AI magazine
Volume 38 Number 1
Spring 2017

Building AI Applications


The Fifth AAAI Conference on
Human Computation and
Crowdsourcing (HCOMP 2017)
www.humancomputation.com/2017
October 24 – 26, 2017
Hilton Regency Québec
Québec City, Canada.
Submissions Due May 4, 2017


Submissions due May 25, 2017
Artificial Intelligence and Interactive Digital Entertainment
Beyond Games
The Thirteenth Annual AAAI Conference
on Artificial Intelligence and Interactive
Digital Entertainment (AIIDE-17)
Little Cottonwood Canyon, Utah
October 5–9, 2017
aiide.org

AI magazine

VOLUME 38, NUMBER 1 Spring 2017 ISSN 2371-9621 (online) ISSN 0738-4602 (print)

Cover: *Building AI Applications by James Gary, New York, New York.*

The guest editors for the 2017 special issue on Innovative Applications of AI are Peter Z. Yeh and James Crawford.

INNOVATION APPLICATIONS OF AI ARTICLES

ARTICLE

COMPETITION REPORT

AI IN INDUSTRY

WORKSHOP REPORT

DEPARTMENTS

@RealAAAI

AI magazine

An Official Publication of the Association for the Advancement of Artificial Intelligence

aimagazine.org

ISSN 0738-4602 *(print)* ISSN 2371-9621 *(online)*

Submissions

Submissions information is available at http://aaai.org/ojs/index.php/aimagazine/information/authors. Authors whose work is accepted for publication will be required to revise their work to conform reasonably to *AI Magazine* styles. Author's guidelines are available at aaai.org/ojs/index.php/aimagazine/about/submissions#authorGuidelines. If an article is accepted for publication, a new electronic copy will also be required. Although *AI Magazine* generally grants reasonable deference to an author's work, the *Magazine* retains the right to determine the final published form of every article.

Calendar items should be posted electronically (at least two months prior to the event or deadline). Use the calendar insertion form at aimagazine.org. News items should be sent to the News Editor, *AI Magazine*, 2275 East Bayshore Road, Suite 160, Palo Alto, CA 94303. (650) 328-3123. Please do **not** send news releases via either e-mail or fax, and do **not** send news releases to any of the other editors.

Advertising

AI Magazine, 2275 East Bayshore Road, Suite 160, Palo Alto, CA 94303, (650) 328-3123; *Fax* (650) 321-4457. *Web:* aimagazine.org. Web-based job postings can be made using the form at https://www.aaai.org/Forms/jobs-submit.php.

Microfilm, Back, or Replacement Copies

Replacement copies (for current issue only) are available upon written request and a check for $25.00. Back issues are also available (cost may differ). Send replacement or back order requests to AAAI. Microform copies are available from ProQuest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106. Telephone (800) 521-3044 or (734) 761-4700.

Copying Articles for Personal Use

Address Change

Please notify AAAI eight weeks in advance of a change of address. Send electronically via MemberClicks or by e-mailing us to membership16@aaai.org.

Subscriptions

AI Magazine (ISSN 0738-4602) is published quarterly in March, June, September, and December by the Association for the Advancement of Artificial Intelligence (AAAI), 2275 East Bayshore Road, Suite 160, Palo Alto, CA 94303, telephone (650) 328-3123. *AI Magazine* is a direct benefit of membership in AAAI. Membership dues are $145.00 individual, $75.00 student, and $285.00 academic / corporate libraries. Subscription price of $50.00 per year is included in dues; the balance of your dues may be tax deductible as a charitable contribution; consult your tax advisor for details. Inquiries regarding membership in the Association for the Advancement of Artificial Intelligence should be sent to AAAI at the above address.

PERIODICALS POSTAGE PAID at Palo Alto CA and additional mailing offices. *Postmaster:* Change Service Requested. Send address changes to *AI Magazine*, 2275 East Bayshore Road, Suite 160, Palo Alto, CA 94303.

AAAI SPONSORS

AI Journal
National Science Foundation
Microsoft Research
Amazon
IBM Research
Baidu
Tencent
Facebook
Capital One
Infosys Limited
Shanghai Xiaoi Robot Co., Ltd
Beijing Bytedance Technology Co., Ltd
Spare5
StichFix
Future of Life Institute
Google
Adobe
Cheetah Mobile
Lionbridge Technologies
USC/ISI
Alegion
Information Evolution
MicroWorkers
University of Texas at Austin
Qatar Computing Research Institute
David Aha
David E. Smith
ACM / SIGAI
CRA Computing Community Consortium
Gesis Leibniz Institute for the Social Sciences
Journal of the Association for Information Science and Technology

Editorial

Expository AI Applications

Ashok K. Goel

AI Magazine and the conference series on Innovative Applications of AI (IAAI) have had a close relationship ever since the latter's inception 1989. This close relationship between *AI Magazine* and IAAI is no coincidence. Much of our society knows about and interacts with AI mainly through its applications. Applications are the source of many problems that AI research seeks to address, and we often measure our progress through successful applications. Applications are how AI makes an impact on the world. Hence, as "the journal of record for the AI community," it is only logical for *AI Magazine* to cover AI applications extensively. Thus, every year since 1990, *AI Magazine* typically has published a special issue based on the previous year's IAAI conference. The articles in this special issue derive from IAAI 2016, and I thank Peter Yeh and James Crawford for putting it together.

I should note another feature of the articles in this special issue on AI applications: they contain few mathematical formalisms and equations. This too is by design. At *AI Magazine,* we are incrementally moving towards expository articles that are accessible to the broader AI community. It is important that the AI community at large has access to serious AI research but in a language it can understand. I thank the authors of the articles in this issue for their cooperation in going through a second round of reviewing to make the articles more explanatory and less "technical" and yet maintain high quality. As we move forward, we will continue the move towards descriptive articles accessible to the AI community at large.

Editorial Introduction to the Special Articles in the Spring Issue

Innovative Applications of Artificial Intelligence 2016

Peter Z. Yeh and James Crawford

■ *This issue features expanded versions of articles selected from the 2016 AAAI Conference on Innovative Applications of Artificial Intelligence held in Phoenix, Arizona. We present a selection of three articles that describe deployed applications, two articles that discuss work on emerging applications, and an article based on the 2016 Robert S. Engelmore Memorial Lecture.*

AAAI's Innovative Applications of Artificial Intelligence Conference was founded in 1989 to showcase the successful application of artificial intelligence technology to real-world problems and its deployment into the hands of end users. Since then, we have seen examples of AI applied to domains as varied as medicine, education, manufacturing, transportation, user modeling, military operations, and citizen science. This year, the 2016 conference continued the tradition with a selection of deployed applications describing systems in use by their intended end users, emerging applications describing works in progress, and challenge problem papers that discuss associated research and development challenges in applying artificial intelligence to real-world problems.

Our first article is based on the IAAI-16 Robert S. Engelmore Memorial Lecture given by Reid G. Smith at AAAI/IAAI 2016 in honor of Bob Engelmore's extraordinary service to AAAI and his contributions to applied AI. The article Smith and coauthor Joshua Eckroth later wrote is titled Building AI Applications: Yesterday, Today, and Tomorrow. It focuses on changes in the world of computing over the last three decades that made building AI applications more feasible. The article also examines lessons learned during this time (drawing from experiences that have been documented in IAAI conference proceedings since 1989), and distills these lessons into succinct advice for future application builders.

In the second article, PAWS — A Deployed Game-Theoretic Application to Combat Poaching, Fei Fang, Thanh H. Nguyen, Rob Pickles, Wai Y. Lam, Gopalasamy R. Clements, Bo An, Amandeep Singh, Brian C. Schwedock, Milind Tambe, and Andrew Lemieux describe a deployed game-theoretic application for optimizing foot patrols to combat poaching in Southeast Asia. The authors report on significant evolution of PAWS from a proposed decision aid to a regularly deployed application over the last two years. Key technical advances that led to PAWS's regular deployment

  ISSN 0738-4602

are outlined. These advances include incorporating complex topographic features, handling uncertainties in species distribution, handling complex patrol scheduling constraints, and more. The authors also report key lessons learned ranging from the importance of firsthand immersion in the security environment of concern to minimizing the need for extra equipment in order to further ease future deployments of PAWS. The benefit of PAWS to its intended end users was demonstrated by the continued deployment of PAWS at existing sites in Malaysia, and steps taken at the time of writing to expand PAWS to additional sites.

In our third article, Deploying nEmesis: Preventing Foodborne Illness by Data Mining Social Media, Adam Sadilek, Henry Kautz, Laren DiPrete, Brian Labus, Eric Portman, Jack Teitel, and Vincent Silenzio describe a deployed application that automatically detects venues likely to pose a public health hazard by applying machine-learning techniques to Twitter data. The authors demonstrate nEmesis's efficacy in the Las Vegas metropolitan area in a double-blind experiment conducted over three months in collaboration with Nevada's health department, and show that the deployed application is 64 percent more effective at identifying problematic venues than the current state of the art. If fully deployed, the nEmesis approach could prevent more than 9000 cases of foodborne illness and 557 hospitalizations annually in Las Vegas alone.

