Grundbegriffe. Wenn man in diesen Grundbegriffen den-

ken und formulieren kann, wenn man versteht, wie man den Mengenkalkül handhabt und weiß, was eine Relation und ein Graph ist, dann ist das schon ein erster Erfolg.

Danach wird es dann ein bisschen komplizierter weil man dann nicht mehr nur einfach eine Menge betrachtet, sondern eine Menge mit einer bestimmten Struktur. Und da gehören dann zum Beispiel diese Ringe und Körper hin, nach denen ihr vorher gefragt habt. Aber unsere Standardbeispiele wie der Ring der ganzen Zahlen und der Körper der reellen Zahlen sind ja schon aus der Schule bekannt - auch wenn sie dort nicht unbedingt Ring oder Körper genannt werden.

Wo genau werden diese besagten Strukturen in der Praxis dann überhaupt angewendet?

Für die Informatik sind die genannten Strukturen im Grunde noch viel zu einfach. Die dort relevanten Strukturen haben oft mehrere Grundmengen. Bei einer Datenbank zum Beispiel hat man ja nicht etwa nur eine Sorte von Daten, sondern man hat Namen, Adressen, Zeitdaten und vieles mehr. Die zulässigen Operationen sind oft nur partiell definiert. Aber die Grundprinzipien des Arbeitens sind so ähnlich wie bei unseren beispielhaften Strukturen, die nur eine Grundmenge haben.

Die wichtigste Struktur, die im zweiten Semester behandelt wird, ist der Vektorraum, und damit zusammenhängend die Lineare Algebra. Wir werfen da z.B. noch einmal einen Blick auf lineare Gleichungssysteme, allerdings ein bisschen anders als in der Schule. Das bringt Vorteile: Wir können mit Hilfe der Vektorräume beschreiben, wie Lösungsmengen prinzipiell aussehen. Zum Beispiel ist die Lösungsmenge eines homogenen Gleichungssystems immer ein Untervektorraum.

Für die praktischen Anwendungen in der Informatik sind dann beispielsweise Vektorräume über endlichen Körpern enorm wichtig, insbesondere über Körpern, deren Betrag eine Zweierpotenz ist. Wenn Sie sich jemals ernsthaft mit Kryptographie oder Codierungstheorie oder technischer Informatik beschäftigen, dann brauchen Sie das unbedingt. Deshalb müssen wir am HPI in der Mathe-Ausbildung diese Grundlagen behandeln. Zumindest hat dann jeder hier schon einmal diese Begriffe gehört, kennt die Grundtatsachen und weiß, wo er nachschauen kann.

Was nun aus dem Stoff der Mathematikvorlesung im späteren Leben wirklich vorkommt, das kann man nicht punktgenau vorhersagen. Dafür ist die Informatik mit ihren vielen Berufsbildern viel zu diversifiziert. Aber die mathematische Denkmethode, die muss trainiert werden, die ist überall sehr wichtig.

In welchen Bereichen der Mathematik wird zur Zeit besonders viel geforscht? Was sind denn die Grenzen, die man heute versucht zu durchbrechen?

Man muss erst einmal sagen, dass die Mathematik in viele Teilgebiete aufgespalten ist: Oberbegriffe sind beispielsweise Algebra, Analysis, Zahlentheorie, Stochastik und - hier für uns besonders wichtig - die Diskrete Mathematik. Jedes dieser Gebiete hat eigene Probleme. Zu den wichtigsten gehören sicher die sieben sogenannten „Millenniumsprobleme“, die 2000 vom „Clay Mathematics Institute“ gestellt wurden. Eines davon ist bereits gelöst. Unter anderem findet man dort auch das bekannte P-NP-Problem, das als wichtigstes Problem der Theoretischen Informatik gilt.

Speziell in der Algebra geht es natürlich um das Verständnis algebraischer Strukturen.

Können sie ein Beispiel für diesen Bereich geben?

Nehmen wir einmal folgendes: Eine endliche Struktur besteht aus einer endlichen Menge und bestimmten Dingen, die dieser Menge eine Struktur

geben. Oft sind das Operationen, wie bei einer Gruppe mit ihrer Gruppenmultiplikation. Jetzt möchte man große und komplizierte Strukturen besser verstehen, und zwar möchte man sie zurückführen auf kleinere und besser verstandene. Da helfen die sogenannten Kongruenzrelationen, die wir in der Vorlesung z.B. im Zusammenhang mit den ganzen Zahlen behandeln. Faktorisiert man eine Struktur nach einer solchen Kongruenzrelation, so erhält man in der Regel etwas einfacheres. Es gibt seit etwa zwei Jahrzehnten (was in der Mathematik recht wenig ist) eine Theorie der „Gezähmten Kongruenzen“, in der - grob gesagt - erklärt wird, wie man die Menge der Kongruenzrelationen einer Struktur als „Bauplan“ für diese betrachten kann. Hier sind noch viele sehr interessante Fragen zu beantworten. An solchen Fragen arbeite ich auch selbst.

Gibt es in der Algebra nicht auch Forschungsgebiete, die die Informatik direkt betreffen? Wenn man jetzt an Boolesche Funktionen denkt...

Sehr richtig! Boolesche Funktionen spielen ja bei der Beschreibung von Schaltkreisen eine wesentliche Rolle, und nicht nur dort. Daraus ergibt sich die mathematische Frage, wie man solche Booleschen Funktionen beschreibt. Zuerst denkt man, das ist ja ganz einfach: Man schreibt sie einfach in Form einer Tabelle auf. Aber wenn diese Funktionen von vielen Variablen abhängen, und das tun sie beim Schaltkreisentwurf, dann funktioniert das nicht mehr. Bei nur 50 Variablen benötigte eine solche Tabelle schon etwa 100 Terrabyte. Das sprengt jeden Rahmen. Daher sind algebraische Beschreibungen mittels Termen (z.B. aus Konjunktion, Disjunktion, Negation) oder mit Hilfe von OBDDs so wichtig. Auch hier sind riesige Probleme noch offen.

Wie stehen sie als Mathematiker zu der Aussage, dass Mathematik doch mehr eine Hilfswissenschaft für alle anderen Disziplinen sei?

Nun ja... Mathematiker hören das nicht so gerne. Mathematik ist schon eine eigene Wissenschaft mit ureigenen Fragestellungen. Solche innermathematische Fragestellungen haben oft überhaupt keine praktischen Anwendungen im Blick. Aber es pas-

**„Ist die Natur mathematisch? Eigentlich ja.
Warum sonst sollte sie uns so oft den Gefallen
tun mathematisch beschreibbar zu sein.“**

siert auf seltsame Weise immer wieder, dass Ergebnisse eines scheinbar abseitigen mathematischen Gebietes plötzlich für die Praxis wichtig werden. Vielleicht ist die Welt ja zu einem Teil irgendwie mathematisch? Warum uns die Natur den Gefallen tut, in wichtigen Teilen mathematisch beschreibbar zu sein, kann ich auch nicht erklären. Sie könnte ja auch völlig chaotisch sein. Da haben wir eben auch etwas Glück. Andererseits kann man natürlich nicht alles mathematisch beschreiben, das muss auch klar gesagt werden. Aber wir werden immer besser. Schauen Sie zum Beispiel auf die Vorhersagegenauigkeit der Meteorologen. Die hat sich in den letzten Jahrzehnten deutlich erhöht, und das liegt auch an verbesserten mathematisch-physikalischen Wettermodellen und an leistungsfähigeren Computern und Software.

Aber natürlich ist die Mathematik in sehr vielen Fällen auch einfach ein Werkzeug für andere Gebiete, aus denen die Mathematik sich auch umgekehrt Anregungen holt. In den letzten Jahrhunderten war die Paarung Mathe/Physik sehr wichtig, heute ist es die Verbindung zwischen Mathe und Informatik.

Wenn man jetzt an den mathematischen Fortschritt denkt, dann ist das Informatikstudium ja heute schon recht reizvoll. Mehran Sahami, ein Professor für Informatik an der Stanford Universität meinte auch einmal, das sei in etwa so wie wenn man Geometrie in der Zeit von Euklid studiert hätte.

Ich finde es schön, wenn sie das so sehen. Andererseits ist mir schon klar, dass nicht alle Studenten

den Mathematik-Anteil am Studium als so reizvoll empfinden. Doch der allergrößte Teil der Studenten weiß, dass Mathe für die Informatik sehr wichtig ist, und dass man da durch muss. Mathe und Theoretische Informatik sind eben Kernfächer, da kommt keiner dran vorbei. Aber ich versuche immer wieder, den Studenten zu zeigen, dass Mathematik eine Freundin ist, und nicht ein Feind.

Ist es vielleicht manchmal noch ein Hindernis, wenn man als unerfahrener Student eben noch nicht die Intuition besitzt, und man bekommt dann so eine riesige Formel vorgesetzt, und muss dann verstehen, wozu die eigentlich gut sein soll?

Wir versuchen ja gerade, es nicht so zu machen, dass wir den Leuten die Formeln vor die Füße knallen. Sondern wir erklären erst einmal wirklich lange und ausführlich, was das alles bedeutet. Wir versuchen, die Studenten langsam an die exakten mathematischen Schreibweisen zu gewöhnen. Aber irgendwann muss es dann auch einmal „sitzen“. In Mathe II ist es schon wichtig, dass man Mengen, Funktionen usw. richtig aufschreiben und die Schreibweisen ordentlich lesen kann. Und wer damit Probleme hat, der muss mich eben danach fragen.

Auch wenn man heute mit der Mathematik versucht, Klarheit in die Informatik zu bringen, so sind die Teilgebiete ja heutzutage noch immer erschreckend divers. Es gibt zB nicht einmal

eine richtige Definition, was „Cloud-Computing“ wirklich bedeutet.

Das liegt aber daran, dass die Informatik noch eine sehr junge Wissenschaft ist. Mathematik treiben wir ja nun schon seit ca. 2500 Jahren. Die Informatik ist ein paar Dutzend Jahre alt. Ich denke, spätestens in hundert Jahren wird sich das gegeben haben. Dann wird man wissen, was die wirklich wichtigen Grundbegriffe sind. Natürlich ist es heute so, dass viele Leute an verschiedenen Stellen an ähnlichen Themen arbeiten, und jede Schule bringt erstmal ihr eigenes Begriffssystem mit. Das war in der Mathematik ja auch nicht anders. Aber wenn eine gewisse Zeit ins Land gegangen ist, dann wird sich das auch vereinheitlichen und vereinfachen.

...heruntergekocht auf ein paar Axiome?

So simpel wird es auch nicht werden. Die Informatik ist kein Teilgebiet der Mathematik mehr und entwickelt gerade ihre eigene Fachsprache.

Aber wenigstens so, wie man heute ein Algebra-Buch aufschlägt und das schnell und systematisch durchgehen kann?

Wann ist dieser systematische strukturelle Ansatz in der Algebra denn so richtig entstanden? Das war auch erst etwa zu Anfang des letzten Jahrhunderts. Riemann, Felix Klein, van der Waerden, Emmy Nöther und viele andere haben damals begonnen, die modernen Formulierungen der Algebra aufzuschreiben. Das ist also schon eine Weile her. Und es brauchte dann noch Jahrzehnte, bis sich die Begriffe durchsetzten.

In der Informatik dann so etwa... 10 Jahre?

Man darf bei der Informatik auch nicht vergessen dass dahinter eine wahnsinnig schnelle technische Entwicklung steckt.

Es ist eben auch viel Ingenieurwesen dabei.

Ja natürlich. Informatik ist ja zu einem großen Teil eine Ingenieurwissenschaft. Das habe ich zum er-

sten Mal richtig begriffen, als ich am Institut für Informatik gearbeitet habe. Die Gruppe von Professor Gössel beschäftigte sich damals mit dem Aufgabengebiet „Fehlererkennung in Schaltkreisen“. Wie merke ich, wenn ein Schaltkreis nicht mehr richtig funktioniert?

Da geht es also um die Korrektheit der Spezifikationen? Dass die nichts Wichtiges außen vor lassen?

Das ist ein höchst wichtiger Punkt, aber ich dachte jetzt gerade an etwas anderes. Und zwar an die Situation, dass man bereits einen fertigen Schaltkreis hat, der schon eingebaut ist, ein Jahr lang tadellos arbeitet, dann aber an einer kleinen unauffälligen Stelle durchbrennt und man bemerkt es vorerst nicht. Er fängt also an, falsch zu arbeiten. In vielen Fällen mag das nicht so schlimm sein, aber nehmen sie mal an, er steuert ein Flugzeug oder er dosiert in einem medizinischen Gerät die Strahlung zu einer Krebstherapie. Solche wichtigen Schaltkreise benötigen zusätzliche Hardware, die überprüft, ob noch alles in Ordnung ist, und die soll natürlich möglichst klein sein.

Als ich dort angefangen habe, gab es eine Frage zur Kodierung der Zustände in Automaten, mit dem Ziel, möglichst wenige Flip-Flops bei Zustandsänderungen schalten zu müssen. Es zeigte sich, dass die Frage auf ein NP-vollständiges Problem führte. Ich bin dann in das Seminar unserer Arbeitsgruppe gegangen und habe gesagt, dass das Problem zu schwer sei, es sei mindestens NP-vollständig.

Daraufhin grinnten mich alle Kollegen nur an, bis dann einer sagte: „Ja, bei uns sind so ziemlich alle Probleme NP-vollständig, und wir müssen sie trotzdem lösen.“ Da ist mir diese Denkweise erst richtig klargeworden. Die Ingenieure können sich nicht hinstellen und sagen: „Tut mir leid, ist unmöglich wegen irgendeiner Komplexitätsklasse“, sondern sie müssen irgendwie eine praktische Lösung finden.

Vor dieser Denkweise habe ich einen großen Respekt. Ohne dieses Ingenieursdenken gäbe es heute wohl kaum unsere leistungsfähigen kleinen Compu-

ter. Wenn man sich heute einen Laptop kauft, dann ahnt man nicht, wieviel Grips in dessen Entwurf und Herstellung steckt.

Mathematik und Informatik haben sehr enge Beziehungen. Schauen Sie sich z.B. den Begriff des Algorithmus an. Bis zum Computer-Zeitalter war der Eigentum der Mathematiker - ohne dass es eine exakte Definition dafür gab. Die ordentlichen Begriffsbildungen gehen ja erst auf Alan Turing und Kurt Gödel zurück. Aber Algorithmen gibt es schon lange. Denken wir z.B. an den Euklidischen Algorithmus für den größten gemeinsamen Teiler oder an das Gaußsche Eliminationsverfahren. Bereits in der Grundschule lernt man die Algorithmen zur schriftlichen Addition und Multiplikation von natürlichen Zahlen. (Hoffentlich auch noch nach der nächsten Schulreform.)

Heute ist der Begriff des Algorithmus Grundvokabular der Informatiker. Und in der Informatik findet auch die relevante Forschung zu diesem Begriff statt.

Die alten Ägypter haben statt etwas zu Multiplizieren, eine Zahl verdoppelt oder halbiert. Das ist ja auch so eine Art Algorithmus um es einfacher im Kopf zu berechnen.

Auch hier führen viele Wege nach Rom. Natürlich hängt das auch damit zusammen, dass es damals noch kein ordentliches Positionssystem zur Schreibweise von Zahlen gab. Es gab wohl schon eines im alten Sumer, mit der Basiszahl 60. (Daher kommt wahrscheinlich die Tatsache, dass wir den Kreis einteilen in $6 \times 60 = 360$ Grad, und dass die Stunde 60 Minuten hat.) Aber unser Zehner-Positionssystem kennen wir erst seit Ende des Mittelalters.

Woher kommt denn dieses 10er System dann? Aus dem alten Rom?

Naja die Römer haben das 10er System gehabt, aber nicht als Positionssystem. Zahlen wurden dort mit Hilfe von Buchstaben geschrieben. Damit ist wirklich schwer zu rechnen. Multiplizieren Sie doch mal

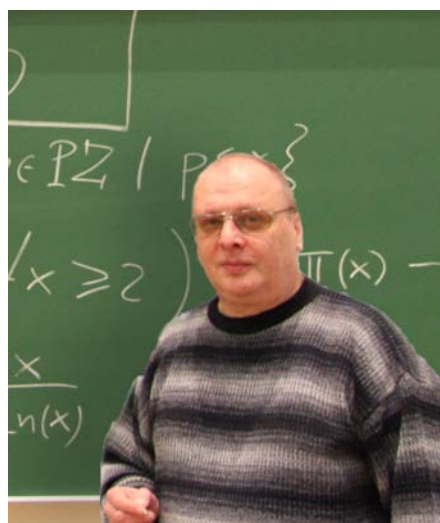
zwei größere römische Zahlen, ohne diese zwischendurch in unser Positionssystem zu bringen. Rechnen war damals eine Kunst.

Unser heutiges System kam wahrscheinlich aus Indien über die Araber nach Europa. Der Name Adam Ries ist heute noch mit der Verbreitung dieser Schreibweise in Deutschland verbunden. Die hohe kulturelle Errungenschaft daran war die Erfindung der Null, so seltsam das auch klingt. Dieses neue Positionssystem hat in Europa vieles verändert, denken Sie zum Beispiel an den Handel und an die Geldwirtschaft.