In our fourth article, Ontology Reengineering: A Case Study from the Automotive Industry, Nestor Rychtyckyj, Venkatesh Raman, Baskaran Sankaranarayanan, P. Sreenivasa Kumar, and Deepak Khemani discuss an effort to reengineer an existing ontology deployed at Ford. Ford has been utilizing an AI-based system to manage process planning for vehicle assembly at its assembly plants around the world for more than 25 years. The knowledge about Ford's processes is contained in an ontology originally developed using the KL-ONE representation language and methodology. However, the scope of this AI system has increased over the years to include additional functionality on ergonomics and powertrain assembly. Hence, the existing KL-ONE ontology needs to be reengineered into a semantic web OWL/RDF ontology to satisfy the increased scope of the AI system and to enable other applications within Ford to easily make use of it as well.

In our fifth article, Automated Volumetric Intravascular Plaque Classification Using Optical Coherence Tomography, Ronny Shalev, Daisuke Nakamura, Setsu Nishino, Andrew M. Rollins, Hiram G. Bezerra, David L. Wilson, and Soumya Ray describe an emerging application to identify different plaque types in blood vessel images using machine-learning methods. An estimated 17.5 million people died from a cardiovascular disease in 2012, and most acute coronary events result from rupture of the protective fibrous cap overlying an atherosclerotic plaque. Hence, early identification of plaque types that can potentially rupture is of great importance. The state-of-the-art approach to imaging blood vessels is intravascular optical coherence tomography (IVOCT), but this is an offline approach where the images are first collected and then manually analyzed one image at a time to identify regions at risk. This process is extremely laborious, time consuming, and error prone. Initial empirical results presented by the authors using real OCT data show that the proposed approach can identify different plaque types efficiently and with high accuracy across multiple patients.

In our sixth article, Using Global Constraints to Automate Regression Testing, Arnaud Gotlieb and Dusica Marijan describe an emerging application to automate regression testing. Regression testing is a crucial verification step in the software development and release process, but the selection of test cases is challenging due to several factors such the limited time available for testing. This problem, called test suite reduction (TSR), is usually addressed by validation engineers through manual analysis or by using approximation techniques. To address these limitations, the authors apply AI techniques such as constraint programming and global constraints to automate the process. Moreover, the authors' work is conducted in the context of an industrial application in the communication domain with the goal to eventually deploy the emerging application to test a complete product line of conferencing systems in continuous delivery mode. Initial experimental evaluations show promising results.

In the tradition of previous special issues on innovative applications of artificial intelligence, and consistent with the goals of the IAAI conference, the articles in this issue describe work that is strongly grounded in the needs of end users. We hope that you enjoy the articles, and that they both provide insight into the application development process and help to expand your view of what is possible with AI technology. We also invite you to submit a description of your next AI application to IAAI.

Peter Z. Yeh is a senior manager of AI technology and senior principal scientist at Nuance Communications. His research interests lie at the intersection of semantic technologies, data and web mining, and natural language understanding. At Nuance, Yeh's research focuses on developing the next generation of intelligent virtual assistants and the underlying AI technologies necessary to enable this. Prior to joining Nuance, he was a research lead at Accenture Technology Labs where he was responsible for investigating and applying AI technologies to various enterprise problems ranging from data management to advanced analytics. Yeh is the author of more than 40 papers in peer-reviewed journals and conferences, and holds eight patents. He received his Ph.D. in computer science from the University of Texas at Austin.

James Crawford is the founder and chief executive officer at Orbital Insight, Inc. He has held distinguished positions at NASA, led startups, and in 2009 became engineering director for Google Books in charge of scanning the world's books. Crawford served as vice president of engineering and executive vice president of engineering at Composite Software, Inc. He served at Ames Research Center, NASA's center of excellence for information technology. Prior to joining NASA, he led the optimization team at i2 Technologies, worked at AT&T Bell Laboratories, and cofounded the Computational Intelligence Research Laboratory (CIRL) at the University of Oregon. Crawford is the author of more than 15 papers in referred journals and conferences, and holds five patents. He earned his Ph.D. in artificial intelligence and master's degree in computer science from the University of Texas at Austin.

Robert S. Engelmore Award Article

Building AI Applications: Yesterday, Today, and Tomorrow

Reid G. Smith, Joshua Eckroth

■ *AI applications have been deployed and used for industrial, government, and consumer purposes for many years. The experiences have been documented in IAAI conference proceedings since 1989. Over the years, the breadth of applications has expanded many times over and AI systems have become more commonplace. Indeed, AI has recently become a focal point in the industrial and consumer consciousness. This article focuses on changes in the world of computing over the last three decades that made building AI applications more feasible. We then examine lessons learned during this time and distill these lessons into succinct advice for future application builders.*

AI Applications of Yesterday and Today

As with the AAAI itself, the Innovative Applications of Artificial Intelligence conference (IAAI) was the brainchild of Raj Reddy. Howie Shrobe summarized the context: "... the emergence of scientific achievements had triggered opportunities to tackle new problems ... The point of the conference was to exchange information about what really works and what the real problems are. The goal was to lead to better technology, to find and remedy current deficiencies, and to solve real problems" (Shrobe 1996).

In the preface to the proceedings of the first IAAI conference in 1989, Herb Schorr, program chair, made some interesting comments about the 1989 state of several of the AI technologies that are now well established (Schorr and Rappaport 1989):

  ISSN 0738-4602

Problems and System Types	Specific Applications
Rule-Based Systems: *Widely applied base technology*	TurboTax
Credit Card Fraud Alert	Netflix Recommender
Insurance	FareCast, Google Flights, Kayak price predictor
Scheduling: *Maintenance, Crew, Gate*	Narrative Science GameChanger
Video Games	IBM Watson
Search Engines	Dragon Speech Recognition
Augmented/Virtual Reality	Amazon Robotics / Kiva Systems
Photo Face Recognition	Roomba
Handwriting Recognition: *Mail Sorting, ATM-Checks*	Kinect
Translation	Driver-Assist / Self-Driving Vehicles
Deep Learning	Siri, Cortana, Amazon Echo
Robotics	

Table 1. AI in Use.

> *Expert Systems:* "Nearly all [applications] are expert systems because it is in this form that AI is most rapidly coming into widespread use."
>
> *Robotics:* "[N]o robot software system for complex tasks is commercially available ... robots seem to be stuck with their early applications and have made small commercial progress in the last few years."
>
> *Neural Networks:* "[W]e know of no neural networks in practical day-to-day use ... while this technology appears to possess vast potential ... we leave it for this book's successor to cover such applications."
>
> *Natural Language Processing (NLP):* "[NLP] has been constrained historically by limitations of computational power, but the fantastic progression of computational cost/performance has eliminated this bottleneck. ... [But] today's applications ... are very limited and very few low-level natural language functions are being deployed."

Expert systems were common and successful in the late 1980s in large part because they were able to incorporate domain- and task-specific knowledge; their reasoning engines were relatively simple and, consequently, these systems could be deployed on computer hardware available at the time.

Herb Schorr's comments about robotics, neural networks, and NLP are prescient. In fact, in each area, the story has completely flipped since 1989: today, robots are common in industrial and service applications such as factory automation and farming, and their deployment continues to grow; neural networks make up significant portions of vision, speech, and text-processing systems, and deep learning is one of the more popular research and application areas in AI today; and natural language processing can be found in many applications that billions of people use every day, such as search engines, personal assistants, and web-connected speakers.

Today, AI is everywhere. By contrast with 1989, when very few AI companies were in existence, today many companies, from early stage startups to mature enterprises, are developing AI applications (Zilis 2015).

The world of AI apps is very different as well. In the early days, AI was viewed with suspicion in industry as only the latest hype. Today, AI apps are all around us. Indeed, AI and machine learning are expected in almost every app.

Many, perhaps most, large organizations are making use of AI technologies for market forecasting, customer support, recruiting, fraud detection, scheduling and planning, and other uses. Consumer-oriented examples of AI include Google's search engine, self-driving cars, and Google Now; Apple's Siri (Cheyer 2014); Microsoft's Cortana and Bing; Amazon's Echo; Facebook's automatic photo tagging; Netflix's movie recommendations; and automated check deposits using one of many mobile banking applications. Table 1 shows even more problem and system types, plus specific applications, several of which have been presented at IAAI or AAAI over the years. Of course, not all of these examples are commonly recognized as AI applications — the AI features have disappeared into the fabric. Modern search engines are a good example of this phenomenon.

Although interest in computer science in general dropped after the dot-com crash of the early 2000s (Thibodeau 2008), the last few years have seen a steady growth in the number of news stories about AI appearing in popular media, as discovered by AAAI's automated AI in the News weekly news bot (Eckroth et al. 2012). Figure 1 shows this trend. Com-

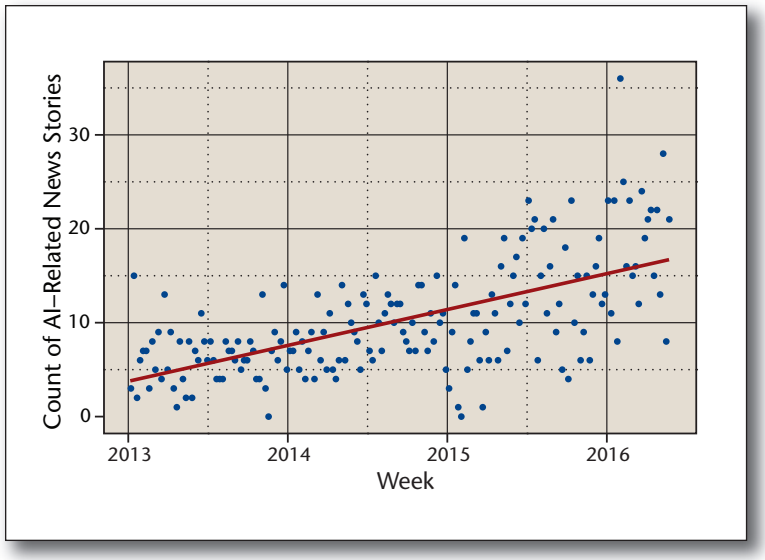


Figure 1. Count of News Stories Found by AI in the News for Each Week.

The line visualizes a linear regression.

puter science undergraduate enrollments have exhibited a similar trend, and as of 2014, more Ph.D. graduates are employed in the field of AI, across academe and industry, than any other subfield of computer science (Zweben and Bizot 2015).

We believe there are several factors contributing to the growth since the first IAAI conference in 1989.

Moore's Law

One of the most important changes is the growth in hardware performance. To illustrate, we will consider a deployed system from the first IAAI conference. Clancy, Gerald, and Arnold (1989) developed an expert system that "assisted attorneys and paralegals in the closing process for commercial real estate mortgage loans." Their system was required to work on IBM PCs with Intel 80286 processors and 640 KB memory. They wrote that the PC's limited memory posed a "critical technical consideration" and they programmed their system to swap subsets of the knowledge base in and out of memory during normal operation. Their solution, and more arcane memory management techniques, are likely familiar to AI system builders who were active in the early days.

Today, consumer hardware is 2500 times faster (from 1.5 million instructions per second, or MIPS, on an 80286 compared to 3783 MIPS on an Intel i7-3770K, quad core).[1] It is now common to find more powerful servers, with more than 10 cores, which means that we can take advantage of a speed-up of 10,000 — and growing.

In addition, consumer hardware contains 25,000 times as much memory (640 KB to 16 GB), and 50,000 times the disk capacity (40 MB to 2 TB). (See also Preshing [2012].) This explosive growth of computing power can be attributed to Moore's law, which summarizes Gordon Moore's observation that the number of transistors in integrated circuits doubles approximately every two years. The diversity of integrated circuits has also grown, resulting in general-purpose GPUs (with multiteraflop performance; that is, trillions of floating-point operations per second) that have helped usher in the era of practical machine learning.

The Internet

The global impact of the Internet on science and society in general cannot be overstated. One of the more interesting effects of the Internet, and the web in particular, on AI systems was noted in Halevy, Norvig, and Pereira's (2009) article The Unreasonable Effectiveness of Data. They note that the web enables easy acquisition of massive amounts of data from billions of web pages, provided by billions of users.

Halevy and colleagues further argue that sophisticated knowledge representation and reasoning systems may be unnecessary, even detrimental when a massive corpus such as the web is available. For example, in the case of the semantic web, they suggest that writing an ontology, adding metadata markup for web pages, and building a complex reasoning system is likely to be more expensive and error prone than simply querying the vast, unstructured corpus with shallow parsing and straightforward statistical analysis. The long tail of real-world concepts defeats any effort to develop a grand model of everyday reasoning, but the long tail is well represented in massive data sets such as the web.

Open Source Software

Frustrated with a trend toward proprietary development practices at the Massachusetts Institute of Technology (MIT), Richard Stallman started the GNU Project in 1983 to create a free and open source UNIX-like operating system. The idea spread and has been harnessed by various groups, resulting in an abundance of high-quality open source software. The internet played a large role in the distribution and development of open source software. In particular, development of the Linux operating system, which was built with GNU project tools, grew rapidly in the 1990s due to the availability of newsgroups, email, and file sharing. As of November 2015, 99 percent of the 500 most powerful supercomputers in the world run the open source Linux operating system. Most software development environments in use today are open source (Oracle's JVM, Microsoft's .Net, C/C++ compilers, Python, and others), and many open source libraries and toolkits are available for AI-specific tasks, a sampling of which are shown in table 2.

Machine Learning

Because available hardware did not allow large-scale

TensorFlow (Google)	Machine learning toolkit
OpenCV (itseez)	Computer vision library
Sphinx (CMU)	Speech recognition toolkit
Drools (Red Hat)	Rule-driven expert system shell, planning engine
GATE (University of Sheffield)	Natural language processing toolkit
Robot Operating System (Open Source Robotics Foundation)	Platform for integrating various algorithms and libraries related to robotics

Table 2. Sample Open Source AI libraries and Toolkits.

numeric computation, early AI systems relied on heuristics encoded symbolically, even in data-heavy tasks such as computer vision. Though significant progress continues to be made in symbolic reasoning, it is clear that the power available to process vast amounts of data — even at high data rates — has enabled practical deployment of machine-learning techniques and resulted in a wide diversity of successful applications from speech recognition and face recognition to self-driving cars. Additionally, it is interesting to note that machine learning seems to dominate the popular perspective of AI today, just as it was considered by early AI researchers to be an essential component of intelligence (Minsky 1961).

We find evidence of this in Google Trends data, shown in figure 2. Interest in AI appeared to wane after the dot-com crash and hit a low around 2009. The search term *computer science* follows a similar trend. Recently, interest has renewed and appears to be supported by machine learning and recent work in deep learning.

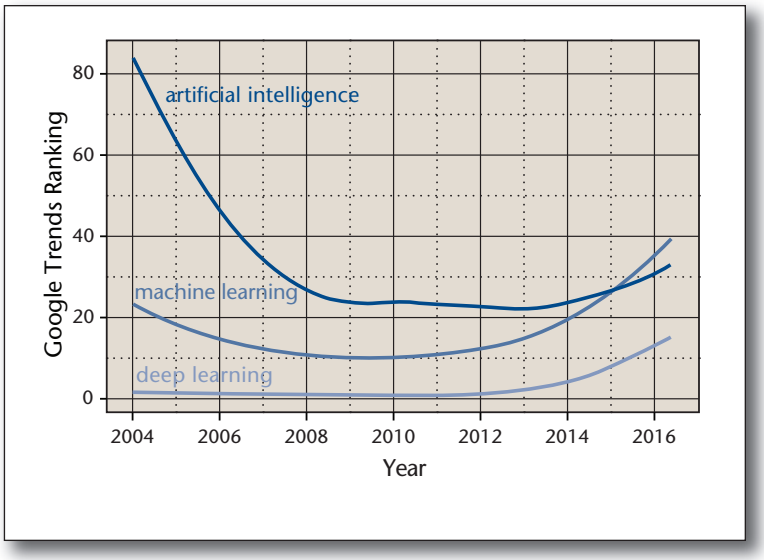


Figure 2. Google Trends Rankings for Various Search Terms.

The y-axis represents smoothed relative interest.

Reduced Business Risk

These points add up to another change. Because of the greater computing power and more readily available data sets and software, today there is less need to build massive technology platforms. Hence, it is cheaper to build AI systems. More effort can be spent on solving specific business problems, thereby reducing the risk associated with artificial intelligence.

Compared to 1989, today it is orders of magnitude easier to integrate AI systems into a company's overall IT portfolio. The reasons include: modern AI systems utilize standard hardware and software (in many cases); they integrate more easily into existing architectures; the iterative development process pioneered in AI projects has become common across IT; and, the success of high-profile AI systems such as Watson and Siri means that most people know that AI can work in the real world. (The authors thank Neil Jacobstein for this insight.)

Distributions and Trends from the IAAI Conferences

At the outset, IAAI included only applications that had been deployed; that is, for which there was experience based on actual use, and for which payoff could be estimated. In 1997, an emerging applications track was added to bridge the gap between AI research and AI application development. The goal was to support information sharing among researchers and system builders: researchers could see which techniques proved fruitful in deployed applications, and builders could learn of emerging techniques that had yet to be proven in the field but showed promise.