Tja, und unsere Studenten müssen nun zusätzlich auch noch im Binärsystem arbeiten können. Da können sie den von Ihnen angesprochenen altägyptischen Algorithmus sicher gut verwenden.

Herr Dr. Börner wir danken ihnen für dieses Gespräch.

– Andrina Mascher, Robin Schreiber



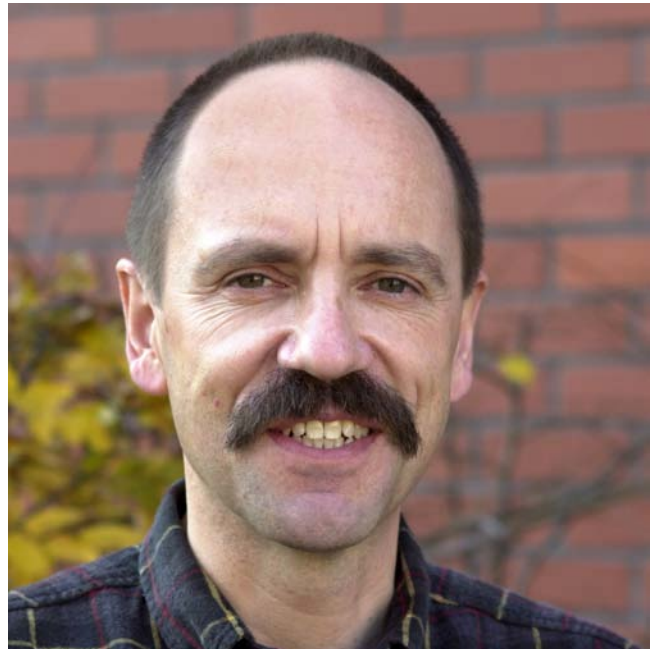
Ferdinand Börner ist Dozent für Mathematik an mehreren Lehreinrichtungen im Raum Berlin-Brandenburg. Am HPI unterrichtet er unter anderem Mathematik II.

Warum Grundlagen lernen?

Viele Studenten der Informatik und verwandter Gebiete haben den Eindruck, dass das Grundlagenstudium und insbesondere die Beschäftigung mit theoretischer Informatik überflüssig sei und dass man die dafür aufgebrauchte Zeit besser für Projekte, praktische Arbeiten und berufsnahen Studienthemmen verwenden sollte. Sie vergessen dabei jedoch, dass ein Studium an einer Universität mehr als nur eine Berufsausbildung ist – dafür wäre ein Studium an einer Fachhochschule oder Berufsakademie wesentlich besser geeignet – und dass gerade die nicht so spezifisch auf eine bestimmte Anwendung orientierten Grundlagenstudien einen in die Lage versetzen, sich schneller an die ständig ändernde Welt der IT anzupassen und grobe Fehler bei der Umsetzung einer Idee in konkrete Software zu vermeiden. Solche Fehler passieren aber schnell, wenn man das Leitziel der Theorie “beweise die Korrektheit der Resultate, die Du erzielst” außer Acht lässt. Ich will im folgenden illustrieren, wie das Missachten der Grundlagen in der Praxis zu gravierenden Fehlern mit erheblichen Folgeschäden geführt hat und warum der gezielte Einsatz mathematisch-theoretischer Hilfsmittel oft zu erheblich besseren Lösungen führt, als die besten Programmierer ohne Verwendung theoretischer Einsichten erzielen können.

Kleine Fehler mit großen Konsequenzen

Wir wissen aus der theoretischen Informatik, dass man die Korrektheit von Software nicht testen kann und nicht darum herum kommt, diese mühsam zu beweisen, wenn es das Einsatzgebiet erforderlich macht. Dennoch glauben auch heute noch viele IT-Praktiker und selbst eine große Zahl von Forschern im Softwarebereich, dass intensives und systematisches Testen eine Garantie für fehlerfreie und zuverlässige Software bieten könne. Es stimmt, dass Testverfahren Fehler in Software aufspüren können und hierfür sind diese Verfahren auch unbedingt erforderlich. Eine Garantie aber können sie nicht liefern, denn im Gegensatz zu physikalischen Objekten, mit denen sich Ingenieure außerhalb des



Christoph Kreitz ist Professor für Theoretische Informatik an der Universität Potsdam und Adjunct Professor am Department of Computer Science der Cornell University.

IT-Bereichs befassen, zeigt Software keinerlei stetiges Verhalten und somit ist eine Interpolation für den Bereich zwischen den Messpunkten nicht zulässig – das Testergebnis für einen Messpunkt sagt überhaupt nichts aus über das Verhalten der Software in der unmittelbaren Nachbarschaft. Genau diese Diskrepanz hat in der Vergangenheit zu einigen groben Fehlern mit kostspieligen und zum Teil fatalen Konsequenzen geführt.

Ich will einige dieser Pannen beispielhaft ansprechen. Bei dem Absturz einer Boeing 767 im Mai 1991 hatte sich als Unfallursache herausgestellt, dass eines der Triebwerke während des Steigfluges auf Schubumkehr geschaltet hatte. Ingenieure des Airbus A320 nahmen dies zum Anlass, eine Sicherheitsschaltung für den Umkehrschub zu programmieren, die verhinderte, dass Umkehrschub eingeschaltet werden konnte, wenn sich die Räder des Landegestells nicht mit mindestens 70km/h drehen. Diese schnelle Lösung hatte fatale Konsequenzen bei der Landung eines Lufthansafluges in Warschau am 14. September 1993. Wegen starken



Der Airbus A320 zeigte, dass auch für „Bugfixes“ umfangreiche Kontrollen nötig sind - ein vermeintlich behobener Fehler kann wiederum neue Fehler enthalten.

Regens führte Aquaplaning auf der Landebahn dazu, dass sich die Räder nur geringfügig drehten und der Pilot die Maschine mangels Umkehrschub nicht rechtzeitig bremsen konnte. Die Maschine schoss über die Landebahn hinaus und ein Passagier und der Copilot kamen ums Leben. Hätte man die Frage „was definiert eigentlich eine Landung“ gründlicher durchdacht, anstatt vorschnell eine Lösung zu programmieren, wäre es nicht zu diesem Unfall gekommen.

Bei der Entwicklung des Pentium I Prozessors im Jahre 1994 wurde erstmalig ein neuer, deutlich schnellerer, Divisionsalgorithmus verwendet. Trotz einer umfänglichen Serie von Tests beobachtete man kurz nach Auslieferung der Chips, dass der Prozessor bei manchen Divisionen falsche Ergebnisse lieferte. Wie sich herausstellte, waren die Hardwaretabellen des Algorithmus unvollständig in den Chipentwurf übertragen worden. Der Rückruf der 2 Millionen Prozessoren kostete die Firma Intel 450 Millionen Dollar.

Beim Jungfernflug der neuen ESA Ariane 501 Rakete im Juni 1996 löste der Steuerungscomputer 37 Sekunden nach dem Start eine Sprengung der Rakete aus, da ein Überlauf in der 16bit Arithmetik der eingesetzten Rechner zu unsinnigen Steuerbefehlen geführt hatte. Wie sich herausstellte, war die Schubkraft der neuen Rakete zu groß für eine Darstellung in 16bit Arithmetik. Nichtsdestotrotz war

die Startsoftware unverändert von der Vorgängerrakete übernommen worden, da die Tests der Software keinerlei Auffälligkeiten zeigten. Der Schaden für die zerstörte Rakete und den beförderten Satelliten betrug mehr als eine Milliarde DM und ein Projekt mit Entwicklungskosten von 11,8 Milliarden DM wurde um mehrere Jahre zurückgeworfen.

Im Jahre 1999 schickte die NASA zwei Sonden zum Mars, zuerst den Mars Climate Orbiter zur Erforschung der Marsatmosphäre und später den Mars Polar Lander zur Erforschung der Marsoberfläche. Der Orbiter sollte den Mars in einer Distanz von 160km umrunden, verschwand aber am 23. September plötzlich von der Bildfläche. Wie sich einige Zeit später herausstellte, hatte die Steuerungssoftware eine Umlaufbahn von unter 60km gewählt, da die Messinstrumente Höhenangaben in Fuß lieferten, die Software aber davon ausging, dass die Angaben metrisch waren. Infolge der niedrigen Umlaufbahn wurde das Steuerungssystem überheizt und die Sonde im Wert von 150 Millionen Dollar stürzte ab. Drei Monate später erreichte der Mars Polar Lander die Marsoberfläche, zerschellte jedoch



Die Ariane 501 explodierte bei ihrem Jungfernflug nach einem Überlauf in der 16-Bit-Arithmetik.

bei der Landung. Wieder waren Einheitenprobleme zwischen Messinstrumenten und Steuerungssoftware die Ursache. Für eine saubere Landung muss die Abschaltung der Bremsraketen 40 Fuß (12 Meter) über dem Boden eingeleitet werden. Die Abschaltung begann jedoch schon bei 40 Metern über dem Boden, so dass die Sonde ungebremst aus 28 Meter Höhe abstürzte - ein Schaden von 165 Millionen Dollar und eine große Blamage für die amerikanische Raumfahrt.

Der ISO/IEC 9796-2 Standard spezifiziert Algorithmen für digitale Signaturen auf der Basis mathematischer Operationen. Der Sinn solcher Standards ist, dass Spezifikationen direkt in sicherheitskritische Anwendungen übertragen werden können. Bei einer Überprüfung der Spezifikation des sogenannten Legendre-Symbols mit einem Theorembeweiser im Jahre 2009 wurde jedoch festgestellt, dass die Spezifikation anstelle des notwendigen Kongruenzsymbols " \equiv " ein Gleichheitssymbol " $=$ " verwendete. Dieser scheinbar unbedeutende Tippfehler hat zur Folge, dass die in kryptographischen Verfahren eingesetzten schnellen Primzahltests fehlschlagen und die Verschlüsselung unsicher wird. Der Fehler bestand seit mehr als 10 Jahren und ist wahrscheinlich ungeprüft in hunderte von Anwendungen übernommen worden. Die Beispiele zeigen, dass die Bandbreite der möglichen Fehler von groben Vereinfachungen zu sehr subtilen Details reicht, die auch mit ausgefeilten Testverfahren kaum identifiziert werden können. Nichtsdestotrotz hätten alle Pannen vermieden werden können, wenn man wenigstens versucht hätte, die Zuverlässigkeit der eingesetzten Soft- und Hardware zu beweisen, anstatt sich nur auf die Ergebnisse der Tests zu verlassen.

Werkzeuge zur Unterstützung von Beweisführung

Nun sind Beweise etwas, mit dem sich Informatiker und Programmierer sehr ungerne beschäftigen. Beweise wirken im Verhältnis zur eigentlichen Entwicklungsaufgabe unproduktiv, sind gerade im Bereich Software wegen der Fülle der Details extrem mühsam und erinnern viele an die unangenehmen Teile ihres Studiums, deren Sinn sie nie so recht einsehen konnten. Aus diesem Grunde richtet sich ein Teil der Grundlagenforschung in der Informatik



Der Mars Polar Lander konnte 1999 wegen Einheitenverwechslung nicht erfolgreich landen – Typechecker hätten das verhindern können.

auf die Entwicklung von logischen Werkzeugen zur rechnergestützten Beweisführung. Die Bandbreite der Tools reicht von Typecheckern in Programmiersprachen, Model-Checkern für Hardware und Theorembeweisern für komplexe logische Probleme bis zu spezifischen Verfahren zur Verifikation, Synthese und fehlerfreien Optimierung von Software. Heute weiß man, dass der Fehler im Pentium Prozessor leicht mit einem Model-Checker hätte gefunden werden können und überprüft deshalb jetzt jeden Hardwareentwurf mit diesem Werkzeug. Der Einsatz von Typecheckern hätte dazu geführt, dass die Einheitenfehler bei den Marssonden und vermutlich auch bei der Ariane-Steuerungssoftware schon bei der Programmierung aufgefallen wären, da man gezwungen wird, sich mit dem Thema Einheiten zu beschäftigen. Der Versuch einer Verifikation der Airbussoftware hätte die Ingenieure dazu bewegt, ihre problematische Definition des Begriff "Flugzeug ist gelandet" offenzulegen.

Auch der Fehler im ISO Standard wurde nur aufgrund der Verwendung eines Theorembeweisers gefunden. Darüber hinaus ist es mit formalen logischen Werkzeugen auch gelungen, systematisch Software aus logischen Spezifikationen zu entwickeln, die nicht nur garantiert korrekt, sondern auch um Größenordnungen effizienter ist als von Hand programmierte Algorithmen. Dies wurde 1993 mit der Synthese von Scheduling Algorithmen für die US Air Force eindrucksvoll unter Beweis ge-

stellt. Mit dem KIDS System konnte innerhalb von wenigen Stunden ein Algorithmus entwickelt werden, der mehr als 2000-mal schneller war als der existierende, in monatelanger Arbeit geschriebene ADA Code. Ein entscheidender Aspekt hierbei war, dass man sich auf die logische Struktur des Problems und konzeptuelle Optimierungen konzentrieren konnte, während die eigentliche Codierung samt Korrektheitsbeweis vom KIDS System erzeugt wurde. Im Bereich Sicherheit sind heutzutage Grundlagen aus Mathematik, theoretischer Informatik und Algorithmen nicht mehr wegzudenken. Alle Verschlüsselungsverfahren, die man bis in die frühen 70er Jahre verwendet hatte, können mit Mitteln der Statistik und der linearen Algebra auf jedem PC innerhalb weniger Sekunden gebrochen werden. Moderne Sicherheitsmechanismen brauchen fundierte Kenntnisse in Gruppentheorie, Zahlentheorie, Informationstheorie und Komplexitätstheorie, um Informationen und Computer erfolgreich gegen Angriffe von außen zu schützen. Auch außerhalb des Themenkomplexes Sicherheit und Zuverlässigkeit gibt es viele Anwendungen, die ohne Grundlagenarbeit nicht zum Erfolg kämen. So spielt die Komplexität von Suchalgorithmen eine entscheidende Rolle für die Effizienz von Graphikkarten und ihren Anwendungen – insbesondere in der Spieleprogrammierung. Ohne fundierte Kenntnisse aus den Grundlagen der Algorithmik und der Komplexitätstheorie kann man heutzutage keine konkurrenzfähigen Graphikanwendungen mehr schaffen.

Relevante Forschung und Lehre an der Universität Potsdam Deswegen befasst sich der Lehrstuhl für Theoretische Informatik am Institut für Informatik mit der Entwicklung logischer Werkzeuge zum Entwurf sicherer und zuverlässiger Software. Unsere Forschungen richten sich auf verschiedene Aspekte:

- Entwicklung und Implementierung effizienter Theorembeweiser für Logiken erster Stufe, insbesondere auch für konstruktive Logik und Modallogiken.
- Verifikation, Synthese, und fehlerfreie Optimierung von Kommunikationssystemen.

Diese Forschungen werden mithilfe eines interaktiven Theorembeweisers in enger Kooperation mit dem Department of Computer Science der Cornell University in Ithaca, NY (USA) durchgeführt. Mit diesem Werkzeug konnte bereits ein subtiler Fehler in einem sorgfältig geprüften Kommunikationsprotokoll aufgespürt werden und die Performanz des implementierten Systems durch logische Optimierungen um den Faktor 10 gesteigert werden.

- Synthese einer vollständig verifizierten elektronischen Chipkarte für US Bundesbehörden. Bei diesem von der NSA geförderten Projekt, das in enger Kooperation mit dem Kestrel Institute in Palo Alto, CA (USA) durchgeführt wird, geht es darum, den gesamten Code der SmartCard – vom Betriebssystem bis zu den kryptographischen Algorithmen – vollständig aus Spezifikationen zu entwickeln und mit einem Theorembeweiser zu verifizieren. Nur so kann das höchste Maß an Sicherheit garantiert werden, was für die vorgesehene Anwendung erforderlich ist.
- Entwicklung theoretischer Grundlagen zur Formalisierung von Informationssicherheit. Dieses noch relativ neue Thema knüpft an den Ende 2009 entstandenen DFG Forschungsschwerpunkt “Reliably Secure Software Systems” an und hat zum Ziel, Theorembeweiser für die Untersuchung von Sicherheitsaspekten nutzbar zu machen. Es ist bereits gelungen, erste Komponenten einer Sicherheitsvorschrift für Banken mit einem unserer Theorembeweiser zu verifizieren.

Die Arbeiten sind gekoppelt mit einem regelmäßigen Lehrveranstaltungsangebot zu Automatisierter Logik und Programmierung, Methoden des automatischen Beweisens, Kryptographie- und Komplexität und einer Reihe von Seminaren zu speziellen Aspekten unserer Forschung.

– Christoph Kreitz

Vehicular Networks

Für die sichere und intelligente Mobilität

Für die Transportsysteme der Zukunft sind Fahrzeugnetze unersetzlich. Das grundsätzliche Prinzip besteht bei dieser Technologie darin, dass Fahrzeuge miteinander und mit Infrastrukturelementen – wie z.B. Ampeln oder Roadside Stations – Informationen per Funk austauschen. Für die Kommunikation wird ein Standard verwendet, der dem WLAN ähnlich ist. Mit Fahrzeugnetzen verfolgt man zwei Ziele: Zum einen soll der Verkehr deutlich sicherer werden, außerdem soll die Verkehrseffizienz erhöht und Staus verhindert werden. Die höhere Sicherheit wird dadurch erreicht, dass Fahrer vor kritischen Situationen gewarnt werden, bevor sie diese selbstständig erkennen können. Hindernisse und Gefahren, wie Eisglätte oder Vorfahrtsverletzungen anderer Fahrzeuge, können somit früher festgestellt werden, was den Spielraum zur Verhinderung von Unfällen stark erhöht. Weiterhin werden Fahrzeugnetze genutzt, um Informationen über den Verkehrsfluss in der Umgebung auszutauschen. Somit können intelligente Navigationssysteme Routenempfehlungen geben, die zur Umfahrung von zähflüssigen Straßenabschnitten genutzt werden können.

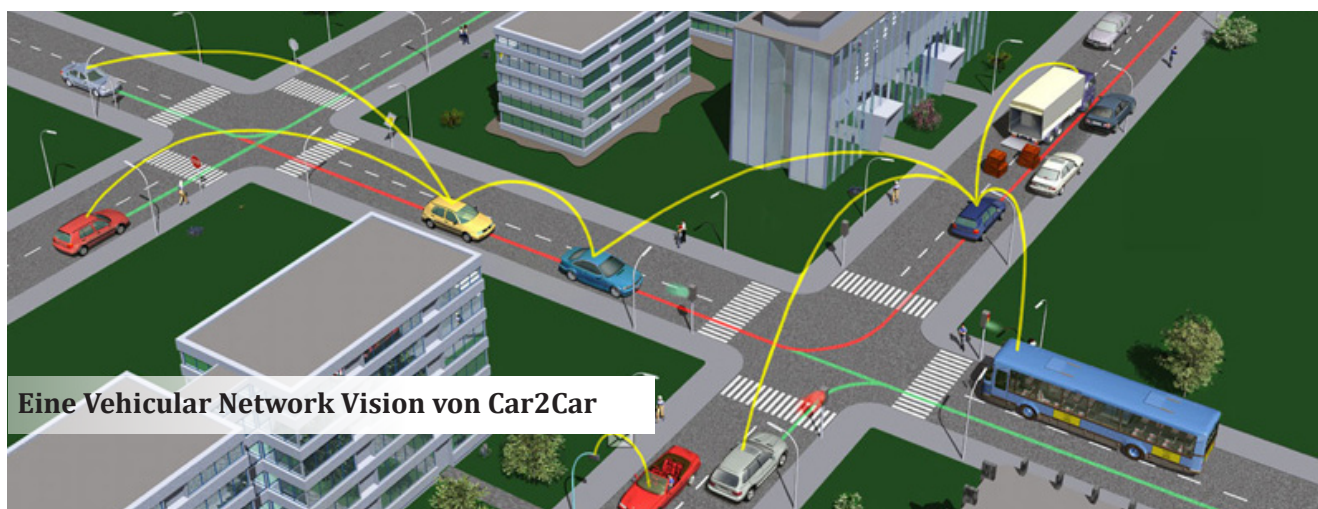
Das Forschungsprojekt simTD bereitet im Moment einen großen Feldtest vor, in dem die Fahrzeugkommunikation unter realen Verkehrsbedingungen getestet werden soll. Gefördert von der Bundesregierung untersuchen hier Automobilher-

steller, Zulieferer und Forschungseinrichtungen gemeinsam die positiven Effekte, die diese Technologie erbringt. Für die Vorbereitung der Feldtests sind umfangreiche Verkehrssimulationen notwendig, bevor die ersten Tests wirklich auf der Straße stattfinden können. Hier kommt die Simulationsarchitektur VSimRTI zum Einsatz, an der HPI-Studenten am Lehrstuhl von Prof. Meinel in einer Kooperation mit dem Daimler Center for Automotive Information Technology Innovations (DCAITI) und Fraunhofer FOKUS mitgearbeitet haben. Weiterhin haben Studenten in einer Lehrveranstaltung am Lehrstuhl von Prof. Meinel das Tool eWorld entwickelt, welches die Vorbereitung von Simulationen deutlich vereinfacht. Das HPI leistet hiermit bei der Entwicklung der innovativen Transportsysteme der Zukunft einen wichtigen Beitrag.

– Björn Schünemann

Links

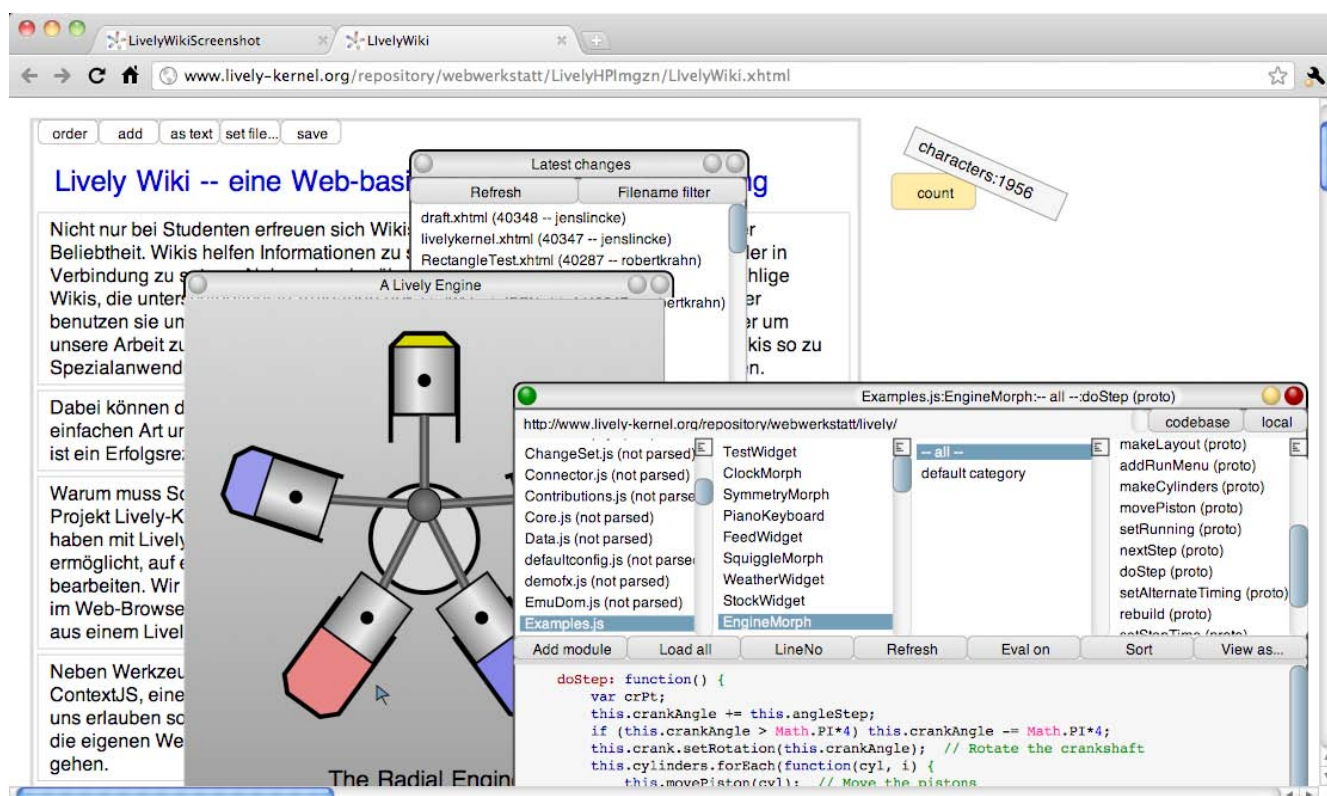
- **Vehicular Networks am HPI:** http://www.hpi-web.de/meinel/projekte/more_projects/vehicular_networks.html
- **VSimRTI:** <http://www.dcaiti.com/research/simulation/>
- **VSimRTI-Video:** <http://www.youtube.com/watch?v=sv2GWsMIfNk>
- **eWorld:** <http://eworld.sourceforge.net/>
- **simTD:** <http://www.simtd.de>



Eine Vehicular Network Vision von Car2Car

Lively Wiki

Webbasierte Entwicklungsumgebung



Nicht nur bei Studenten erfreuen sich Wikis seit ein paar Jahren stetig wachsender Beliebtheit. Wikis helfen Informationen zu sammeln, zu strukturieren und miteinander in Verbindung zu setzen. Neben dem berühmtesten Wiki, der Wikipedia, gibt es unzählige Wikis, die unterschiedlichste Aufgaben übernommen haben. Wir Softwareentwickler benutzen sie, um unsere Projekte zu dokumentieren, Anforderungen festzuhalten, oder um unsere Arbeit zu koordinieren. Durch Konventionen und Erweiterungen können Wikis so zu Spezialanwendungen werden, ohne dass diese extra programmiert werden müssen.

Dabei können die Nutzer in den meisten Wikis nur in einer wenig benutzerfreundlichen aber einfachen Art und Weise strukturierten Hypertext bearbeiten. Aber gerade diese Einfachheit ist ein Erfolgsrezept dieser Technologie.

Warum muss Softwareentwicklung schwerer sein als eine Wiki-Seite zu bearbeiten? Im Projekt Lively-Kernel beschäftigen wir uns mit solchen und damit verbundenen Fragen. Wir haben mit Lively-

Wiki eine webbasierte Entwicklungsumgebung geschaffen, die es uns ermöglicht, auf einer Wiki-Seite nicht nur Text und Grafiken in WYSIWYG-Manner zu bearbeiten. Wir können das Verhalten dieser grafischen Objekte ändern und aus ihnen direkt im Web-Browser Anwendungen bauen. Dabei entwickeln wir die Entwicklungsumgebung aus einem Lively-Wiki heraus.

Neben Werkzeugen und der Infrastruktur forschen wir an Spracherweiterungen wie ContextJS, einer kontext-orientierten Programmiersprach-Erweiterung für JavaScript, die es uns erlauben, solche sich selbst unterstützenden Systeme zu entwickeln, ohne dass dabei die eigenen Werkzeuge beim Programmieren wichtiger Systemteile versehentlich beschädigt werden.

Der Lively-Kernel (<http://www.lively-kernel.org/>) wird am Fachgebiet Software-Architekturen von Robert Krahn und Jens Lincke in enger Zusammenarbeit mit Dan Ingalls (SAP) entwickelt.

— Jens Lincke

Automatisierte Software-Analyse und Visualisierung

Vom Programmierer bis zum Projektleiter: Wer an der Entwicklung komplexer Softwaresysteme beteiligt ist, verbringt einen Gutteil seiner Zeit damit, Systemstrukturen und Entwicklungsvorgänge zu ermitteln, zu analysieren und zu verstehen. In der Anfangsphase erweisen sich Softwareprojekte im Allgemeinen als überschaubar, vor allem wenn wenige Entwickler beteiligt sind, die meist die Spezifikationen selbst erstellt haben und sich das Wenige, was es zu dokumentieren gibt, merken können. Mit der Zeit wird die Sache schwieriger: Das Team wächst, der Quellcode ebenfalls, es treten die ersten undokumentierten Seiteneffekte auf und irgendwann müssen auch erfahrene Entwickler zugeben, die Software nicht mehr vollständig zu durchschauen. Eine entscheidende Frage in dieser Situation lautet: Wie kann so schnell wie möglich ein verlässlicher, objektiver Überblick über den Zustand des Softwaresystems und -projekts erlangt werden, um z. B. Entwicklungsverzögerungen, hohe Fehlerbehebungskosten oder gar das Scheitern des Projektes zu vermeiden?

Dieses zentrale Problem für die professionelle, nachhaltige Entwicklung und Fortführung von Softwaresystemen stellt einen Ausgangspunkt für die Konzepte und Verfahren zur automatischen Software-Analyse und Visualisierung dar, die am Fachgebiet Computergrafische Systeme von Prof. Dr. Döllner und seinem Team erforscht werden. Die Grundidee für die Lösungsansätze ist dabei, dass automatisiert Informationen über das jeweilige Softwaresystem und seines Entwicklungsprozesses extrahiert, zusammengeführt, aufbereitet, visualisiert und damit den am Softwareentwicklungsprozess Beteiligten verständlich zugänglich gemacht werden.

Konzepte für automatisierte Software-Analyse und Visualisierung

Grundlage der Konzepte und Verfahren bilden alle Implementierungsartefakte, wie sie in jeder Softwareentwicklung vorliegen, d.h. im einfachsten Fall Quellcodes, Executables und Source-Code-Repositories. Als weitere Quellen können zudem Bug-Trackingsysteme, allgemein Issue-Trackingsysteme, Systeme zur Anforderungsverwaltung einbezogen werden. Aus diesen Quellen lassen sich auf unterschiedliche Weise im ersten Schritt (Abb. 1) Informationen extrahieren, die Auskunft über die Strukturen und das Laufzeit-Verhalten des Softwaresystems sowie über dessen entwicklungsbezogene Evolution geben. Die gewonnenen Daten werden dann in einer Datenbank verwaltet. Auf den Daten selbst können z.B. Algorithmen zur Musteranalyse operieren, die Sekundärinformationen ableiten.

Visualisierung spielt bei der Software-Analyse eine entscheidende Rolle. Ein prominentes Beispiel ist Debugging: Es ist eine Art Detektivarbeit, bei der ein Entwickler sich nach und nach an die Fehlerursache im Quellcode herantastet. Eine ursprüngliche grobe Hypothese über die Fehlerursache wird durch Exploration des Systemverhaltens verifiziert oder falsifiziert. Anschließend wird die Hypothese verfeinert und die Suche/Exploration geht weiter. Dieser zyklische und iterative Debugging-Prozess kann mit spezialisierten Visualisierungstechniken vereinfacht werden, die den Entwickler durch visuell-analytische Verfahren bei der Verifikation/Falsifikation seiner Hypothesen unterstützen. Die folgenden Abschnitte geben einen Einblick in einige unserer Forschungsarbeiten. Zuerst werden wir an einem komplexen Softwaresystem skizzieren, wie mit „Software Maps“ Qualitätsschwachstellen in der Systemimplementierung identifiziert werden können. Anschließend werden verschiedene Techniken und Anwendungsfelder für die Visualisierung des Laufzeitverhaltens von komplexen Softwaresystemen („Trace-Visualisierung“) vorgestellt.

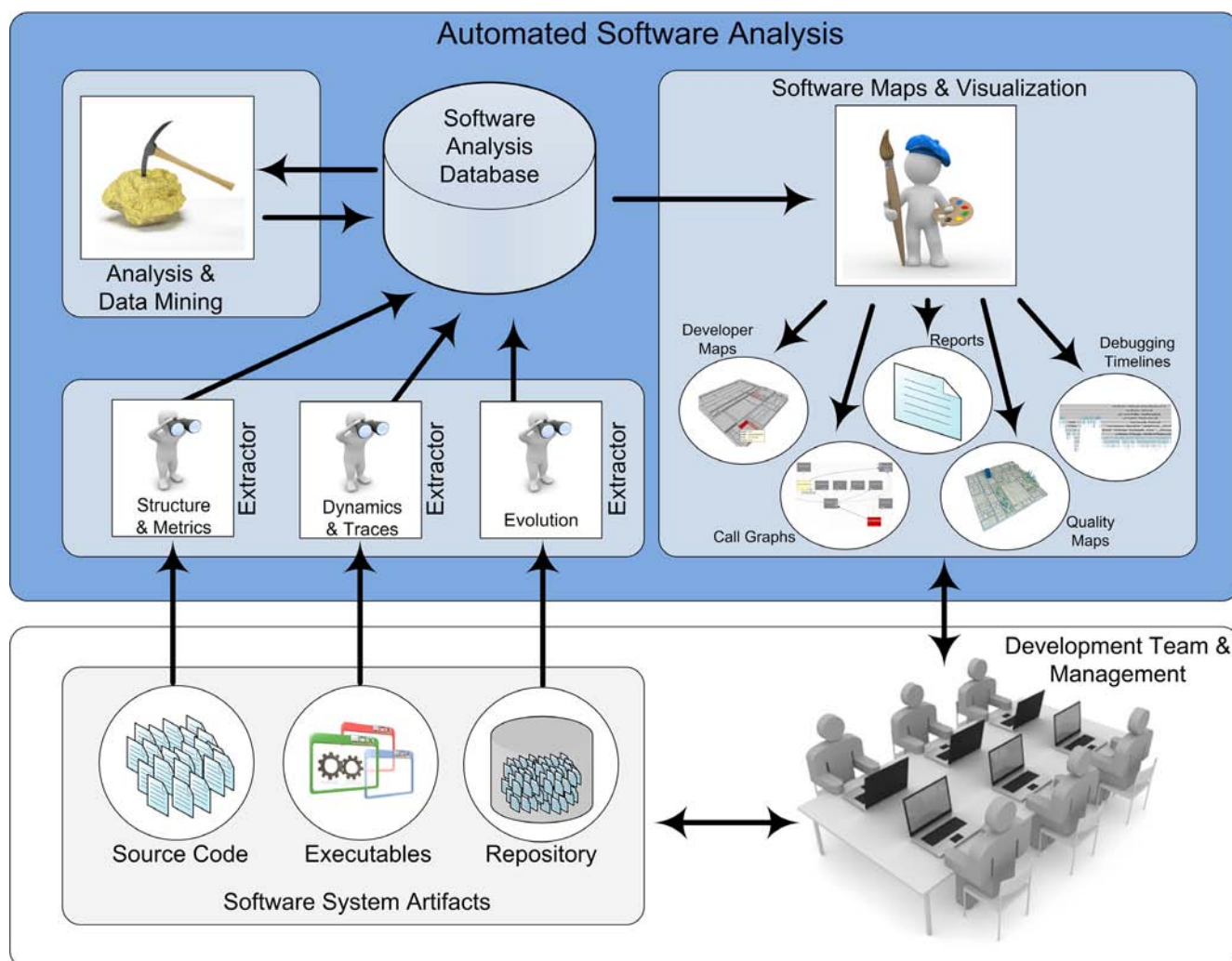


Abbildung 1: Prinzipieller Ablauf der automatisierten Software-Analyse.