An analysis of the topics covered by IAAI articles is shown in figures 3 and 4. This analysis is provided by i2k Connect (i2kconnect.com), whose goal is help organizations find, filter, and analyze unstructured

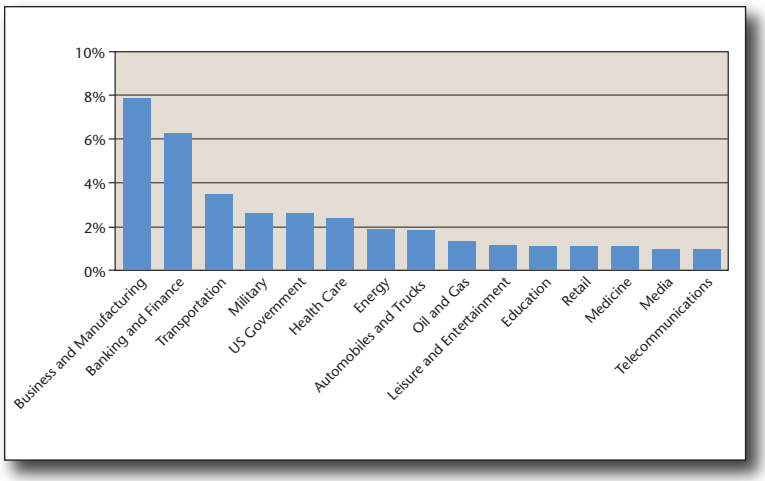


Figure 3. Top 15 Industry Topics in IAAI Articles, 1989–2016.

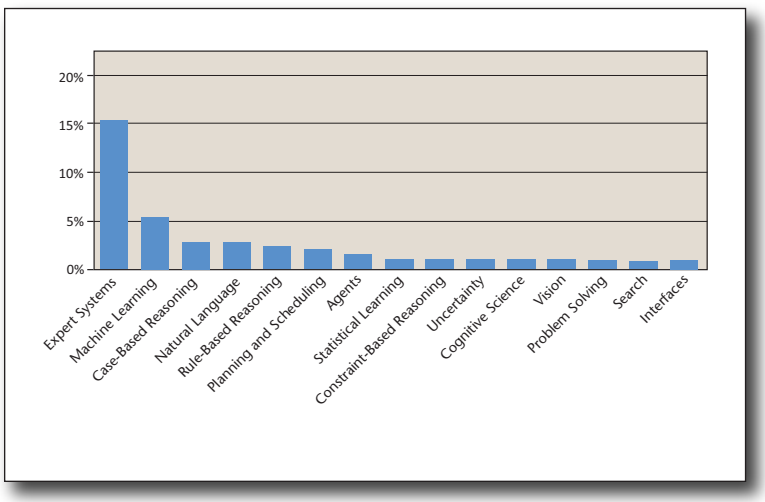


Figure 4. Top 15 Technology Topics in IAAI Articles, 1989–2016.

data by transformation into structured data. The platform automatically tags documents with accurate and consistent metadata, guided and enriched by subject matter expertise. The figures show data from deployed applications only — 316 of them. The figures include only 15 of a long tail of more than 100 industry and technology topics that have been covered in IAAI. Figure 3 also excludes information technology applications, that is, AI applied to our own business, which would otherwise be number one.

Another way of analyzing IAAI topics over the years is shown in figure 5. The figure shows that the technology mix has evolved. Expert systems clearly dominated the early days of IAAI. Machine learning is notably absent early on. Over time, however, the mixture has become more diverse, with no topic clearly dominating in recent conferences. We note that it is not the case that expert systems died. Rather, after a few years, they became more standard practice than innovation. Fewer papers were published about novel applications of expert systems. They disappeared into the fabric, now applied everywhere, from the high-end emulation of rare human experts, to the embedding and application of rule books and procedure manuals. However, we will likely see more hybrid machine-learning technologies that can automatically update their reasoning engines as the application data change over time.

Some technologies have not been represented much at IAAI, like speech understanding and robots. They do appear, just not in the top 15. In the recent 2016 deployed papers track, four technologies are applied: spatial reasoning, crowdsourcing, machine learning, and ontologies. We also note that there have been two papers on deep learning, one in 2015 and one in 2016, neither documenting a deployed application. Some of this may be due to self-selection in that our data are limited to IAAI conferences, which may not accurately reflect how often these technologies are utilized in the overall application world.

In our final analysis of IAAI articles, figure 6 shows a quick overview of the top concepts mentioned over the years. The analysis was done with a modified form of the C-value/NC-value method (Frantzi, Ananiadou, and Mima 2000), which extracts significant concept names found in text, as opposed to just the most frequently used phrases. Note that there may be some temporal bias in this analysis due to the data set reflecting the past decades of IAAI papers, versus trends in the most recent papers.

High-Impact AI Applications

Many of the past IAAI program chairs and cochairs and AAAI Fellows kindly responded to a request for their views on what have been the high-impact applications, including some that opened up a new area, presented at IAAI conferences over the years.

Because we have selected high-impact applications and it takes time to establish whether an application has had high impact, some of the examples may look a bit dated. Note, however, that in several cases, a recent update has been presented at IAAI.

A few of the applications that were singled out by several respondents as being high impact are summarized in the following.

1983: Process Diagnosis System (PDS)

The Process Diagnosis System (Fox, Lowenfeld, and Kleinosky 1983) started out as an expert system shell. It has been in active use and continuous development since 1985. 1985! Though the origin of PDS predates IAAI, it serves as an early example of deployed AI. It started with a presentation by Mark Fox at Westinghouse. Over the 30-year period, Westinghouse sold the business to Siemens, where it is now at the heart of their Power Diagnostics Center that performs centralized rule-based monitoring of over 1200 gas tur-

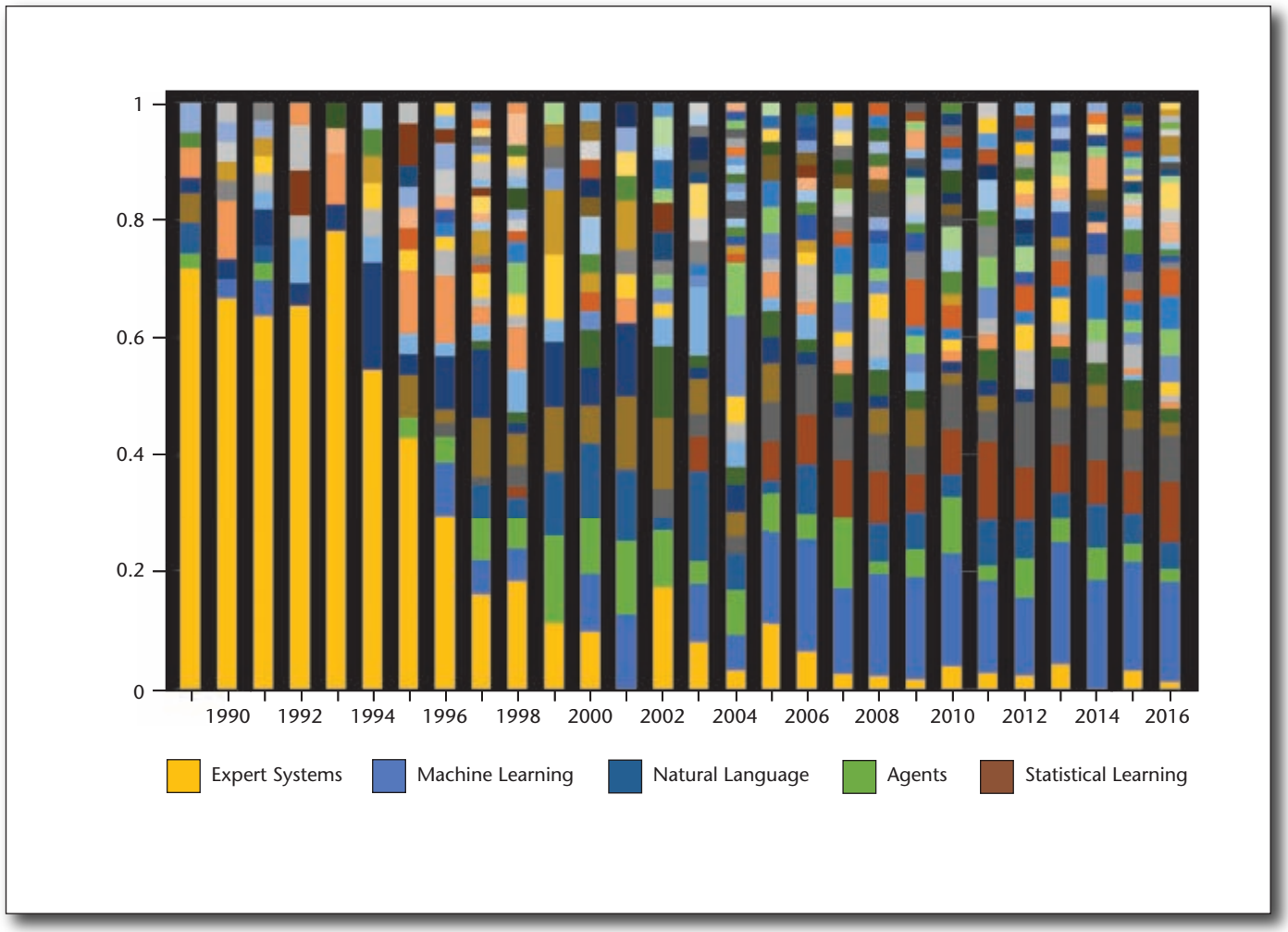


Figure 5. Mix of Technologies Deployed and Emerging in IAAI Articles, 1989–2016.