Software Maps als „Frühwarninstrumente“ für Qualitätsrisiken

Unter dem Begriff Software Maps fassen wir, grob gesprochen, Darstellungen zusammen, die Struktur- und Artefaktinformationen über ein Softwaresystem grafisch repräsentieren. Visuelle bzw. grafische Attribute der Darstellung ermöglichen es, weitere Informationen, die mit den Strukturen bzw. Artefakten in Verbindung stehen, zu visualisieren, wie z. B. Ergebnisse von Software-Metriken zu Code Complexity, Coupling/Cohesion, Code Coverage, Execution Time, etc.

In Abb. 2 ist eine Software Map zu sehen, die als eine Art Frühwarninstrument für Projektmanager agiert. Beim gezeigten Softwaresystem handelt es sich um „Blender“, ein in C/C++ implementiertes System für 3D-Modellierung mit ca. 500.000 Lines-of-Code (LOC). Jeder Block in der Darstellung entspricht ei-

ner Quellcodedatei. Die Dateien sind entsprechend ihrer Enthaltenseins-Beziehung in der Verzeichnisstruktur in der Darstellung räumlich gruppiert und hierarchisch als sogenannte Treemap angeordnet, sodass die Modulstruktur des Systems sichtbar wird. Die Software Map visualisiert zugleich Metrik-Informationen, die z. B. Entwicklungsaktivitäten oder Qualitätscharakteristiken widerspiegeln. In Abb. 2 wird folgendes verwendet:

- Die Grundfläche der Blöcke gibt proportional die Anzahl der Statements (ausführbare Codezeilen, ohne Kommentare) wieder – ein Indikator für „schwierige Wartbarkeit“, denn monolithisch gewachsene Quellcodedateien weisen meist keine klare innere Strukturierung auf.
- Die Höhe der Blöcke kodiert die Anzahl der Änderungen an den jeweiligen Dateien – ein Indikator für kritische Entwicklung, da eine

Datei, die mit hoher Frequenz verändert wird, mit großer Wahrscheinlichkeit keinen klar abgrenzten Funktions- und Verantwortlichkeitsbereich hat.

- Die Farbe kodiert, wie viele unterschiedliche Entwickler an Änderungen an einer Datei beteiligt waren. Je mehr unterschiedliche Entwickler dies sind, desto höher ist auch hier die Wahrscheinlichkeit, dass der Verantwortlichkeitsbereich der Datei nicht klar definiert ist.

Diese kombinierten drei Qualitätsindikatoren ermöglichen es beispielsweise Projektmanagern, sich schnell einen Überblick über laufende Entwicklungsaktivitäten zu verschaffen und frühzeitig z. B. zu erkennen, ob die Entwickler gerade in schwer wartbarem Code aktiv sind (breite, rote und hohe „Türme“). Zur Qualitätsanalyse können selbstverständlich noch weitere Metriken herangezogen werden. Als Beispiele wären zu nennen: McCabe Komplexität, Verschachtelungstiefe von Kontrollflussanweisungen im Code, Fan-In-/Out Metriken, Testüberdeckung u.a.

Insgesamt kann mit einer interaktiven Software Map jederzeit vollautomatisch ein genaues „Bild“

sowohl von Qualitätsaspekten des Softwaresystems als auch der Entwicklungsaktivität ermittelt werden. Mit ihnen kann z. B. besser entschieden werden, wann es sich lohnt, Softwarequalitätsmaßnahmen vor neu zu entwickelnden Features den Vorzug zu lassen. Im Beispiel erkennt man auf einen Blick, dass es sich z.B. lohnt, die „Ausreißer“-Datei `readfile.c` zu refaktorisieren. Denn mit ihren knapp 9.000 Statements ist sie relativ groß und wird zudem mehrfach pro Woche von den Entwicklern verändert. Vermutlich ist sie eine „Entwicklungsbremse“ und birgt zudem ein hohes Risiko, dass Fehler bei Codeänderungen in das System gelangen.

Trace-Visualisierung als Technik zum Verstehen von Laufzeitvorgängen

Softwareentwickler benötigen für praktisch jede Arbeit an Softwaresystemen (z. B. Systemerweiterung, Fehlerbehebung, Portierung, Maintenance) ein grundlegendes Verständnis über das Laufzeitverhalten des Systems. Es ist jedoch eine zeitintensive Aufgabe, ein solches Verständnis zu erlangen, insbesondere mit steigender Zahl der Entwickler und Entwicklungsjahre. Mit herkömmlichen Werk-

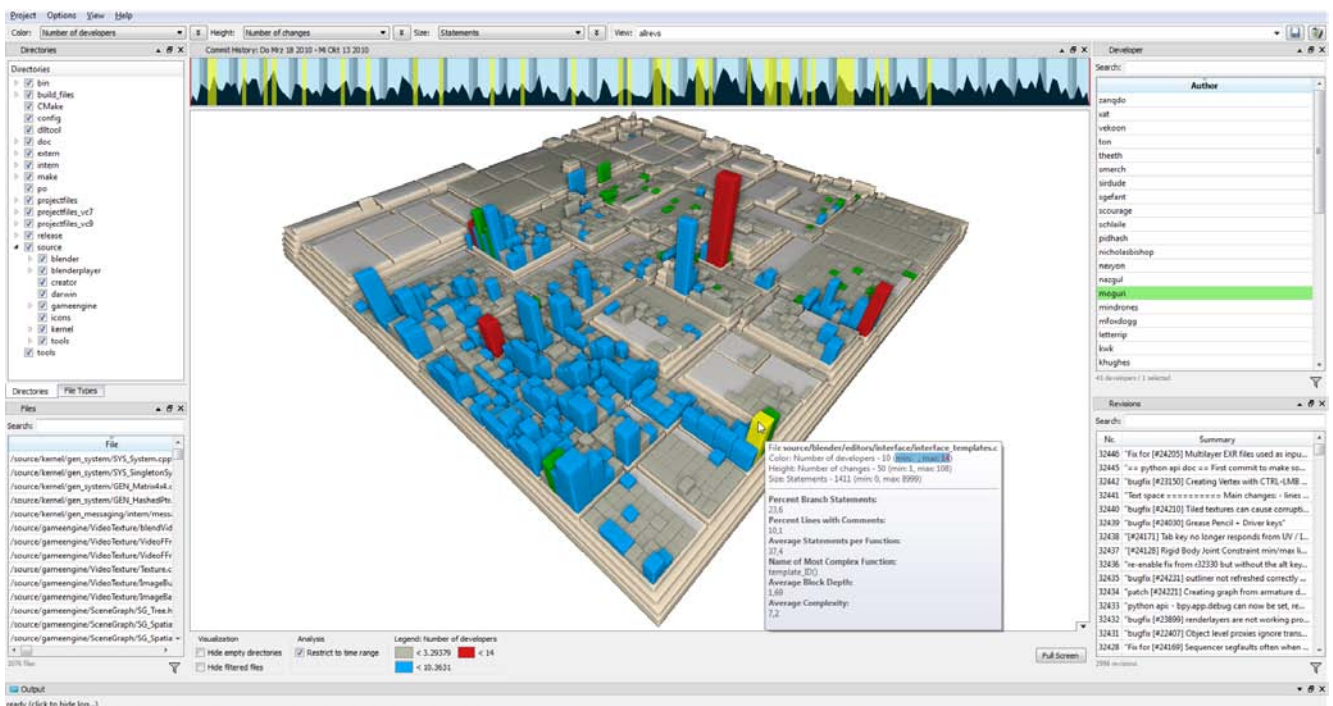


Abbildung 2: Hotspots der Entwicklung im „Blender“-System (500.000 LOC).

zeugen wie Debuggern und Entwicklungsumgebungen ist es schwierig, sich ein Bild des Systemverhaltens zu erarbeiten, da diese Werkzeuge zwar konkrete Informationen über den aktuellen Systemzustand anzeigen, aber nur beschränkt Informationen über die in der Vergangenheit aufgetretene Ereignisse liefern. Während Wartungsarbeiten ist es jedoch häufig erforderlich zu erfahren, wie vorangegangene Ereignisse zum aktuellen Systemzustand führten, da aus dem aktuellen Systemzustand nur unzureichend auf zuvor aufgetretene Ereignisse geschlossen werden kann. Tracing bezeichnet Techniken, die Laufzeitereignisse chronologisch protokollieren; sie beruhen im Allgemeinen auf der Instrumentierung des ausführbaren Systems. Tracing liefert nicht nur punktuelle Informationen über das Systemverhalten, sondern ermöglicht es auch, die Historie eines Programmablaufs insgesamt zu erfassen. Ein Hauptproblem bei Tracing ist die im Allgemeinen sehr große Zahl von protokollierten Laufzeitereignisdaten und damit die Datenmenge, z.B. können schon innerhalb weniger Sekunden mehrere Millionen Funktionsaufrufe aufgezeichnet werden, in Verbindung mit der Forderung, den eigentlichen Systemablauf und die Performance soweit wie möglich nicht zu beeinflussen. Das Fachgebiet Computergrafische Systeme erforscht verschiedene Ansätze für das Tracing mit effizienten Instrumentierungstechniken. Um große Mengen von Trace-Daten auszuwerten und interpretieren zu können, werden zudem spezielle Verfahren zur Visualisierung von Trace-Daten entwickelt. Erst die Visualisierung ermöglicht es, "Einblicke" in die komplexen, massiven Trace-Daten zu erhalten. Die am Fachgebiet entwickelten Aufzeichnungs- und Visualisierungstechniken für Trace-Daten unterstützen momentan C/C++, Java und .Net Softwaresysteme.

Abbildung 3 zeigt beispielhaft Visualisierungen von Trace-Daten, die während der Laufzeit des Google Chrome Webbrowsers (ca. 4 Millionen LOC in C/C++) aufgenommen wurde. Konkret wurde hier mit Trace-Visualisierung untersucht, welche benutzerbezogenen Daten der Webbrowser während dessen Nutzung sammelt und an zentrale Google-Server verschickt. Im Folgenden wollen wir

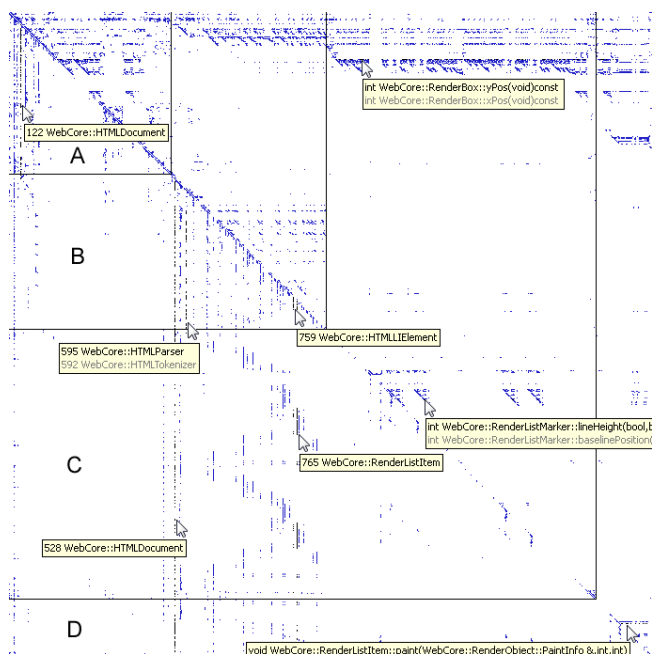


Abbildung 4: Visualisierung von Objektverhaltensmuster

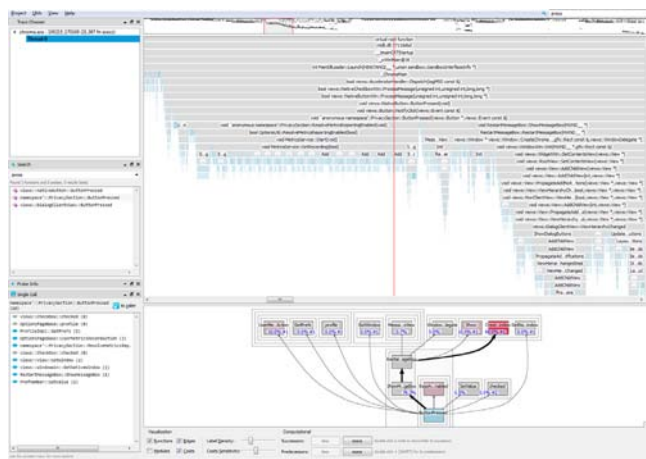


Abbildung 3: Visualisierungen des Laufzeitverhaltens des Google Chrome Webbrowsers.

einige Forschungsschwerpunkte am Fachgebiet im Bereich Trace-Visualisierung vorstellen:

- Trace-Visualisierung von objektorientierten Softwaresystemen
- Trace-Visualisierung von Multithreaded Softwaresystemen
- Trace-Visualisierung von Verteilten Softwaresystemen
- Trace-Visualisierung von Eingebetteten Softwaresystemen
- Trace-Visualisierung von Softwaresystemen in Produktivumgebungen

Trace-Visualisierung von objektorientierten Softwaresystemen

Bei der Entwicklung und Wartung objektorientierter Softwaresysteme kommt es darauf an, das Verhalten von Objekten zu verstehen – dies umfasst sowohl deren Lebenszyklen als auch deren Inter-Objekt-Kommunikation. Wir erforschen daher, wie Erzeugungs-, Zerstörungs- und Kommunikationsmuster von Objekten analysiert und visualisiert werden können.

Mit Hilfe der von uns entwickelten Techniken können Softwareentwickler typische Lebenszyklen von Objekten derselben Klasse verstehen und Ausreißer erkennen. Sie erhalten Einblicke darin, wie Objekte miteinander kommunizieren und welche Objekte z. B. lange leben und somit Teile des Systemzustandes über verschiedene Programmausführungsphasen hinweg transportieren. Abb. 4 zeigt eine Visualisierungstechnik, die auf die Darstellung Objektverhaltensmuster spezialisiert ist.

Trace-Visualisierung von Multithreaded Softwaresystemen

Nebenläufige Systeme zu verstehen ist oftmals viel schwieriger als Programme, dessen Funktionalität auf der sequentiellen Abarbeitung durch einen Thread basiert. Bei Multithreaded Systemen kann es nicht nur mehrere, parallele Aufrufe von Methoden geben; darüber hinaus müssen noch Thread-Erzeugungen, -Zerstörungen und -Synchronisationsstellen

berücksichtigt werden. Softwareentwickler müssen zudem Fehlerquellen, die erst durch Nebenläufigkeit möglich werden, vermeiden: Race Conditions, Deadlocks, Live Locks, ungewollt überschriebene Daten, um nur einige zu nennen. Konventionelle Debugger bieten oft nur unzureichende Unterstützung für die Entwicklung und Wartung von Multithreaded Systemen: der Kontext eines Threads, die Kollaboration mit anderen Threads zur Laufzeit oder auch die Historie von vorhergehenden Ereignissen, ist oft nicht verfügbar oder nur aufwendig manuell rekonstruierbar. Darüber hinaus gibt es Fehlerkonstellationen, in denen sich ein teilweises Anhalten des Systems durch Debugger verbietet, da dies zu Zuständen führen kann, die in der Realität nicht auftreten und so bei der Fehlersuche irreführend wären.