The dominant technologies include expert systems, machine learning, agents, natural language, and statistical learning, and are included in the figure legend.

bines, steam turbines, and generators. Ed Thompson and Ben Bassford celebrated the 30th anniversary of the system with the IAAI community when they presented an update at the 2015 conference in Austin (Thompson et al. 2015). Their paper summarizes the many changes that have been incorporated into the system over its lifetime, to deal with change in requirements, the customer business organization, and underlying computer technologies.

1989: Authorizer's Assistant

A knowledge-based credit-authorization system for American Express, the Authorizer's Assistant (Dzierzanowski et al. 1989) was the forerunner of now standard credit card transaction analysis. It created a capability that we all take for granted today — and complain about every time we are called to verify a charge — until at least we ourselves are the victims of fraud.

Expansion, improvement, and testing were planned from the start to ensure consistency as the knowledge base changed as well as ensure general system performance. The team found that consistency, audit tracking, and evaluation were key to acceptance and return on investment (ROI). They observed, "the [Authorizer's Assistant] proved to be better than all but the most expert credit card authorizers ... and that translated directly into huge ROI." The system's internal expert system incorporated 890 rules and ran on rack-mounted Symbolics Lisp machines connected to an IBM mainframe.

Phil Klahr generously provided these retrospective insights.

1989: Applications of Artificial Intelligence to Space Shuttle Mission Control

This NASA application originated in the Mission Control Center for STS-26 as a rule-based real-time

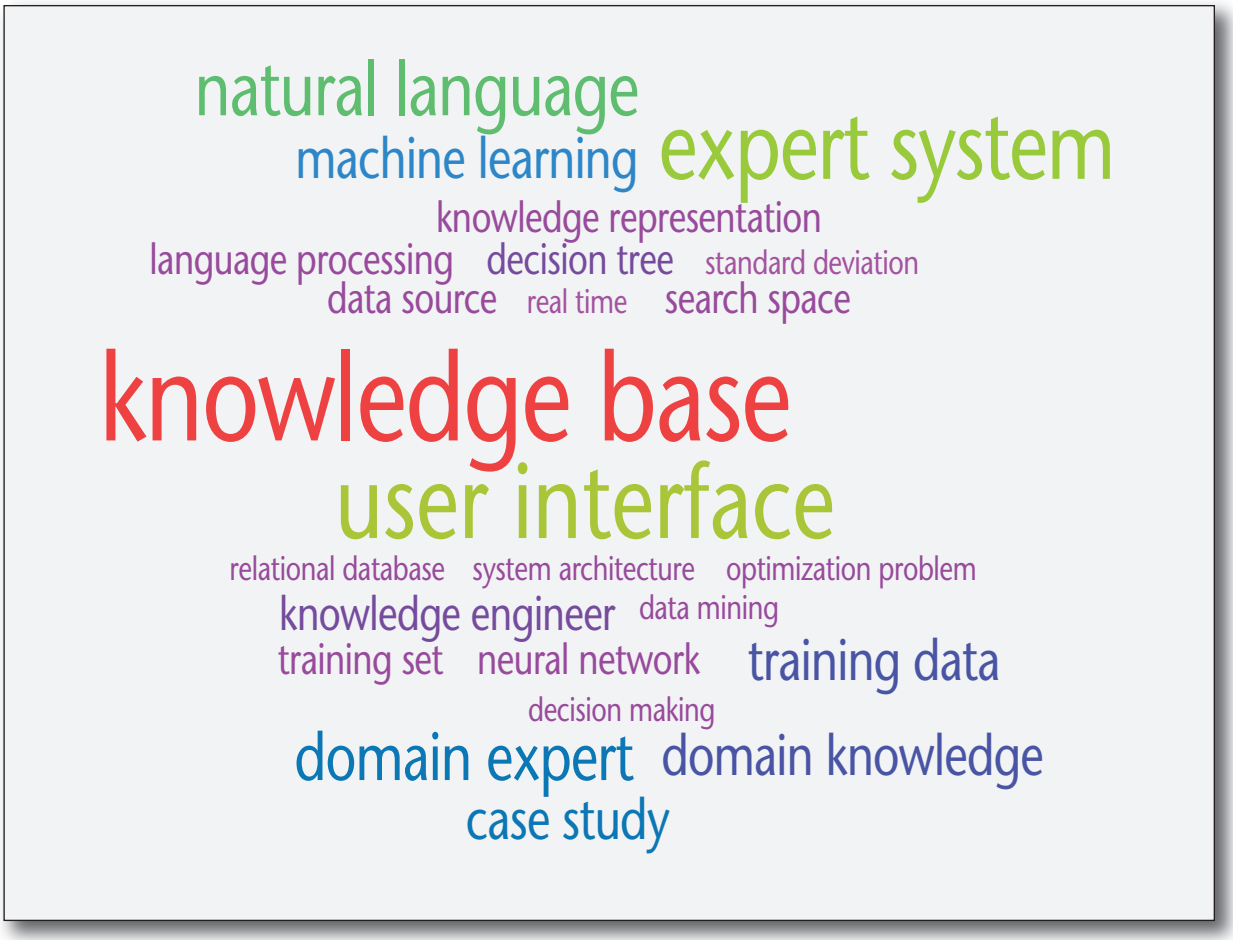


Figure 6. Top Concepts Mentioned in IAAI Papers, 1989–2016.

Integrated Communications Officer (INCO) expert system (Muratore et al. 1989). The system monitored space shuttle telemetry data and advised flight controllers on fault detection and diagnosis. It provided fault identification and diagnosis before the traditional INCO console could update the parameters of the faulty unit.

The system made use of the C Language Integrated Production System (CLIPS) Expert System Shell, now available as open source software.[2] It was also the first of more than 30 NASA applications reported to date at IAAI and, as is well known, AI systems later flew in space — and navigated autonomously on the moon and Mars.

The next two applications are intelligent assistants that played in center ring from first deployment. Both have been presented twice at IAAI — covering initial deployment and a 10-year update.

1989 and 1999: Ford Motor Company Direct Labor Management System

The Direct Labor Management System is integrated into Ford's Global Study Process Allocation System (GSPAS) (O'Brien et al. 1989, Rychtyckyj 1999). Its purpose is the automatic generation of work instructions for vehicle assembly, with associated times. It does so by analyzing high-level structured English descriptions. The system is also able to make accurate estimates of direct versus indirect labor time and to plan for mix/volume changes and line balancing. It has become an integral part of Ford's assembly process planning business.

The natural language component in this application was one of the few from the early days. The system was implemented on the NIKL/KL-ONE (Woods and Schmolze 1992) knowledge representation model — one of the first such applications.

Over the years, the system has undergone several knowledge base upgrades and ports to different platforms to keep the system viable and up to date through various organizational and business practice changes. It was later called the Global Study Process Allocation System.

1995: The FinCEN Artificial Intelligence System and 1999: The NASD Regulation Advanced-Detection System (ADS)

Created to identify potential money laundering from

reports of large cash transactions, the FinCEN Artificial Intelligence System (FAIS) used link diagrams to support detection of money laundering (Senator et al. 1995). ADS: NASD Regulation Advanced-Detection System (Kirkland et al. 1999) used temporal sequences to support detection of securities fraud. Their different domains of use dictated different knowledge representations.

FAIS links and evaluates reports of large cash transactions. To give an idea of the money involved, fincen.gov reported suspicious transactions totaling approximately $28 billion in October 2015. The FAIS key idea is "connecting the dots" — thus link diagrams, now commonplace in social network analysis, was an appropriate choice. The appropriate representation choice in FAIS enabled a reporting app based on the original detection system. This was an unanticipated bonus.

ADS monitors trades and quotations in the Nasdaq Stock Market, to identify suspicious patterns and practices. In this application, temporal sequences are key — not so much links as in FAIS — so a representation that supports them was a good choice.

2005 and 2014: Engineering Works Scheduling for Hong Kong's Rail Network

The Hong Kong rail network moves 5 million passengers a day through the city's rapid transit subway, airport express, and commuter rail lines. The AI application streamlines the planning, scheduling, and rescheduling process and provides automatic detection of potential conflicts as work requests are entered; verification that no conflicts exist in any approved work schedules before execution; generation and optimization of weekly operational schedules; automatic update to repair schedules after changes; and generation of quarterly schedules for planning (Chun et al. 2005, Chun and Suen 2014).