Visualisierung von Verteilten Softwaresystemen

Ein weiterer Forschungsschwerpunkt bildet die Analyse und Visualisierung verteilter Softwaresysteme – z. B. Systeme, die auf Service-orientierten Architekturen basieren. Hierbei ist es u. a. notwendig, auf entfernten Rechnern unter verschiedenen Systemplattformen Trace-Daten zu erfassen, diese zentral zu sammeln und schließlich zu visualisieren. Zusätzlich zum Systemverhalten innerhalb der einzelnen beteiligten Softwarekomponenten sind die zwischen den Einzelsystemen ausgetauschten Nachrichten von besonderem Interesse. Für die Vi-

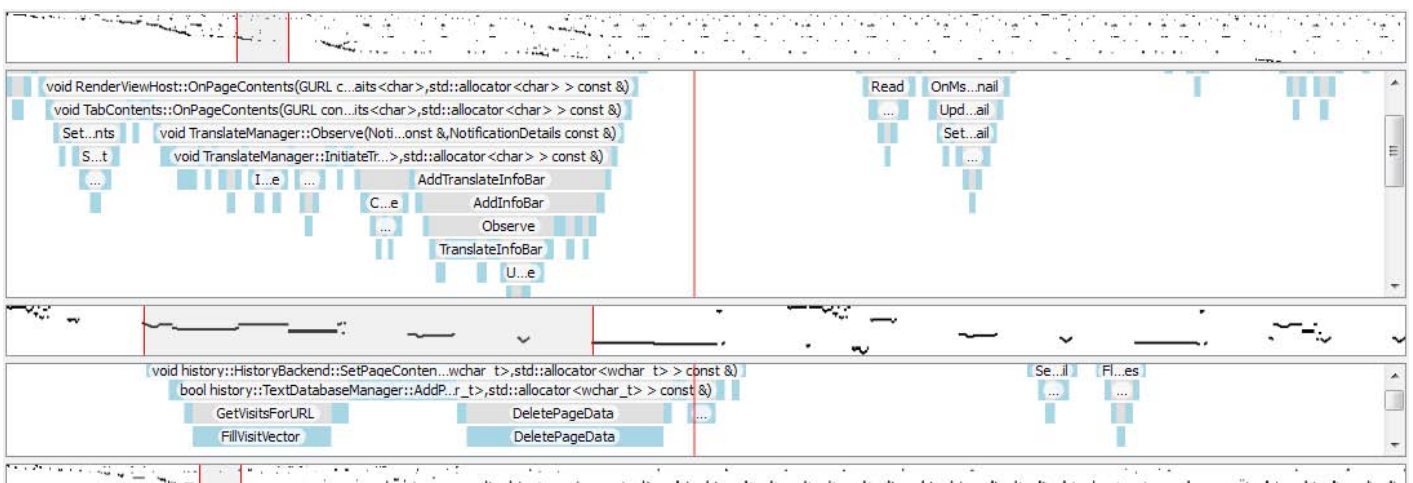


Abbildung 5: Visualisierung des Zeitverhaltens zweier nebenläufig ausgeführter Threads des Google Chrome Webbrowsers

sualisierung von Trace-Daten verteilter Systeme entwickeln wir Werkzeuge, die die interaktive und visuelle Analyse des Laufzeitverhaltens des Gesamtsystems unterstützen. Dabei müssen große, hierarchische Datenmengen unterstützt werden, die mit strukturellen oder zeitorientierten, gerichteten Relationen verknüpft sind. Dies können beispielsweise die erwähnten Nachrichtenflüsse oder Abhängigkeiten in verteilten Softwaresystemen sein. Abb. 7 zeigt ein Beispiel hierfür.

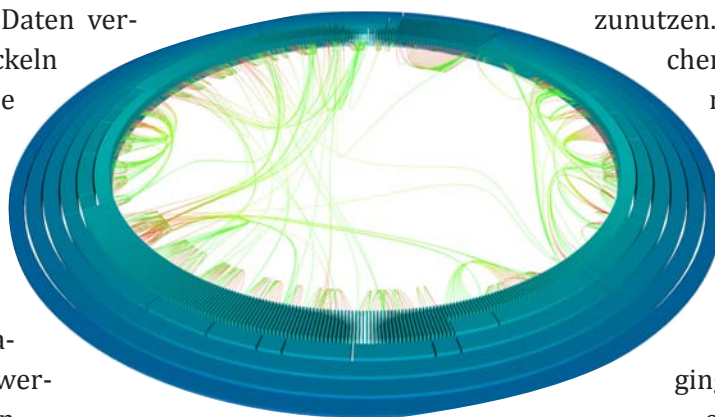


Abbildung 7: Visualisierung der Abhängigkeiten zwischen Elementen der Modulhierarchie eines 500.000 LOC Softwaresystems

Die Herausforderungen bei der Visualisierung liegen u. a. darin, unzählige, sich überlagernde Nachrichten klar darzustellen und visuelle Vergleiche verschiedener Trace-Daten zu ermöglichen. Dabei untersuchen wir derzeit, wie Techniken anderer Visualisierungsbereiche auf die Softwarevisualisierung übertragen werden können (z. B. Flow Visualisation).

Trace-Visualisierung von Eingebetteten Softwaresystemen

Bei Eingebetteten Systemen gilt es, die vorhandenen beschränkten Ressourcen möglichst gut aus-

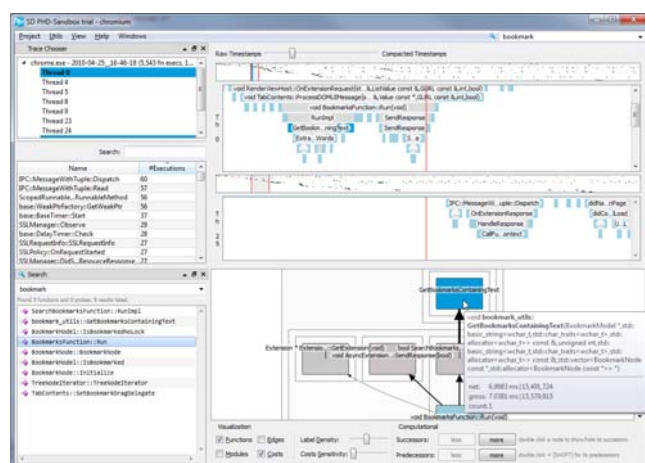


Abbildung 6: Visualisierung des Zeitverhaltens zweier nebenläufigen Google Chrome Threads und dazugehörige Funktionsaufrufgraphen.

zunutzen. Die Fehlersuche in solchen Systemen erweist sich noch immer als zeitaufwendige und komplizierte Aufgabe. Obwohl es im Bereich Eingebetteter Systeme separate Prozessoren mit Debugging-Schnittstellen gibt, ist es nicht immer sinnvoll, diese zu nutzen, da sie typischerweise die aufwendige Erstellung eines weiteren Leiterplatten-

layouts erforderlich machen. In der Regel müssen Entwickler daher mit derselben Hardware auskommen, die auch an Kunden verkauft wird und aus Kostengründen entsprechend nur mit absolut notwendigen Ressourcen ausgestattet ist. Auch ist eine simulierte Ausführung des Systems nur dann möglich, wenn alle externen Schnittstellen ebenfalls simuliert werden können. Zudem gilt es oft Antworten auf Fragestellungen aus dem Multithreading zu finden, in denen ein teilweises Anhalten des Systems die Fehlersuche erheblich erschwert. Gerade in solchen Situationen ist Trace-Visualisierung eine sinnvolle Technik – zusätzlich zum herkömmlichen Debugging. Wir entwickeln aktuell eine Technik zur Trace-Generierung und -Visualisierung für eingebettete Softwaresysteme.

Trace-Visualisierung von Softwaresystemen in Produktivumgebungen

Treten Fehler in der Produktiv- und Einsatzumgebung beim Kunden auf, ist es oftmals schwer, diese Fehler nachzuvollziehen: Ein mühseliger Kommunikationsprozess zwischen Kunde und Entwicklungsabteilung beginnt, der neben den entstehenden Kosten zudem die Kundenbeziehung belasten kann. Techniken wie Remote Debugging des Systems in der Produktivumgebung können nur in seltenen Fällen genutzt werden. Bei hochverfügbaren Systemen, wie z. B. Webservern, ist es zudem undenkbar, dass das System zu Debugging-Zwecken angehalten wird. Traces, die auf dem Produktivsystem im Falle eines Systemfehlverhaltens erzeugt

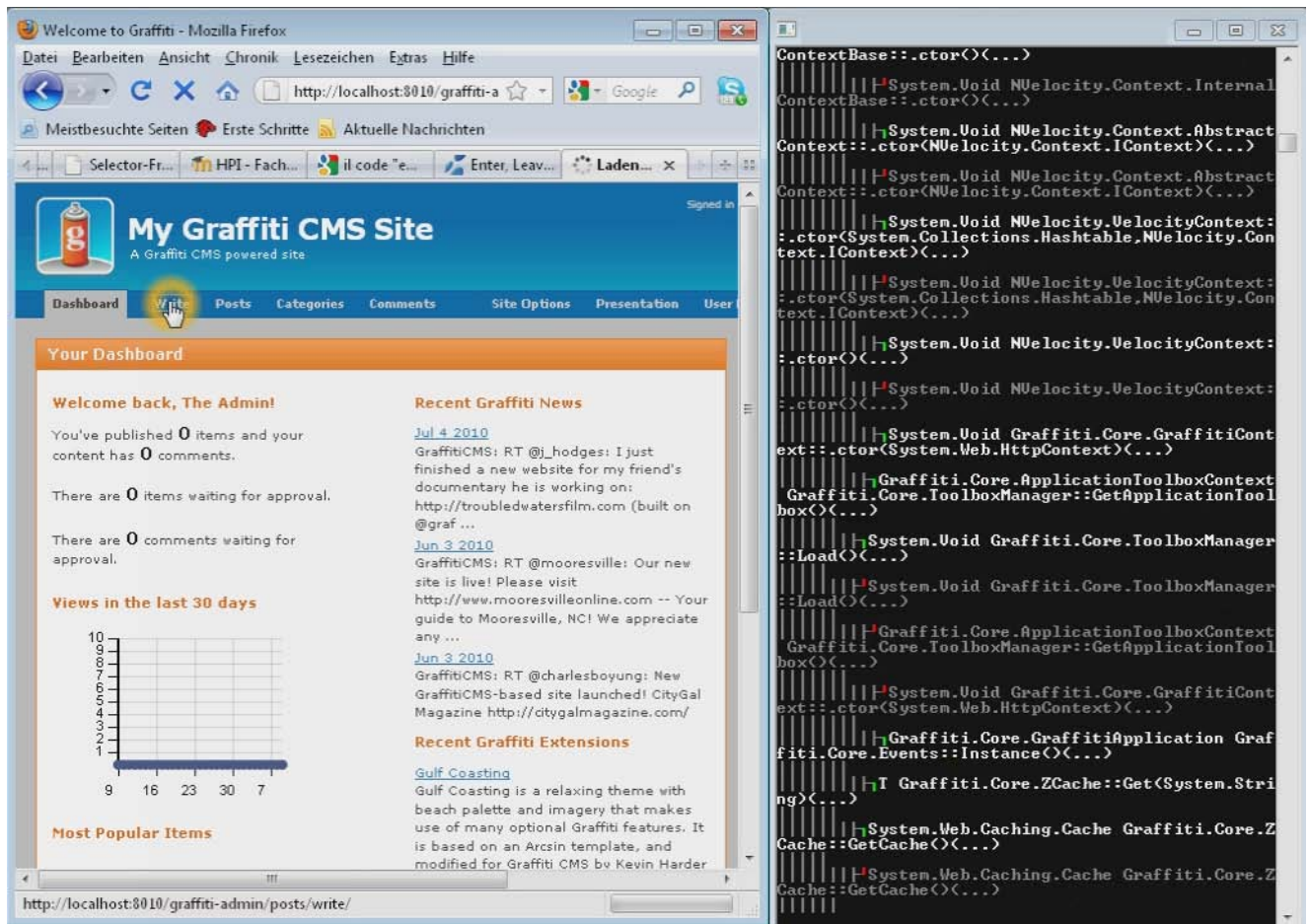


Abbildung 8: Prototyp für das Tracing eines ASP.NET Systems. Links: Web-Frontend von CMS Graffiti. Rechts: Methodenaufwurf-Sequenz während Benutzerinteraktionen mit dem Web-Frontend.

werden, können die Entwickler bei der Fehlersuche unterstützen (Abb. 8). Wir entwickeln dazu Tracing-Lösungen, die es ermöglichen, Tracing-Funktionalität automatisiert in ausführbare Programme zu integrieren. Damit können C/C++-Systeme, Java-Systeme (z. B. Websphere Application Server oder Tomcat Web Server) oder .Net-basierte Systeme (z. B. ASP.Net Web Server) "im Betrieb" analysiert und visualisiert werden.

Schlussbemerkung

Dieser Beitrag gab einen kleinen Einblick in die automatisierte Software-Analyse und Visualisierung. Es ist ein relativ junges, wenig erforschtes Gebiet im IT-Systems Engineering. Die entsprechenden Entwicklungswerkzeuge gewinnen zunehmend an Relevanz, da die effiziente, langfristige und nachhaltige Entwicklung komplexer IT-Systeme eine grundlegende Aufgabe in Wirtschaft und Gesellschaft darstellt. Studierende können dieses Themenfeld im

Fachgebiet Computergrafische Systeme im Bachelor- und Masterstudium in Form von Projekten, Seminaren und Vorlesungen kennenlernen. Zusammen mit Herstellern von Softwareentwicklungswerkzeugen besteht außerdem die Möglichkeit, in konkreten Entwicklungsaufgaben mitzuarbeiten. Auch für Promotionsthemen bietet Software-Analyse und Visualisierung eine Fülle von Themen.

— Johannes Bohnet, Jonas Trümper, Stefan Voigt,
Martin Beck, Jürgen Döllner

Vom Roten Kreuz zu rotem Backstein

Die Geschichte des Campus Griebnitzsee

Die meisten Menschen mögen mit dem Begriff „Griebnitzsee“ höchstens einen S-Bahnhof oder einen kleinen Bestandteil der südwestlichen Berliner Seenlandschaft assoziieren. Für die circa 450 Studenten des HPI ist er jedoch gar nicht mehr aus dem Wortschatz wegzudenken. Täglich strömen sie zum Campus Griebnitzsee, verbringen in den drei Gebäuden rund um den Lake HPI viele lehrreiche Stunden und verlassen das Gelände abends mit dem guten Gefühl, ein Stück zur Erhöhung des geistigen Kapitals dieser Welt beigetragen zu haben. Im Jahre 2000 wurde der erste Stein für die roten Backsteingebäude unseres heutigen Instituts gelegt, und wir alle erinnern uns an die Festlichkeiten des 10-jährigen Bestehens. Doch kaum einer weiß, was sich auf dem Gelände unseres heutigen Campus bis dahin zugetragen hat.

Leben kam in das bewaldete Gebiet am Ufer des Griebnitzsees zum ersten Mal vor circa 135 Jahren mit der Entstehung der Villensiedlung Neubabelsberg, die sich damals nördlich des heutigen Bahnhofs befand, der zur gleichen Zeit entstand.

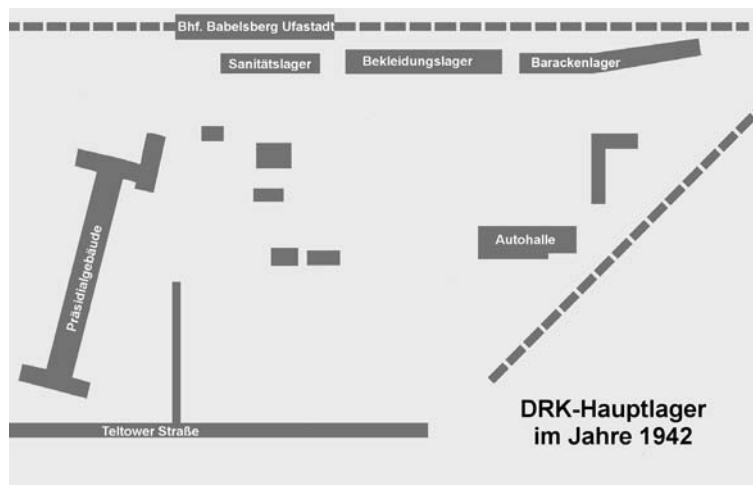
Bereits einige Jahre später beschloss man, das Gelände südlich der Bahnstrecke, auf dem sich heute der Campus Griebnitzsee befindet, aufgrund seiner durch den Bahnhof günstigen Verkehrslage auszunutzen. 1896 errichtete der Preußische Landesverein vom Roten Kreuz hier Lagerhallen, die als Depot für mobile Lazarettbaracken dienten. Bei einem simplen Lager sollte es jedoch nicht bleiben. Die Einrichtung wurde kontinuierlich ausgebaut, so dass es schließlich zum Zentraldepot mit mehreren

Wohn-, Verwaltungs- und Nutzungsgebäuden wurde.

Während das Deutsche Rote Kreuz ursprünglich auf Friedenstätigkeit orientiert war und sich politisch neutral verhielt, sollte sich das mit der Macht ergreifung Hitlers ändern. Die Organisation wurde nach und nach gleichgeschaltet und in den nationalsozialistischen Staat integriert. Adolf Hitler wurde bereits 1934 zum Schirmherrn des DRK ernannt und übertrug daraufhin 1936 dem SS-Führer Dr. Ernst-Robert Grawitz, der als „Reichsarzt der SS“ auch für die Konzentrationslager zuständig war, die Leitung des DRK. Grawitz sorgte für eine Umstrukturierung der Organisation in eine stark hierarchische, militärische Einrichtung, die ideologisch auf den Krieg ausgerichtet war. In Vorbereitung auf den zweiten Weltkrieg wurde weiterhin das Gelände in Griebnitzsee zum Hauptlager ausgebaut. Dadurch entstand eine zentrale Belieferungsstelle für das ganze deutsche Reich.

Doch militärische Vorbereitung genügte nicht. Um dem nationalsozialistischen Prestigebedürfnis gerecht zu werden, war ein DRK-Präsidialgebäude vonnöten, das ebenfalls auf dem Gelände Griebnitzsee errichtet wurde, um die Nähe zum Hauptlager zu gewährleisten. Noch heute finden wir diesen Bau auf dem Campus, zu großen Teilen angelehnt an die nationalsozialistischen Gebäude, die unter architektonischer Führung Albert Speers entstanden. Auch wenn Speer die Gestaltung des Präsidialgebäudes damals nicht selbst überwachte, sondern einen Beauftragten entsandte, ist sein Einfluss auf das monumentale Gebäude südlich des S-Bahnhofs bis heute im Gedächtnis geblieben.





Der Bau sollte architektonisch mit der geplanten „Filmstadt“ Babelsberg harmonieren und damit kulturelle und militärische Symbiose symbolisieren. Wegen des Krieges wurde die Planung der Filmstadt jedoch nie realisiert, so dass dem Präsidentengebäude bis heute sein geplantes Gegenüber fehlt. Die Bauarbeiten wurden hauptsächlich von Kriegsgefangenen verrichtet, die im nahe gelegenen Kriegsgefangenenlager untergebracht waren.

1944 reichte diese Arbeitskraft jedoch nicht mehr aus. Nachdem ein Bombenangriff der alliierten Lufttruppen, der große Teile der Kraftfahrzeughalle zerstörte, die Gefahr der allgemeinen Sicherheit verdeutlichte, wurde der Bau eines Luftschutzbunkers geplant. Um dieses Vorhaben zu realisieren, wurde eine Außenstelle des Konzentrationslagers Sachsenhausen auf dem heutigen Universitätsgelände errichtet. Ungefähr 100 Häftlinge waren in diesem sogenannten „Außenkommando Griebnitzsee“ untergebracht: eine von Stacheldraht umzäunte und von der SS bewachte Baracke, die sich östlich des Präsidentengebäudes befand.

Mit dem Ende des zweiten Weltkrieges endete auch die langjährige DRK-Nutzungsgeschichte des Geländes. Wie viele andere Mitwirkende in nationalsozialistischen Einrichtungen, entschieden sich

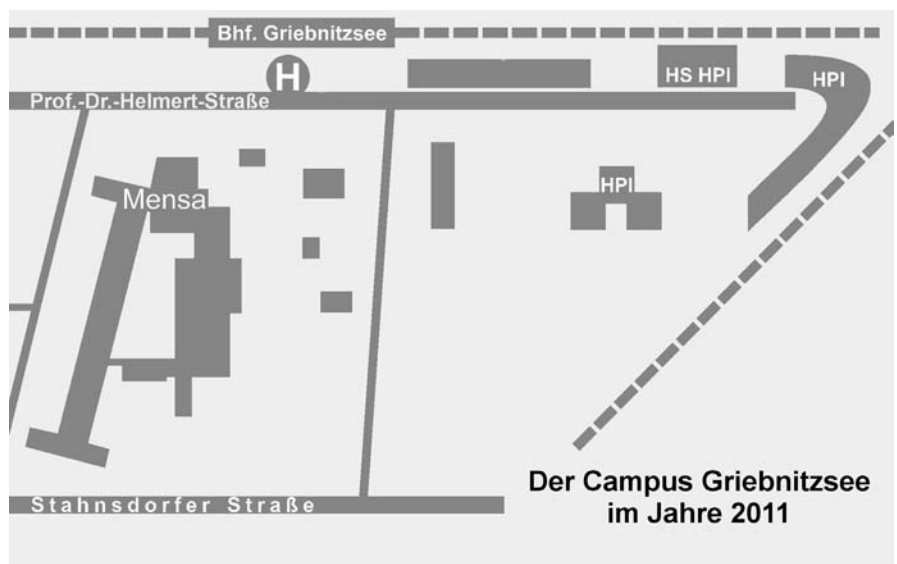
auch viele DRK-Mitarbeiter für die Flucht oder den Freitod. Auch Grawitz setzte als Geschäftsführer des DRK seinem eigenen Leben sowie dem seiner Familie ein Ende.

Im April 1945 besetzten die sowjetischen Truppen Babelsberg und daraufhin auch das DRK-Hauptlager. Das Präsidialgebäude wurde von nun an von der sowjetischen Militäradministration genutzt. Das DRK wurde im September auf Befehl derselben aufgelöst.

1952 wechselte das Gelände abermals seine Funktion. Die bereits 1947 gegründete „Deutsche Verwaltungsakademie“ übernahm das verkehrsgünstig gelegene Gelände

am Griebnitzsee. Die Aufgabe dieser Akademie war die Ausbildung von Mitarbeitern in Verwaltung, Wirtschaft und Justiz, die lernen sollten, sich der neuen politischen, wirtschaftlichen und gesellschaftlichen Ordnung anzupassen. Ein Jahr später wurde die „Deutsche Verwaltungsakademie“ mit der „Deutschen Hochschule der Justiz“ in Babelsberg zur „Deutschen Akademie für Staats- und Rechtswissenschaften ‚Walter Ulbricht‘“ zusammengeschlossen. In Zusammenhang mit der verstärkten internationalen Anerkennung der DDR trat die Ausbildung von Kadern für das Außenministerium und die Auslandsvertretung der DDR in den Vordergrund.

Nach dem Ende der Machtperiode Walter Ulbrichts erhielt die Universität 1972 den Namen



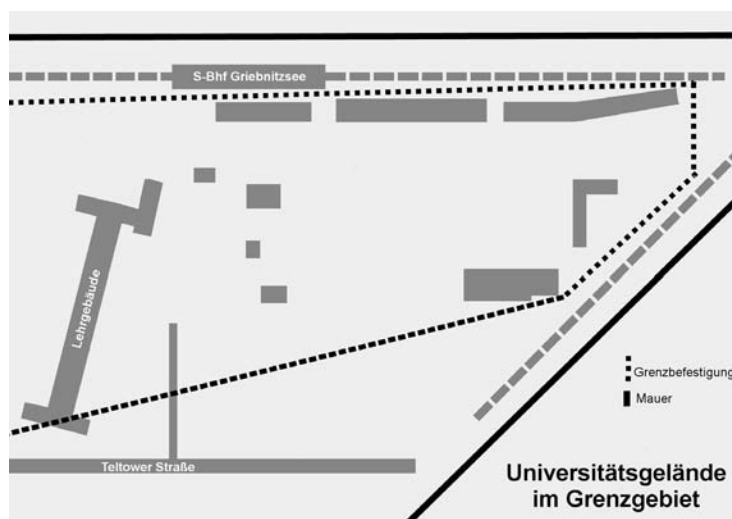
„Akademie für Staats- und Rechtswissenschaften der DDR“.

Auch der Mauerbau 1961 und die Grenzbeobachtung sollten den Campus Griebnitzsee beeinflussen. Der S-Bahnhof Griebnitzsee wurde zum streng kontrollierten Grenzbahnhof und eine Mauer trennte ihn vom damaligen Campus. Die Grenze verlief direkt südlich des Geländes. Ein Stacheldrahtzaun, der Bestandteil der Mauerbefestigung war, grenzte direkt an den südlichen Teil des ehemaligen Präsidialgebäudes. Die Fenster wurden teilweise vergittert und ein großer Teil des Geländes wurde zum Sperrgebiet. Auch dieser traurige Teil der Geschichte blieb nicht ereignislos. Bei einem Fluchtversuch zweier Jugendlicher aus der DDR kam es im Jahre 1962 am Gleisdreieck Griebnitzsee unweit des Campusgeländes zu zwei Toten. Zwischen dem Offizierschüler Peter Böhme und dem Postenführer Jörgen Schmidtchen, der seine Flucht verhindern wollte, kam es zu einer Schießerei, in der beide ums Leben kamen. Einem Freund, der mit Böhme zusammen fliehen wollte, gelang hingegen die Flucht.

Mit der Wende im Jahre 1990 beginnt der letzte Abschnitt der Geschichte des Griebnitzsee-Standortes. Nach der Umbenennung in „Hochschule für Recht und Verwaltung“ wird die Einrichtung bald darauf in die „Brandenburgische Landeshochschule“ überführt. Schließlich sollte sie mit der Gründung der Universität Potsdam am 15. Juli 1991 in ihre heutige Gestalt gebracht werden.

Doch vergessen ist die Geschichte des Campus Griebnitzsee nicht.

Noch heute bilden einige der DRK-Gebäude einen Bestandteil unserer akademischen Ausbildung. So befindet sich in dem ehemaligen Bekleidungslager des DRK heute die Universitätsbibliothek Babelsberg und das prunkvolle Präsidialgebäude, das unter dem Einfluss der nationalsozialistischen Führung entstand, ist heute das Haus 1 der Universität Potsdam und beinhaltet Hörsäle, Seminarräume sowie die Mensa. Sämtliche Hinweise auf die Nutzung durch die sowjetische Militäradministration wurden soweit wie möglich entfernt, und seit 1993 steht das Gebäude unter Denkmalschutz.



Der östliche Teil des DRK-Präsidiums einschließlich Barackenlager, Autohalle, Druckerei und Desinfektionsgebäude mussten jedoch den heutigen, modernen HPI - Einrichtungen weichen.

Auch das Gelände des ehemaligen Außenkommandos Griebnitzsee ist heute von Gebäuden der Universität überbaut, doch eine Gedenktafel an der Universitätsbibliothek soll die traurige Vergangenheit der Zwangsarbeit auf dem heutigen Campus vor der Vergessenheit bewahren.

Auf dem ehemaligen Todesstreifen befindet sich heute das Studentenwohnheim der Stahnsdorfer Straße und bietet mit 9 Häusern und 543 Zimmern den Studenten die Möglichkeit praktisch auf dem Campus zu wohnen.

Wenn auch die Universität Potsdam eine der jüngsten akademischen Einrichtungen Deutschlands ist, so ist die Geschichte des Standortes Griebnitzsee keineswegs bedeutungslos. Spuren davon ziehen sich wie ein Leitfaden über den Campus und weisen den Weg durch die moderne Geschichte Deutschlands. „In Potsdam die Zukunft erfinden.“, sagte Ministerpräsident Matthias Platzeck vor einigen Jahren in Zusammenhang mit dem HPI. Eine Zukunft, die die Verbindung zu der Vergangenheit bewahrt, denn in Griebnitzsee leben wir tagaus tagein inmitten von deutscher Historie.

– Susanne Bülow

Mehr Platz für alle...

Das neue Gebäude verbessert nicht nur die Studienbedingungen am HPI

Für den einen oder anderen mag das neue Hauptgebäude, das im Juli 2010 eröffnet wurde, bereits zu Gewohnheit geworden sein – fest steht aber: Mit den hinzugewonnen Räumlichkeiten haben sich Bedingungen für Forschung und Lehre am HPI erneut deutlich verbessert. Studenten haben nun – dank neuer Pool-, Seminar- und Clubräume – noch bessere Möglichkeiten, um alleine oder in Gruppen an Projekten oder Übungen zu arbeiten.

Aber nicht nur die Studenten profitieren vom neuen Gebäude. Auch Teile des 2005 eingerichteten Graduiertenkollegs, der HPI Research School sowie das im Juni 2010 eröffnete HPI-Spitzenforschungslabor „Future SOC Lab“ für Hochleistungsrechenverfahren und die drei Fachgebiete Internet-Technologien und –Systeme, Computergrafische Systeme sowie Human Computer Interaction sind nun im neuen Hauptgebäude untergebracht. Selbstverständlich ist nun auch die Institutsleitung hier zu finden. Insgesamt verfügt das Gebäude, das sich über vier Etagen erstreckt, über 6700 qm Bruttogeschossfläche – was etwa 4000 qm Hauptnutzfläche ausmacht – und 100 Räume. Die Baukosten belaufen sich auf gut 25 Mio. Euro, von denen etwa 9 Mio.

das Land Brandenburg bezahlt und 16 Mio. Hasso Plattner.

Das Obergeschoss wurde von HPI School of Design Thinking bezogen. Die gesamte Etage ist durch raumhohe Verglasung vom Foyer abgeschirmt und ist bewusst als „Open Space“ gestaltet.

Aufgrund der Erweiterung konnte die HPI School of Design Thinking die Anzahl der von ihr angebotenen Studienplätze für das Wintersemester 2010/2011 erneut erhöhen. Insgesamt 100 Studenten aus 14 Ländern haben im Oktober das Zusatzstudium im Bereich Innovation aufgenommen – die Zahl der Studenten, die an der HPI School of Design Thinking studieren hat sich somit seit deren Gründung im Jahr 2007 verdreifacht. Gut die Hälfte der neuen Studierenden kommen aus den Bundesländern Berlin und Brandenburg. Sie stammen von 12 Hochschulen und aus 36 verschiedenen Disziplinen. 76 von ihnen besuchen den Basis-Kurs, 24 den Fortgeschrittenen-Kurs. Das Zusatzstudium an der HPI-School of Design Thinking erfreut sich zunehmender Beliebtheit. 250 Studenten aus der ganzen Welt und 89 Disziplinen hatten sich um einen Studienplatz beworben.

– Magdalena Noffke



Wie gründet man ein HPI?

Der organisatorische und rechtliche Aufbau des Hasso-Plattner-Instituts.

Es heißt Hasso-Plattner-Institut, aber auf der Webseite steht, es sei eine GmbH. Getragen wird es von der Hasso-Plattner-Stiftung und die Studenten sind eingeschrieben an der Universität Potsdam. Wie passt das zusammen und was genau ist das Hasso-Plattner-Institut nun eigentlich?

Am Anfang steht eine Stiftung. Die Hasso-Plattner-Stiftung für Softwaresystemtechnik, gegründet von Hasso Plattner. Ihr ausgewiesener Stiftungszweck ist laut Stiftungsverzeichnis „die Förderung der Lehre und Forschung auf dem Gebiet der Softwaresystemtechnik.“ Hasso Plattner hat in diese Stiftung sein Kapital für das Institut eingebracht. Außerdem hat das Land Brandenburg das Grundstück gestiftet, auf dem Vorlesungs- und Hauptgebäude stehen. Stiftungen ist zu eigen, dass die eingebrachten Werte nicht wieder entnommen werden können. Die Stifter können aber über deren Einsatz entscheiden. Im Falle der Hasso-Plattner-Stiftung hat ein vierköpfiger Stiftungsvorstand die Fäden in der Hand. Neben dem Stifter und einem Berater sitzen zwei Vertreter des Landes Brandenburg im Vorstand. Momentan sind das die Präsidentin der Universität Potsdam, sowie die Bildungsministerin des Landes. Daneben gibt es einen Stiftungsrat, der aus zwölf Personen besteht und in allen wichtigen Angelegenheiten, wie zum Beispiel dem Haushalt, gehört wird.

Nun haben wir eine Stiftung, aber noch kein Institut. Zuerst einmal muss man das Institut als Körperschaft im Rechtsraum erschaffen. Hier kommt die GmbH (Gesellschaft mit beschränkter Haftung) als Gesellschaftsform der Wahl ins Spiel. Eine GmbH hat den Vorteil, dass die Haftung der Gesellschaft auf das Gesellschaftsvermögen beschränkt ist. Das Vermögen von Gesellschaftern wird nicht angerührt. Genau genommen ist das HPI sogar eine gGmbH (gemeinnützige GmbH). Sie muss gemäß ihrem Zweck arbeiten um den gemeinnützigen Status zu erhalten. Dieser ist, wie bei der Stiftung, die Förderung von Forschung und Lehre. Das Institut strebt also nicht nach Gewinn, sondern viel mehr nach neuen wissenschaftlichen Erkenntnissen.

Eine Gesellschaft braucht Gesellschafter. Die einzige Gesellschafterin des HPI ist die Hasso-Plattner-Stiftung selbst. Die ausführende Instanz im HPI ist die

Geschäftsführung. Diese besteht aus Prof. Dr. Christoph Meinel als eingesetztem Geschäftsführer sowie Dr. Timm Krohn als Prokurist.

Es fehlt nur noch eins, um das HPI zu einem offiziellen wissenschaftlichen Institut zu machen. Es muss zum Beispiel an eine Universität angegliedert sein. Dazu wurde ein Kooperationsvertrag mit der Universität Potsdam geschlossen, der das HPI zum An-Institut macht. Dadurch ist das Institut Teil der mathematisch-naturwissenschaftlichen Fakultät und die Forschung und Lehre gehören zum organisatorischen Teil der Universität. Wirtschaftlich bleiben beide Parteien aber getrennt. Diese Aufteilung zeigt sich zum Beispiel an der Berufung der Professoren des Instituts: Das Land Brandenburg ruft, vertreten durch die Universität Potsdam, die Professoren. Mit der Berufung werden sie sofort beurlaubt und können so bei der HPI GmbH angestellt werden. Dennoch bleiben sie frei in Forschung und Lehre.

Innere Organe

Im HPI selbst gibt es weitere Organe, die die Richtung des Instituts bestimmen. Zum einen gibt es das Professorium, die Versammlung der Professoren. Bei diesem regelmäßigen Treffen besprechen die Professoren aktuelle Themen am Institut. Gegebenenfalls bestimmen sie auch gemeinsam die Meinung des Professoriums.