To be successful, the system must coordinate with the staff members who carry out the scheduled work. To this end, the developers found that the system must be able to explain the schedules it creates. As a result, they veered away from the original genetic algorithms approach toward heuristic search. A recent report about this system appeared in *New Scientist* (Hodson 2014).

1994 and 2004: Plastics Color Formulation Tool

Since 1994, GE Plastics (later SABIC) has employed a case-based reasoning (CBR) tool that determines color formulas that match requested colors (Cheetham 2004). FormTool has saved millions of dollars in productivity and material (that is, colorant) costs. It is the basis for the online color-selection service called ColorXpress Services.

Determining the colorants and loading levels that can be added so the plastic matches a given color is a difficult problem for multiple reasons. For example, there is no accurate method to predict the color produced when a set of colorants is added to plastic. Unlike paint, where light primarily reflects off the surface, in plastics a significant percentage of light penetrates the surface and reacts with the internal structure to produce a color that depends on both the internal structure and the lighting conditions (natural sunlight versus fluorescent lighting).

The AI system used case-based reasoning to replace programs that used prohibitively expensive exhaustive search to determine the colorant-loading proportions for a color formula that matches a customer's desired color.

1995: Scheduling of Port of Singapore Authority

This expert system (Weng et al. 1995) is responsible for assisting with planning and management of all operations of the Port of Singapore Authority. With hundreds of vessels calling at Singapore every day, a fast and efficient allocation of marine resources to assist the vessels in navigating in the port waters is essential. Manual planning using pen and paper was erroneous, uncoordinated, and slow in coping with the rapid increase in the vessel traffic. Included in the purview of the application is scheduling the movement of vessels through channels to terminals, deploying pilots to tugs and launches, allocating berths and anchorages to ships, and planning stowage of containers.

To generate accurate, executable deployment schedules, the automated scheduler requires real-time feedback from the resources on their job status, any estimated delays, and end times of their jobs. This is achieved by integrating the system with the port's mobile radio data terminal system.

2006: Expressive Commerce and Its Application to Sourcing

This application has produced one of the largest ROI figures of any system thus far reported at IAAI. Originally CombineNet, later renamed SciQuest, it improves procurement decisions for spend categories that are typically beyond the capabilities of traditional eSourcing software. Even in the early days of 2006, it had already handled $35 billion in auctions and delivered $4.4 billion in savings to customers through lower sourcing costs (Sandholm 2007).

The challenge in developing an expressive commerce system is handling the combinatorial explosion of possible allocations of businesses to suppliers. Their key development is a sophisticated tree search algorithm. Much has been written about this algorithm (refer to Sandholm [2007] for a list of articles), though some of its details are kept proprietary.

2014: CiteSeerX

CiteSeerX (Wu et al. 2014) is a database and search engine for more than 4 million research articles from

various disciplines. Starting in 1997 as CiteSeer, the service was the first to extract and index citations from documents automatically. Today, it is also capable of extracting metadata from individual paragraphs and sentences as well as tables and figures. The metadata and the original documents are made freely available for researchers who work on advanced information-retrieval algorithms.

The CiteSeerX service is accessed 2 million times per day and an average of 10 articles are downloaded per second. The size of the document database (after deduplication) has grown significantly over the years, from 500,000 in 2008 (when CiteSeerX debuted) to nearly 3 million in 2013. Today, between 50,000 and 100,000 PDFs are analyzed per day.

CiteSeerX's implementation makes use of several AI components, including document classification, duplicate detection, metadata extraction, author name disambiguation, and search indexing. The researchers' 25 years of experience is documented (Wu et al. 2014) and serves as a showcase of the variety of AI techniques available, for example, rule engines, neural networks, probabilistic graphical models, and the importance of choosing the Appropriate technique.

Lessons Learned

As the story of developing, implementing, and upgrading applications has unfolded, so has the conventional wisdom about building successful AI applications.

The Power Is in the Knowledge ... But Manual Knowledge Acquisition Is Hard

The first lesson learned by builders of early AI systems is that "the power is in the knowledge." By 1989, thanks to the pioneering efforts of Ed Feigenbaum, Bruce Buchanan, and many others, we understood that domain-specific knowledge (chemistry, medicine, and others) and task-specific knowledge (turbine maintenance, plant scheduling, and others) are more important for high performance and accurate reasoning than general problem-solving approaches.

But manual knowledge acquisition is hard and takes a long time. In the past, we called this the "knowledge acquisition bottleneck." Furthermore, ongoing knowledge base maintenance and curation are essential. Knowledge is perishable — everything changes over the lifetime of an application: the domain evolves, new use cases arise, new experts arrive with different knowledge, new data sets become available, the technology advances, and so on. If system builders are engaged in manual knowledge acquisition, then they will also need an army of people to keep the knowledge base up to date. Therefore, they need a lot of revenue (hence, a lot of users) to support that effort. Due to the difficulty of knowledge acquisition and maintenance, many systems fell by the wayside, even if they were excellent at one time. For example, the field of medicine is too large and changes too rapidly for manual knowledge acquisition (Myers, Pople, and Miller 1982).

Today, knowledge comes in a variety of forms. Early knowledge systems made use of symbolic rules that encoded experts' knowledge because such rules are compact, they are representationally adequate for many tasks and domains, and low-powered machines (by today's standards) are sufficient to perform the required inference procedures. However, rule-based expert systems are not effective for "big data" problems such as visual object recognition (for example, faces) and speech recognition. Progress in neural networks and deep learning, in particular, probabilistic graphical models and other machine-learning (ML) techniques, has greatly expanded the reach of AI systems. Yet, systems that use machine learning still use knowledge. Rather than expert-defined rules, ML systems make use of knowledge in several forms: training data including procedures for their acquisition and preprocessing, feature selection, model selection, and various parameters found by experimentation. It has been said, "there is no such thing as a free lunch," and in AI and ML, there is no such thing as free knowledge. There is no escape from the need to maintain and curate knowledge and data, even if some aspects are automated by machine learning.

Knowledge Representation Matters

The structure of knowledge in the system has a large impact on the system's reasoning capabilities and performance. The pioneers taught us that selecting the appropriate representation has a big impact, for five reasons: adequacy, efficiency, flexibility, maintainability, and explainability.

Adequacy

As John McCarthy and his colleagues stated, a system cannot reason about what it cannot represent (McCarthy 1960, 1981). Davis, Schrobe, and Szolovits (1993) referred to a knowledge representation as a "surrogate" for real-world entities. An adequate surrogate has a "correspondence" with real-world entities and these correspondences have high "fidelity," that is, they closely match the relevant characteristics of the real-world entities.

Efficiency

Every programmer knows that representations, or data structures, can have an impact on efficiency. For example, linked lists do not support quick random access while arrays do not support quick addition or deletion of elements. Similar trade-offs characterize knowledge representations. In general, the more simplistic the representation, the more efficient the reasoning algorithms. For example, reasoning over propositional logical is often quite efficient, while few tools exist that are capable of efficiently and reliably reasoning over first-order logic with types (Sut-

cliffe and Pelletier 2016). Likewise, nearest neighbor classification procedures require virtually no training while neural networks typically require significant training. During inference, however, a naïve nearest neighbor algorithm (for example, linear lookup) will likely be significantly slower than a neural network.

Flexibility

A good knowledge representation will support growth in the knowledge base that was not anticipated during initial knowledge acquisition. Additionally, long-running systems must be capable of evolving over time as the customer changes and the context in which the system was initially deployed migrates to other contexts and use cases. A good knowledge representation will be able to represent new knowledge concepts and adapt existing ones without significant updates to the representation or reasoning algorithms. The American Authorizer's Assistant, the FinCEN system, and the ADS systems are all good examples of this lesson in action.

In summarizing 10 years of work on Mycin, Buchanan and Shortliffe (1984) attributed the success of the program to flexibility — in the rule-based representation that allowed rapid modification and in the reasoning that allowed reaching reasonable conclusions with imperfect or missing information.

Maintainability

It is helpful if the knowledge representation can be understood and modified by subject matter experts, who may not be (and typically are not) experts in computer programming and knowledge representation. Ideally, software engineers will not need to be called in whenever the knowledge base needs an update. As John McCarthy noted, declarative representations are more learnable and maintainable than procedural ones (McCarthy 1960). The reason is that declarative representations are better separated from internal conventions of the reasoning algorithms, thus allowing subject matter experts to focus on the knowledge being represented.

Today, it is particularly important for subject matter experts to be able to interpret and modify the results of machine-learning systems that are driven primarily by raw data and empirical validation. They may, for example, suggest that there are critical data sets missing that would improve the analysis, and avoid simply handing the problem over to data scientists.