Auf studentischer Seite steht der Fachschaftsrat als gewählte Vertretung der Studierendenschaft. Er bildet die Schnittstelle zwischen Studenten und Geschäftsführung. Bestimmend für die Lehre ist der Studienausschuss. Wer die Studienordnung des HPI schon einmal genauer gelesen hat, kennt die entsprechenden Regelungen (Studienordnung vom 24.06.2004, §21). Er tagt vier Mal im Jahr. Neben vier Professoren besteht er aus zwei wissenschaftlichen Mitarbeitern und zwei Studenten. Er achtet auf die Einhaltung der Studienordnung und behandelt besondere, in der Ordnung bestimmte Fälle.

— Patrick Rein

Sport am HPI

Von der „Human Playing Interaction“ zum HPI-Sportplatz

Sport für Informatiker – glaubt man den Vorurteilen, passt das ungefähr so gut wie ein Studentenwohnheim im Villenviertel. Dass Letzteres funktioniert, dürfte hinlänglich bekannt sein; Ersteres wird gerade in die Tat umgesetzt.

Denn als derzeit einziges HPI-Team gehen die Volleyballer der Mannschaft „Human Playing Interaction“ ins Rennen um die Punkte der Uni-Liga Volleyball. Gegründet wurde das Team von Michael Backmann. Der Neuling am HPI griff die Idee einer Kommilitonin auf und setzte sie in die Praxis um. Dabei stieß er auf reges Interesse. So besteht die Mannschaft dann auch hauptsächlich aus Erstsemestern, wird dabei jedoch auch von älteren Semestern und zwei externen Studenten unterstützt.

Derzeit befindet sich die Mannschaft auf dem achten Platz (von neun). Natürlich muss sich solch ein junges Team erst noch finden; erste Erfolge konnten jedoch schon errungen werden und es zeigen sich deutliche Fortschritte.



Das Volleyballteam des HPI

Sportplatz für Denksportler

Die Gründung des Volleyball-Teams ist noch allein auf die Initiative eines Studenten zurückzuführen. Inzwischen tut sich auch von „offizieller“ Seite etwas. Wer ab und zu das HPI-Gelände erkundet, wird sicher schon festgestellt haben, dass in der hintersten Ecke des HPI-Grundstückes in Richtung Berlin, eingekeilt zwischen Bahnschienen, ein neuer Multifunktionsplatz errichtet wurde. Sobald die-

ser benutzbar ist – zurzeit fehlt noch der endgültige Belag – kann man hier Fußball, Basketball und sogar Tennis spielen. Schließlich sind nur ausgeglichene Studenten auch glückliche Studenten.



Der neue Multifunktionsplatz: Ab Frühling soll hier Fußball, Basketball und Tennis gespielt werden können.

Während derzeit zwar keine „offiziellen“ HPI-Teams geplant sind, soll zumindest die Verwaltung und Zeiteinteilung von Sportaktivitäten auf dem Multifunktionsplatz in die Obhut eines neu zu gründenden Mini-Studentenklubs gelegt werden. Dieser würde die Verwaltung des HPI entlasten und mehr Verantwortung in die Hände der Studenten legen. Vorstellbar als Aufgabe für den Klub ist auch die Koordination und Organisation von Sportteams und deren Gründung. Es geht nicht darum, einzelne Teams zu leiten - vielmehr soll ein Prozess angestoßen werden. Der Bedarf zeichnet sich ab.

Initiative ist also gefordert. Warum sich das lohnt? Einer muss es ja wissen – Michael Backmann: „Einfach mal probieren, sich in einer Sportart zu engagieren. Bisher macht Volleyball an der Uni sehr viel Spaß und das tolle Team ist dabei noch das i-Tüpfelchen.“

Bei Interesse am Sport-Organisations-Klub:
Lena Hocker, lena.hocker@hpi.uni-potsdam.de

– Franz Liedke

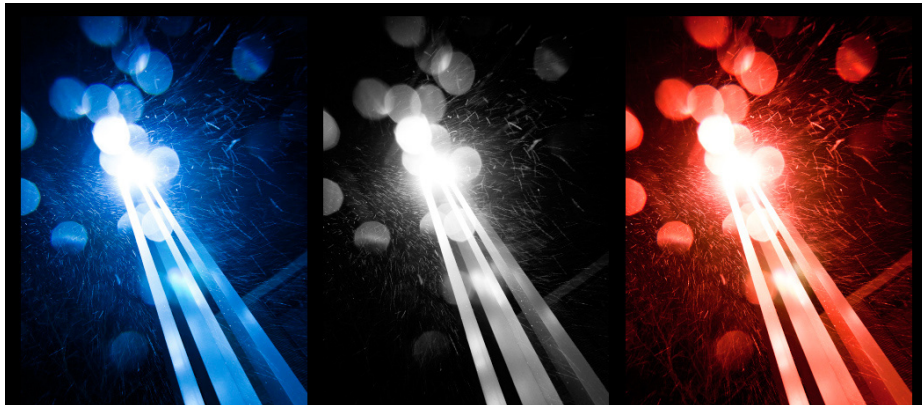
Der 2. HPI-Fotowettbewerb

Auch in diesem Jahr bot der Klub Fotografie den kreativen Köpfen des Campus' die Möglichkeit sich in Form unbewegter Bilder zu messen - Diesmal mit einer zusätzlichen Schwierigkeit: Alle Einreichungen sollten das Thema „Neubau und Neugebautes“ widerspiegeln. Doch ließen die Regeln allen Teilnehmern ebenso mehr Möglichkeiten als im Vorjahr, denn auch Bildreihen waren zugelassen.

Eine 4-köpfige Jury bewertete die Fotos. Kein Teilnehmer durfte Jurymitglied sein und die Jury hat alle Bilder anonymisiert und in zufälliger Reihenfolge bewertet.



Platz 1: „Neubau“ von Ludwig Wilhelm Wall



Platz 1: „New Lights“ von Ludwig Wilhelm Wall



Platz 2: „Turm“ von Ralf Diestelkämper



Platz 3: Von Franz Becker



Sonderpreis: „Newborn01“ von Pedro Flemming

Was den Teilnehmern der HPI-Sudoku-Meisterschaft erspart blieb...

Anleitung zum Rätsel lösen

In diesem Jahr sollte es das erste Mal eine Sudoku-Meisterschaft am HPI geben. Der Sieger sollte sich direkt für die Endrunde der Deutschen Sudoku-Meisterschaft qualifizieren. Auf diese Idee kam ich im Frühjahr letzten Jahres, als ich über Aktionen nachdachte, die ich schon immer mal organisieren wollte. Es gab bereits mehrere regionale Qualifikationen für die Sudoku-Meisterschaft. Neben Gießen, Hamburg, Augsburg und Recklinghausen sollte die östliche Fraktion mit einem Teilnehmer aus Potsdam vertreten werden. Nach einer Absprache mit dem FuV stellte sich die Durchführung auch als machbar heraus. Ich erstellte also im Sommer die Rätsel. Mein Ziel war es, euch auf Varianten des klassischen Sudokus aufmerksam zu machen und zu begeistern. Aus eigener Erfahrung kann ich berichten, dass diese wesentlich mehr Reize auslösen. So entstand unter anderem ein Differenz-Sudoku. Das Rätsel tauchte das erste Mal auf der Deutschen Sudoku-Meisterschaft im Jahre 2008 auf. Die Regeln sind nicht schwer:

In jeder Zeile, Spalte und in jedem 3×3-Block müssen die Ziffern von 1 bis 9 stehen. Die Zahlen am Rand geben die Summe der Ziffern in der entsprechenden Zeile bzw. Spalte an. In diese Summe gehen die Ziffern auf weißen Feldern positiv und die auf grauen Feldern negativ ein.

	9	29	13	15	-33	-1	9	-17	25
-7									
19									
-7									
3									
9									
23									
11									
19									
-21									

So weit, so gut. Wie geht man nun an diese Rätselart heran?

Man kann sich schnell überlegen, dass jede Summe aus zwei eindeutigen Teilsummen - die der positiven und die der negativen Zahlen - besteht. Deshalb sollte man sich zuerst diese Teilsummen aufschreiben. Die Summe 9 bildet sich z.B. wie folgt: $(45 + 9)/2 + (45-9)/2 = 27 - 18$.

Der nächste Schritt ist zu suchen, welche Summen aus bestimmten Zahlenkombinationen bestehen. Eindeutig ist zum Beispiel +6 in der mittleren Spalte. Die bildet sich aus drei (unterschiedlichen) Zahlen, also aus 1, 2 und 3. Schauen wir uns die Spalten direkt daneben in Kombination mit dem Block in der Mitte an. In Spalte 6 besteht die -23 aus 6, 8 und 9. Somit bleiben in dem betrachteten Block nur noch drei Zahlen übrig: 4, 5 und 7. Wenn wir uns die Summen in der Spalte 4 ansehen, stellen wir fest, dass $-15 = -1-2-3-4-5$ ist und somit für das weiße Feld nur die 7 verbleibt.

Die -15 hilft uns auch im letzten, grauen Feld. Da dort nur 1, 2 oder 3 stehen kann, muss in der Summe 12 in der letzten Zeile mindestens die 4 und 5 enthalten sein. Andernfalls bliebe für das graue Feld keine Zahl. 4 und 5 ergibt bereits 9, wodurch für die anderen beiden weißen Felder nur noch 1 und 2 bleibt. Demnach steht definitiv eine 3 in dem grauen Feld. Sonst wären nicht genug kleine Zahlen für die 12 da. Man sollte sich die Kandidaten in einem Feld aufschreiben, sobald man diese durch Schlussfolgerungen, welche man nicht sofort wiederfindet, herausgefunden hat. Wir markieren uns also auch die 1 und 2 in den entsprechenden grauen Feldern der Spalte.

	9	29	13	15	-33	-1	9	-17	25
-7									
19				12					
-7									
3				7	123	689			
9				45	123	689			
23				45	123	689			
11									
19				12					
-21				3					

	9	29	13	15	-33	-1	9	-17	25
-7									
19				12					
-7									
3		9		7	3	8		5	
9	4			5	1	9		23	
23				4	2	6		1	
11								234	
19				12					
-21				3					24

Betrachten wir in der Zeile 6 die -11. Bei zwei der drei grauen Felder sind bereits Kandidaten gegeben. Wenn man sich die kleinstmögliche Summe überlegt, stellt man fest, dass für das dritte Feld nur noch die 1 bleibt und in den anderen beiden Felder 4 und 6 stehen. Diese Überlegung, welche Summe sich minimal und welche maximal aus den Feldern ergibt, wo nur noch einige Zahlen stehen können, ist oft hilfreich.

Es kann auch hilfreich sein, wenn noch keine Kandidaten in den Feldern stehen. In diesem Fall betrachtet man für alle Felder die minimale oder maximale Summe. In Spalte 8 heißt das, die eindeutige Zahlenkombination für 14 zu finden, da die 1 bereits in einem grauen Feld steht. Anschließend kann man wieder den vorherigen Tipp in Zeile 4 anwenden. Die 24 hat bereits eine 7, wodurch noch 17 fehlen. Da in dem zweiten weißen Feld maximal 3 und in dem dritten maximal 5 steht, kann in dem ersten nur noch eine 9 stehen. Damit sind auch 3 und 5 fest. Es lassen sich mit den normalen Sudoku-Regeln auch schon einige Zahlen bestimmen, wo vorher noch mehrere Kandidaten standen. Auch steht die erste Zahl in der mittleren Zeile durch die Summe fest.

Betrachten wir uns nun die Zeile 7, speziell das vorletzte Feld. Die grauen Felder müssen sich zu -17 addieren. Da das vorletzte Feld weiß ist, kann diese Zahl nicht in der Summe sein. Da 17 bei fünf Zahlen maximal eine 7 enthält, gibt es nicht viele Kombinationen für diese fünf Zahlen. Da in dem weißen Feld nur 2, 3 oder 4 stehen kann, „blockiert“ diese Zahl einige von diesen Möglichkeiten. Wir können uns alle sechs Felder (die grauen und das eine weiße) ansehen. Die minimale Summe für sechs Felder ist 21. Das wären die Zahlen von 1 bis 6. Wenn die -17 eine -7 enthalten würde, bliebe keine passende Zahl für das weiße Feld. Demnach müssen die sechs Felder genau die Zahlen von 1 bis 6 beinhalten. Da die fünf grauen Felder in der Summe 17 ergeben, bleibt nur die 4 für das weiße Feld. Von den anderen drei weißen Feldern steht dadurch übrigens eine weitere Zahl fest.

Als nächstes schauen wir auf die zweite Zeile. An vorletzter Stelle steht eine Zahl größer oder gleich 6. Damit die Summe -13 erfüllt wird, entfallen die 8 und die 9. Es bleiben nur noch zwei Kombinationen: 1, 2, 3 und 7 oder 1, 2, 4 und 6. In den ersten drei grauen Feldern können also nur Zahlen von 1 bis 4 stehen. Das mittlere Feld ist eine 4, wie man schnell überprüfen kann. Demnach ist das letzte graue Feld eine 6.

	9	29	13	15	-33	-1	9	-17	25
-7									
19			12	12	4			6	
-7									
3		9		7	3	8		5	
9	4			5	1	9		3	
23				4	2	6		1	
11	89			89		7		4	
19				12					
-21				3				2	

	9	29	13	15	-33	-1	9	-17	25
-7			4	89	7			89	35
19			1	2	4			6	
-7			7	6	89			89	4
3	1	9	6	7	3	8	4	5	2
9	4	2	8	5	1	9		3	
23			3	4	2	6		1	
11	89	1	2	89	6	7	35	4	35
19			9	1	5	2		7	
-21			5	3	89	4		2	1

Der nächste Schritt nutzt wieder die kleinstmögliche Summe aus: die -13 in Zeile 8. Somit stehen die drei grauen Felder und einige sich daraus ergebene Zahlen fest. Ein Tipp beim Betrachten der minimalen oder maximalen Summen ist, wenn das errechnete Ergebnis nur um eins von der vorgegebenen Summe abweicht, ist die Kombination ebenfalls eindeutig. Zum Beispiel steht in der letzten Spalte schon eine 1. Dadurch bleiben für die minimale Summe der 10 nur noch 2, 3 und 4, also 9. Die einzig mögliche Zahlenkombination ist 2, 3 und 5.

Inzwischen ist das Sudoku ohne Beachtung der Zusatzregeln größtenteils lösbar. Man muss nur hin und wieder nach Teilsummen suchen, in denen nur noch eine Zahl fehlt. Eine Standardsudoku-Regel möchte ich noch erklären. In der dritten Zeile findet sich ein Paar, das heißt in dem Fall, dass in zwei Feldern nur noch die 8 und die 9 stehen können. Deshalb können in allen anderen Feldern der Zeile diese beiden Zahlen „gestrichen“ werden. Dadurch kann im vierten Feld nur noch die 6 stehen. Diesen Trick kann man auch in der ersten Zeile verwenden.

Wenn Fragen zu dem Rätsel oder generell zu Sudokus auftauchen, könnt ihr mich gerne persönlich oder per Mail fragen (maria.graber@student.hpi.uni-potsdam.de).

<Student> Das hatten wir schon in GDS
 <Dozent> Helfen sie mir nochmal kurz, was ist GDS?

<Dozent> Achtung, die Hazards kommen!

<Dozent> Am Anfang war der Anfangszustand.

<Dozent> Jetzt stellen Sie sich vor es kommt ein böser Mensch.

<Dozent> Also stellen Sie sich vor, ich komme.

(Es geht um Bucket-Sort)

<Student 1> Gibt es dafür auch einen deutschen Namen?

<Student 2> Töpfe sortieren.

<Dozent> Das nennt man auch kochen.

<Student> Squeak ist die Spielwiese der Objektorientierung, aber irgendwann muss man auch erwachsen werden...

<Dozent> Der Neubau steht vor der Tür!
 (28.4.2010)

Thorin und die zwölf Orks

Kurzgeschichte von Stefan Schaefer



Nun lauschet wohl und höret gut,
Von der Zwergen Kraft und Mut.
Von Thorin Eichenschild, der wohlbekannt
Im fernen, weiten Zwergenland.

Ich traf ihn einst in seinem Loch,
Wo er verkriecht sich heute noch.
Ein prahlerischer Narr ist er,
Ihn nicht zu mögen, fällt recht schwer.

Fragt ihn nur nach seinem Namen,
Ihr hört sogleich all seine Gram,
Und seine Abenteuer obendrein,
Von Drachen, Gold und Weibern fein.

Dies ist nun eine seiner Mären,
Die doch so viel glaubhafter wären,
Würd er nicht prahl'n damit zu jeder Stund.
So glaubt's ihm recht kein Straßenhund.

Einst ging er, so erzählt' er mir,
In die Taverne, genoss das Bier.
Doch ehe er es sich versah,
War ein Dutzend Orkse da.

Nun ist das schon verwunderlich,
Doch so ist die merkwürdig Geschichte.
Zwölf Orks in einem Zwergensaal -
Den füllt ein einz'ger schon drei Mal.

Thorin freilich zögert nicht -
Drum ist es ja seine Geschichte -

Spaltet dem ersten Ork den Kopf,
Zu Boden geht der arme Tropf.

Mutig schwingt er seine Axt umher.
Den Orks, den fällt es ziemlich schwer.
Der kleine Zwerg wirbelt herum,
Schon wieder fällt ein Ork tot um.