Explainability

In organizations, the AI system encodes and represents the decision criteria of the management. Thus, when the AI system suggests a decision, it should be able to explain that decision to the user so that the user (and the management) can "own it and be able to defend it" in terms of the organization's decision criteria. Explanation may be unnecessary if the algorithm makes money, for example, Wall Street trading. But in many other contexts, without an explanation system, organizational and user acceptance of AI applications is more challenging. This has been understood since SHRLDU in the Blocks World (Winograd 1972), and Mycin, in the knowledge-intensive world of medicine (Buchanan and Shortliffe 1984).

Explainability is more problematic in the case of noninterpretable models such as neural networks, in which it is not at all clear exactly what knowledge has been stored as a result of training.

Lately, there has been some discussion in the press about "algorithmic accountability" (Diakopoulos 2013, Lohr 2015), and several companies are pursuing explanation as a differentiator, for example, Watson Paths and the Narrative Science extension for the Qlik visual analytics tool (Hammond 2015).

Separate the Knowledge Base and the Inference Engine

As a corollary of the maintainability of declarative representations, the pioneers also taught us that it is a good idea to separate the knowledge base and the inference engine because a separate knowledge base is easier to change, update, debug, and explain. Recognizing the importance of separation, from a knowledge representation and a knowledge delivery perspective, many people have devoted their time to the development of expert system shells (for example, M1, S1, ART and CLIPS), knowledge representation languages (for example, KL-ONE and OWL), ontology editors (for example, Protégé), and general-purpose machine-learning models.

Successful Applications Incorporate a Variety of Techniques

Successful AI applications incorporate a wide range of techniques, strategies, and knowledge, embodying rules, objects, ontologies, statistics, and signal processing to name a few. Self-driving cars are an obvious example. Their capabilities include modeling, simulation, sensing, motion planning, object recognition, obstacle avoidance, machine learning, error recovery, and so on. The learnings have been reported multiple times at AAAI (Montemerlo et al. 2006, Thrun 2006) and IAAI (Urmson et al. 2009).

Modern text-analytics systems also illustrate the point. For example, the i2k Connect platform uses a variety of knowledge and AI techniques to perform document reading and enrichment. It uses ontologies to represent domain-specific knowledge about, for example, the oil and gas industry, the field of artificial intelligence, and topics related to supply-chain management and health care. Document text and metadata are extracted using machine-learning methods. Visual and language rules are used to extract the document's title and summary. Documents are then analyzed with a variety of rules in order to identify the domain-specific topics that the document is about. Multiple technologies from AI and elsewhere are needed for this processing pipeline.

A combination of various kinds of knowledge and techniques should be expected in any large-scale AI

application that is required to integrate multiple sources and types of information. The architecture of such an AI application should make such integration feasible by, for example, separating different processing tasks into distinct modules and supporting a common interface for communication among the components. The Robot Operating System[3] is a paradigmatic example of such an architecture. Different robots may have vastly different components and purposes, yet ROS offers high-level abstractions that enable various sensors, actuators, and algorithms to communicate using a common language.

AI Applications Must Integrate into Existing Work Flows

Perhaps the most important lesson learned by AI system builders is that success depends on integrating into existing workflows — the human context of actual use. It is rare to replace an existing work flow completely. Thus, the application must play nicely with the other tools that people use. Put another way, ease of use delivered by the human interface is the "license to operate." Unless designers get that part right, people may not ever see the AI power under the hood; they will have already walked away.

As AI systems began to function well enough that they were able to play in the center ring, so to speak, risk mitigation, project management, and budgetary control became more important. The systems were no longer in a "research" or "proof of concept" phase. In other words, standard IT rules — and consumer mobile app acceptance rules — apply. Many AI practitioners have made these points in the context of AI applications in particular. But the rules are valid for all applications of information technology.

In the early days, we talked as if AI systems had a big box of AI — the important stuff — and a small box of all that other messy IT stuff. We quickly learned that in real-world systems, it was mostly the other way around. The AI was a piece of the puzzle, and sometimes not a very big piece.

Consider the Dipmeter Advisor (Smith and Baker 1983), started at Schlumberger in the early 1980s and based on the knowledge of the legendary oil finder, Al Gilreath, shown in figure 7. The Dipmeter Advisor demonstrated the challenges of infrastructure: getting the data from the field systems was a bigger problem than originally anticipated; and the challenges of technology transfer: nontraditional hardware (D-Machines) and software (Interlisp-D) became major stumbling blocks, though without these technologies Schlumberger would have had no system at all.

The amount of effort that had to be devoted to the non-AI components was dominant. The user interface accounted for almost half the code. The rule engine and knowledge base accounted for 30 percent. Of course, lines of code do not necessarily tell the whole story, but the numbers are consistent with the development effort expended. Much of the coding effort went into the interactive graphics system, not the AI. For some clients, interactive graphics was the most important element.

Security and privacy have become increasingly crucial over time, and the application's performance characteristics in the deployed setting must meet industry or consumer expectations.

Additionally, change management is unavoidable (Hiatt 2006). But the amount of change management required is inversely proportional to the power of the new technology. It is also directly proportional to the amount of change in existing work flows required to adopt it.

Convincing people to make substantial changes to their existing work flows to take advantage of a new technology that isn't much better than the old technology requires a great deal of change management effort. On the other hand, convincing people to make small changes to their existing work flows to take advantage of new technology that is an order of magnitude better than the old technology requires only modest change management effort.

As Mehmet Goker put it in a private communication to the authors: "Applications with a small and flexible core that solve a real-world problem have the biggest impact and are the easiest to put into the workplace."

To summarize, in any large organization, standard IT rules apply and the AI application should fit into the broader IT infrastructure to ensure successful adoption. Management, end user, and IT support and participation are essential. Budget approval will be challenging without business unit management support, deployment into a company's existing infrastructure is not possible without support from the IT organization, and adoption is unlikely without continuous end-user participation in system development.

In the real world of applications, our experience also suggests that the dichotomy suggested by Markoff (2015) between artificial intelligence and intelligence augmentation or amplification does not exist. They are two ends of a spectrum that meet in most applications. The successful systems enable people to do what people do best and use computers to do what computers do best.

A Way Around the Knowledge Acquisition Bottleneck

Machine learning offers a way around the knowledge acquisition bottleneck ... but success depends on human insight folded into the methods, like the choice of features.

One thing has not changed over the history of IAAI. It is still very hard to build, curate, and maintain large knowledge bases by hand. The manual knowledge-acquisition bottleneck is still firmly in place.

Aside: This is a special case of a larger point. Manual information governance is not sustainable. Very few

Figure 7. The Dipmeter Advisor System.

humans have the passion and consistency to tag and manage their own unstructured data ... look at your own hard drive or your organization's file shares if you doubt it. This is one of the main reasons why so much unstructured corporate data is "lost in the cloud." It may be there, but you are likely to struggle to find it if you didn't write it yourself. More than half of employees in companies surveyed worldwide express deep dissatisfaction with the findability of corporate information.[4] In contrast to Internet content, today it is rare to see search engine optimization applied to intranet content.

But now armed with billions of crowdsourced examples from the web, we have learned that data-driven, statistical methods are "unreasonably effective" in several domains. The statistics bring the ability to deal with noise and to cover problems where humans either have difficulty explaining how they do it, or where they don't do it very well in the first place.

The bottom line is that machine learning is a way around the knowledge-acquisition bottleneck in a surprisingly broad number of domains, but two caveats are worth considering:

Howie Shrobe made an observation that rings true. "... when you look closer at successful statistical approaches, a lot of the success is in the choice of features to attend to or other similar ways of conveying human insight to the technique ..." (private communication). Indeed, mitigating this problem is a focus of some research on deep learning algorithms — to learn feature representations from unlabeled data.[5]

There is a very long tail on the types of problems encountered in the world. Developers will not have millions of examples for all of them. In those cases, some kind of reasoning is essential; for example, from basic principles captured via case-based reasoning or encoded in a rule-based system.

Apps Can Be Built with Components That Reason from Different Starting Points.

In the early days of expert systems running on machines with relatively little processing power and memory, the standard starting point for delivering domain and task-specific knowledge can be characterized by labels like *slow, cognition, search, top-down, model-driven.*

Today, armed with the compute power, data, and machine-learning algorithms now available to us, we are much better equipped to build apps that reason

from a starting point characterized by labels like *fast, recognition, look-up, bottom-up, data-driven*.

For example, *Fast versus Slow.* The focus of Daniel Kahneman's 2011 book is a dichotomy between these two modes of thought: "fast, instinctive, and emotional" and "slow, more deliberative, and more logical" (Kahneman 2011).

Alternatively, Herb Simon put it this way: "The situation has provided a cue; this cue has given the expert access to information stored in memory, and the information provides the answer. Intuition is nothing more and nothing less than recognition." (Simon 1992). *Fast* corresponds to *recognition*. *Slow* corresponds to *cognition* or *search*. In this regard, compare the recognition approach of the human chess master to the search approach of Deep Blue (Campbell, Hoane, and Hsu 1999). (Because of this, a typical grandmaster does six orders of magnitude less search per move than Deep Blue did.)