So geht es weiter und so fort,
Nun ist kein lebend Ork mehr dort.
Tot sind sie oder gefloh'n,
Begleitet von des Zwergen Hohn.

Der frohe Wirt beschenkt ihn reich,
Überschüttet ihn mit Gold sogleich,
Fehlen durft auch nicht das Bier,
Bekam fünf Fässer, sagt' er mir.

Auch die Tochter jung und fein,
Gab der Wirt ihm obendrein -
Nun sah ich nie ein Zwergenweib.
Stellt euch nur vor den massig Leib!

Thorin aber war doch sehr entzückt,
Hat sie am selben Tage noch beglückt!
Ein dreckig Lachen, gelbe Zähne
Entblößt er stets bei dieser Szene.

Nun kennt ihr des Thorins gute Mär,
Zumindest mal so ungefähr
Hört ihr sein Gelächter hall'n?
Ich hoff, es hat euch wohl gefall'n.

Kurzgefasst

Besuch des Präsidenten #1

Ich bereite den Besuch des Bundespräsidenten vor, und wollte mit Ihnen noch einmal kurz über das Protokoll reden. Wenn Herr Wulff den Raum betritt, dann ist es gewöhnlich so, dass alle aufstehen. Das ist kein Gesetz, aber ein Zeichen von Respekt, damit ist es natürlich Ihnen überlassen, was sie tun. Aber ich möchte dazu sagen, dass es nicht aus Respekt vor Christian Wulff als Person geschieht, sondern vor dem Amt, das er verkörpert.

So, oder zumindest so ähnlich, klingt es, wenn das Staatsoberhaupt der Bundesrepublik Deutschland an einer Universität angekündigt wird. Es ist der 09. November 2010 und im Rahmen seines Antrittsbesuchs stattet Christian Wulff in Begleitung von Ministerpräsident Matthias Platzeck dem HPI einen Besuch ab. Höhepunkt ist eine Podiumsdiskussion der Staatsmänner mit Studenten sowie Absolventen des Instituts- und mit Hasso Plattner. Recht locker übernimmt der Präsident selbst die Moderation des



Gesprächs und stellt Fragen, die mit Politik nur peripher in Berührung kommen. So geht es darum, wie sich für „die Betroffenen“ der Übergang von Schule zu Universität gestaltet, wie hoch der Arbeitsaufwand ist, und wie hoch die Kosten für so ein Studium überhaupt sein. Immer wieder kommt auch Prof. Plattner zu Wort, dessen Besuch zumindest für die Studenten eine Überraschung war. Kritisch hinterfragt der Stifter, wer für die Finanzierung eines Studiums eigentlich verantwortlich sei, und welche Akzente Deutschland im 21. Jahrhundert set-

zen möchte: „Deutschland unterstützt den Bergbau, aber Computertechnologie haben wir ziehen lassen. Wenn wir auf den ersten 10 Plätzen der Industrienationen bleiben wollen, müssen wir in IT und Software investieren“, so der Professor.

Es scheint, als hätte auch Wulff dies erkannt. Er möchte das Bellevue als Denkfabrik etablieren und dort auch die IT nicht vergessen. So findet sich in den Akzenten seine Präsidentschaft neben dem „Zusammenhalt der Gesellschaft“ auch der „Mut zum Wandel“ wieder.

Viel Zustimmung gab es auch für den dritten Akzent, die „Zukunft der Demokratie“, die der Bundespräsident gefährdet sieht. „Ich denke vor allem an die breitere Beteiligungsmöglichkeit, dass vielleicht die traditionelle Form [...] erweitert werden kann um Foren, Abstimmungen und Meinungsbildnisse. In wie weit man also die Interaktionen eines Netzes nutzen kann, um Entscheidungsprozesse auf breitere Grundlage zu stellen, um mehr Akzeptanz zu finden.“



Im Anschluss an die Diskussion stellte sich Wulff im Foyer direkt den Fragen der Studenten und verbrachte die nächste halbe Stunde mitten in einer dichten Traube neugieriger junger Menschen.

Am Ende des Abends zeigten sich die Zuhörer überwiegend positiv überrascht: „Ich habe weniger erwartet, nämlich einen Politiker, bekommen aber habe ich Gott sei Dank Jemanden, der sich über die Politik stellt.“ so Tim Wieschalla, Student im fünften Semester.

Inhaltlich schien das Gespräch die Zielgruppe zumindest erreicht zu haben: „Ich hatte absolut das Gefühl das [was Wulff ansprach] ist Liquid Feedback, ich hatte gleichzeitig das Gefühl, er kennt Liquid Feedback nicht, was nicht schlimm ist und um so grandioser, wenn er selbst auf diese Idee gekommen ist“ erzählte uns ein HPI-Student.

Besuch des Präsidenten, #2

Die Zeit der Desktopcomputer ist vorbei. Mehr und mehr stürmen mobile Endgeräte die Märkte, und auch „die Cloud“ gewinnt an Relevanz. John. L. Hennessy, Präsident der Stanford-Universität und Professor für Informatik war zum Besuch am HPI, und stellte in einem Kolloquium die Zukunft der elektronischen Datenverarbeitung vor.

Dienstag, 16. November 2010. Es ist knapp nach 10 Uhr, und Professor Hennessy begrüßt die Zuhörer. „Dekan, Provost, Präsident... das kommt alles daher, dass ich mich nie in einem Job halten kann.“

Begeistert macht sich der Professor an seinen



Vortrag. Er vertritt die These, dass wir uns zu lange auf Pipes und eine Vielzahl von Transistoren verlassen haben, und dadurch ineffizient geworden sind. „Wenn ein Prozessor vorhersagen muss, was als nächstes berechnet werden soll (zum Beispiel der gcc bei Verzweigungen), dann sind wir am Ende der Parallelität angekommen. Erwähnter gcc zum Beispiel liegt in 25% der Fälle falsch, und muss komplett fertig berechnete Verzweigungen wieder verwerfen.“

Dabei rückt die Energieeffizienz gerade jetzt, wo mobile Endgeräte wichtiger werden, wieder in den Fokus der Aufmerksamkeit.

Hennessy sieht die Zukunft der Informatik in feingranularem oder simultanem Multithreading.

„Eine schwere, wenn nicht sogar die schwierigste, Frage der modernen Informatik ist, wie man einfach und effizient für Multiprozessorarchitekturen programmieren kann. Ich bin mir aber sicher, dass es auf sie eine eindeutige Antwort geben wird. Und, dass sie von jemandem unter 40 gefunden werden wird.“

AmCham-Award für Hasso Plattner

Am 15. November wurde unser Instituts-Vater mit dem AmCham Transatlantic Partnership Award für außergewöhnliches wirtschaftliches, wissenschaftliches und gesellschaftliches Engagement geehrt. Die Auszeichnung der Amerikanischen Handelskammer in Deutschland wird jährlich verliehen und geht an engagierte Personen aus Deutschland und den Vereinigten Staaten, die einen herausragenden Beitrag zur Förderung der transatlantischen



Beziehungen geleistet haben.

Grund für die Ehrung Prof. Plattners ist dessen vielfältige Förderung junger Talente, zum Beispiel am HPI in Potsdam oder dem ‚Hasso Plattner Institute of Design‘ in Palo Alto, mit der er Verantwortung für die Gesellschaft und kommende Generationen übernimmt. Die Jury bezeichnete den SAP-Grüner als ein „Musterbeispiel eines Coporate Citizen“. Die Laudatio anlässlich der Preisverleihung im Berliner Museum für Kommunikation hielt der Präsident der Stanford-University, Prof. John L. Hennessy. Er

würdigte Plattner als außerordentliche Persönlichkeit, die Generationen, Fachgebiete und Länder beeinflusse. Auch Microsoft-Gründer Bill Gates ließ es sich nicht nehmen, seine persönlichen Glückwünsche zu überbringen.

Connect!

Anfang Dezember 2010 veranstaltete der Studentenklub Connect seine alljährliche Connect-Messe, bei der sich Studenten und Firmen kennenlernen und Kontakte knüpfen sollten. Um das Ganze für beide Seiten noch attraktiver und effizienter zu gestalten, wurden die Studenten in zwei Gruppen aufgeteilt. Während der eine Teil ein umfangreiches Bewerbungstraining von Experten aus der Praxis bekam, begab sich der andere Teil auf eine Speed Dating-Runde mit den Unternehmen. Dabei stellten sich die Vertreter der Firmen den Studenten innerhalb einer festen Zeitspanne vor und bewarben ihre Unternehmen. Danach wurden die beiden Gruppen getauscht. Im Anschluss daran und auch während des Speed Datings war genügend Zeit für Nachfragen und ausführlichere Gespräche. Durch die neue Organisation bekamen die Studenten die Möglich-

keiten, sich ein Bild von allen vertretenen Unternehmen zu machen und je nach Interesse Kontakte herzustellen. Auch für die Unternehmen war die Messe sicherlich lohnenswert, da die eine oder andere Firma für so manchen Studenten sicherlich sehr interessant war und auch immer noch ist.

Gründung des Filmklubs

Ab dem neuen Jahr wird es am HPI mit dem Filmklub einen weiteren Studentenklub geben. Er wird sich dem Entwurf, der Umsetzung und der Nachbereitung eigener, zunächst kleiner, Filmprojekte widmen. Zu Beginn soll durch Vorträge und Workshops, unter anderem zu den Themen "Struktur eines Filmprojekts", "Schnitt" und "Lichtgestaltung", Grundlagenwissen angeeignet werden.

Parallel werden die Klubmitglieder das Erlernte anhand eines ersten kleinen Films praktisch anwenden können. Später werden dann, aufbauend darauf, auch größere Filmprojekte in Angriff genommen.

Wenn ihr nun neugierig geworden seid und gerne selbst einmal vor oder hinter der Kamera stehen würdet, dann wendet euch einfach an david.heller@student.hpi.uni-potsdam.de



Studenten informieren sich beim Speed-Dating direkt bei Firmen.



Beim Training erfahren Studenten wissenswertes über Bewerbungen.

Sudokumeisterschaft am HPI

Am 21. Oktober fand im Hörsaal des HPI die erste lokale Sudokumeisterschaft statt. Die Teilnehmer rätselten zwei Runden à 45 min. Zur Auswahl standen neben klassischen Sudokus sowohl bekannte als auch neue Varianten. Eine Variante davon wird in dieser Ausgabe von der Autorin persönlich vorgestellt. Am Ende des Wettbewerbs landeten Franziska Häger (Master) und Stefan Lehmann (Bachelor) punktgleich auf dem ersten Platz. Die beiden Sieger einigten sich untereinander, sodass Franziska an der offiziellen Deutschen Sudoku Meisterschaft 2010 in Augsburg teilnehmen durfte. Sie hatte sich als Siegerin der HPI Sudoku Meisterschaft direkt qualifiziert. Auf den dritten Platz rätselte sich Alexander Schulze (Bachelor).



wendung von Objektorientierung und Erarbeitung komplexer Algorithmen vermittelt. Vorgestellt und diskutiert wurden die Arbeitsergebnisse in einer abschließenden Präsentation am Sonntag, 23. Januar. Sieben Teilnehmer des HPI-Camps aus dem Vorjahr hatten es bis ins Finale des Bundeswettbewerbs geschafft. Einer davon wurde sogar als Preisträger ausgezeichnet.

Den Bundeswettbewerb Informatik (www.bwinfo.de) gibt es seit 1980. Der Wettbewerb ist Kern der Initiative „Bundesweit Informatiknachwuchs fördern“ (BWINF) der Gesellschaft für Informatik e.V. (GI), des Fraunhofer-Verbands IuK-Technologie sowie des Max-Planck-Instituts für Informatik. Schirmherr ist der Bundespräsident.



Schüler entwerfen Möglichkeiten wie Scotland Yard auf einem Computer funktionieren kann.

HPI-Schülercamp „Fit for BWINF“

40 besonders talentierte Nachwuchsinformatiker aus ganz Deutschland hatte das Hasso-Plattner-Institut vom 21. bis 23. Januar zu Gast: Die 34 Schüler und 6 Schülerinnen hatten sich für die zweite Runde des Bundeswettbewerbs Informatik qualifiziert. Bei dem Camp am HPI werden sie fit gemacht für die Bewältigung der kniffligen Aufgaben. Die Teilnehmer lernen zum Beispiel, bestmögliche Taktiken für eine elektronisch simulierte Verbrecherjagd zu entwickeln. Wie im Brettspiel Scotland Yard geht es darum den Verbrecher mit Bus, Bahn und Schiff dingfest zu machen. Ein kleiner Unterschied ist, dass die Polizei nicht in London jagt, sondern in Potsdam. Die jungen Nachwuchsinformatiker wurden bei dem Camp vom HPI-Studentenklub betreut und bekamen wichtige Grundlagen der Softwareentwicklung wie Arbeit in kleinen Teams, An-

Open House an der D-School

Was um alles in der Welt ist Design Thinking? Wie können die neuen Ideen entstehen, die den Alltag verbessern? Fragen, die sich außerhalb des HPI oft auftun. Antworten darauf konnten Interessierte am 12. Januar in der HPI School of Design Thinking finden. Dort gab es die Möglichkeit, einen Blick hinter die Kulissen zu werfen und die spezielle Kultur der D-School kennenzulernen. Design Thinking Dozenten, Studenten und Mitarbeiter der HPI D-School standen dazu Rede und Antwort.

computer scientists

Item Name	Image	Description	Date Of Birth	Nationality	Known For
Alan Turing		Alan Mathison Turing, OBE, FRS was an English mathematician, logician, cryptanalyst and computer scientist. He was highly influential in the development of	23 June 1912	British	Computing pioneer
Charles Babbage		The Charles Babbage Institute (CBI) is an archives and research center dedicated to preserving the history of information technology and promoting ... Finding aids	26 December 1791	English	Mathematics, computing
Donald Knuth		Donald E. Knuth (高德纳), Professor Emeritus of The Art of Computer Programming at Stanford University, welcomes you to his home page. ...	January 10, 1938	American	3 possible values
Marvin Minsky		Marvin Minsky has made many contributions to AI, cognitive psychology, mathematics, computational linguistics, robotics, and optics. In recent years he has	1927-08-09	United States	Artificial intelligence
Adriaan van Wijngaarden		Adriaan J. van Wijngaarden (2 November 1916 – 7 February 1987) was an important mathematician and computer scientist who is considered by many to have been the	1916-11-02	Dutch	
Nicholas Negroponte		Nicholas Negroponte (born December 1, 1943) is a Greek-American architect best known as the founder and Chairman Emeritus of Massachusetts Institute of	December 1, 1943	American	Founder, MIT Media Lab
Claude Shannon		Claude Elwood Shannon (April 30, 1916 – February 24, 2001) was an American mathematician, electronic engineer, and cryptographer known as "the father of	April 30, 1916	American	Pioneer in information theory
Guido van Rossum		However, if you're American, you may also pronounce it as the Italian "Guido". I'm not too worried about the associations with mob assassins that some people have :-).	1960-01-31	Dutch	Python programming language

Impressum

Bildquellen:

Gespräch mit Herrn Börner Robin Schreiber

Kreitz-Artikel

Bild 1: Christoph Kreitz

Bild 2: <http://www.flickr.com/photos/neuwieser/4815369666/>Bild 3: [http://www.esa.int/esa-mm/mmg.pl?b=b](http://www.esa.int/esa-mm/mmg.pl?b=b&keyword=501&single=y&start=3&size=b)Bild 4: <http://mars.jpl.nasa.gov/msp98/slides/mplslide20.html>**Alumni Artikel Alumni Klub****Umfrage** Alle Logos sind Eigentum der jeweiligen Unternehmen**Kurzgeschichte**

Bild 1 „Dwarf Portrait- Full Pencils“, by „In-

kthinker“ at deviantArt.com

Bild 2 „Tavern Brawl“, by „Velinov“ at deviantArt.com

HPI@SAP Bild 1 und 2: Sebastian Oergel**Kurzgefasst HPI Connect** Katrin Honauer**Kurzgefasst weitere Bilder** Kay Herschelmann**Google Squared** Google**Vehicular Networks** CAR 2 CAR Communication Consortium**Druckerei:**

Druckerei Steffen

Inhaber: Helge Steffen

Friedrich-Ebert-Str. 74

14469 Potsdam

Auflage: 600 Stück**Redaktion:**

Maria Graber, Sebastian Oergel, Josefine Harzmann, Ste-

fan Schaefer, Magdalena Noffke, Leonid Berov, An-

drina Mascher, Patrick Rein, Robin Schreiber, Franz

Liedke, Jan Teske, Susanne Bülow, David Heller

Layout:

Robin Schreiber, David Heller, Magdalena Noffke, Patrick Rein

Covergestaltung:

Ludwig Wilhelm Wall, Robin Schreiber

Lektorat:

Maria Graber

Dr. Timm Krohn

V.i.S.d.P.:

Magdalena Noffke

Patrick Rein

Kontakt:

klub-zeitung@hpi.uni-potsdam.de

This delightful blank page loves you and wishes you a happy semester break

$(a_m b_k) x^{m-k} \Rightarrow Y_0(x) < \deg p(x)$
 $g(x) - (a_m b_k) x^{m-k} g(x)$
 $+ (a_m b_k) x^{m-k} g(x)$

8	7
4	4
1	3
1	3