Another example, well known to American football fans, is that of the Manning brothers, Peyton and Eli. It has been widely reported that their father Archie started the boys learning football and quarterbacking at the earliest possible age. This maximized the time they had to store millions of the small chunks of recognition knowledge, later buttressed by countless hours spent studying game film.

Rod Brooks championed what he called a new approach to artificial intelligence and robot design — which can be called "bottom-up" — as an alternative to the "top-down" model-driven approach of the pioneers (Brooks 1991).

Today, some authors seem to see a conflict between "data-driven" (new think) systems and "model-driven" (old think) systems as if the "good" applications today are all data driven and work well, in contrast with the "bad" model-driven applications of the old days that didn't work well.

Many AI apps have combined reasoning from opposite starting points, going way back to the early days. The Hearsay II speech-understanding system combined top-down and bottom-up processing (Erman et al. 1980). Mycin used backward and forward rule chaining (Buchanan and Shortliffe 1984). And the Dipmeter Advisor was both data driven, converting raw signals to patterns, and model driven, using rules to classify stratigraphic and tectonic features from the patterns (Smith and Baker 1983). Overall accuracy depended on the contributions of all the components — data driven and model driven.

We also don't accept the criticism that the early AI community was too focused on model-driven approaches when it should have been focused on data-driven approaches. We believe the pioneers were doing the best they could with the machines and data available to them. They were forced into cognitive approaches in some cases (for example, vision) because they had to do something to finesse the need for orders of magnitude more processing power, storage, and sensors than were available to them in the day.

The good news these days is that all the components are substantially more powerful, thanks to the computing and data revolutions. We are not restricted to either a "fast" or a "slow" starting point. We can have both.

That said, it is important for developers to give due consideration to the new possibilities offered by the substantial increases in processor speed and memory available today — and to not implicitly be stuck in the "slow" thinking mind set of the early days. Going forward, there is the possibility of storing massively larger knowledge bases that are composed of small chunks of very specific domain and task knowledge, retrieved by fast recognition processes (more of what Simon was referring to).

Thus, a knowledge base for a domain would have powerful rules (as in the past, thousands of them) plus these small chunks of very specific experiential knowledge (millions of them). With modern sensors, the small chunks may be very easy to capture. Certainly, there will be things missing that might have been implied by rules (that is, not everything possible is actually observed and remembered as a chunk). But overall, knowledge acquisition will have become far easier to do and cheaper. These "hybrid" knowledge base architectures will dominate in applications. This also seems like a fruitful avenue for reconsidering older models of human cognition. (The authors thank Ed Feigenbaum for this observation.)

Checklist for Tomorrow's Application Builders

Our examination of nearly 30 IAAI conferences, our personal experiences, and stories related to us by colleagues and friends, lead to the checklist in table 3. We briefly explain each entry in the following.

As will be apparent to experienced application developers, much of this advice mirrors general software engineering best practice. But some of the points are even more important for AI systems. We invite your feedback and your own lessons learned.

Select Problems with a Solid Business Case

Successful IT applications in general start with a focus on the business case and the customer — not the technology. This is particularly true for AI applications. In the early days of AI applications, the mind share of the developers tended more heavily to the technology (the knowledge-representation methods and the reasoning machinery) than it did to the customer need. In retrospect, this was to be expected. The early implementers were almost always AI researchers, infringing on an IT community that was by and large skeptical of the hype and the baggage that came along with the technology — nonstandard hardware and software, methods that were not understood by the

■ Select problems with a solid business case.
■ Minimize changes required in existing work flows.
■ Identify domain- or task-specific knowledge and data for the problem.
■ Select appropriate knowledge representations and data sources.
■ Develop knowledge and data acquisition and maintenance plans.
■ Select appropriate reasoning/learning strategies.
■ Develop a set of test cases and performance metrics.
■ Add safeguards and opt-out capabilities.
■ Test with real data from users or operating environment.

Table 3. Checklist for Builders of AI Applications.

community, the need to bring in outside experts.

Over the years, AI application developers have made a major mind shift. We have learned the hard way that success starts with solving problems important to the customer.

One caveat: Although public interest in AI is on the rise, do not add an AI component to an application just for the sake of it. AI can introduce complexity, and systems should always only be as complicated as is necessary to model the domain and task. Again, focus on customer over technology.

Minimize Changes Required in Existing Work Flows

Think about the integration of AI with other tools and parts of the larger system. It is rare to completely replace an existing work flow. Thus, it is prudent to build new systems so that they can slot into the approaches already used by the customers as much as possible. Few new AI systems solve stand-alone problems that require no user interaction. Most are used as "intelligent assistants" and the amount of change management required to succeed in adoption is directly proportional to the magnitude of the changes required in existing work flows. Ease of use is the "license to operate."

Identify Domain- or Task-Specific Knowledge and Data for the Problem

The history of successful AI applications shows that "the power is in the knowledge," both expert-provided knowledge and knowledge extracted out of data with the appropriate preprocessing, feature selection, learning techniques, and parameter tuning. Devote up front the effort needed to acquire enough of the knowledge and data so that you will better understand how to design the rest of the system to best match the domain and task.

Select Appropriate Knowledge Representations and Data Sources

Depending on the nature of the domain- and task-specific knowledge, choose a knowledge representation that most closely models the world while still supporting efficient reasoning strategies. Prefer declarative knowledge since it is easier to understand, explain and change than procedural knowledge. For machine-learning approaches, select high-quality data sources (for example, data with expert-verified ground truth) when feasible or develop a strategy for learning and reasoning with noisy data sources.

Develop Knowledge and Data Acquisition and Maintenance Plans

Consider knowledge/data acquisition and maintenance to be an ongoing process. Make the process iterative: repeatedly evaluate if the knowledge and data are appropriate for the reasoning/learning strategies and domain/task and refine accordingly.

Select Appropriate Reasoning and Learning Strategies

Most large AI systems will require various kinds of reasoning and learning strategies for various subproblems. Design a system architecture that supports decoupling of these disparate components so that refinements in one component will not require drastic changes in other components.

Depending on the constraints dictated by the domain and task, select an approach and components that are data driven or model driven, or use a combination.

Develop a Set of Test Cases and Performance Metrics

Due to the complexity of most AI systems, testing and performance evaluation are critical.

The word *performance* encompasses a variety of concerns, including run-time speed and use of resources plus adequacy of the knowledge and reasoning components. Run-time speed and use of resources are standard computational concerns that must be addressed in any system that is to be deployed and scaled. They are not specific to AI appli-

cations — but adequacy of the knowledge and reasoning components is specific to AI applications.

Sometimes the desired behavior of the system is clear, as was the case when IBM was developing Watson to win the *Jeopardy!* game show against human contestants (Ferrucci et al. 2010). Keep track of the performance of every revision and consider a policy (as IBM did with Watson) that rejects any revisions that do not push performance closer to the goal.

When the goal criteria are not as clear, make extensive use of regression tests to ensure that solved cases are never broken in the future. Sometimes, even regression tests are too precise as multiple different outcomes may be equally good. A good technique in these situations is to build machinery to automatically identify any changes in the system's output after each code or knowledge base revision. Knowing what changed after an update is a valuable first step in identifying if development is on the right track.

Add Safeguards and Opt-Out Capabilities

AI has been known, on occasion, to produce odd and unpredictable results due to complex reasoning systems, large data sets, and large knowledge bases. Hence, special care should be taken to verify data produced by AI subsystems. In addition, there is a premium on testing carefully for machine-learning systems that do not have transparent reasoning processes.

This advice should be heeded more diligently for builders of AI applications that make use of human input and applications that are responsible for making decisions for users. For that matter, such applications should provide an opt-out capability that lets the user complete an action without AI assistance. An AI system is even stronger when it can explain its decisions and can help users make sense of the AI's assistance and better decide if they prefer to continue making use of it.

Test with Real Data from Users or Operating Environment

At i2k Connect, we have learned that there is a long tail of the kinds of documents humans (and computers) produce and that may be fed into our document enrichment service. During the early development effort, we focused on straightforward cases such as research articles in PDF form and Microsoft Word documents made up mostly of text. However, real data from real users can be drastically different and highly variable. For example, we learned that our system did not properly handle large text files produced by computer software (such as log files or data dumps), and needed extra logic to examine each file before deciding what kinds of processing would be appropriate. In other examples, roboticists know that robots must be tested in the real world and not just simulations, and developers of personal assistants, chatbots, search engines, and other tools know that humans are an unpredictable source of a wide range of inputs.

Conclusion

For AI to benefit humankind it must be deployed; for successful deployment, good AI ideas must be integrated into the human context of actual use and into the IT context of organizations. In this article, we have tried to summarize what has been learned about building, maintaining, and extending AI applications. We have boiled it down into a simple checklist for the developers of today and tomorrow.

Going forward, we can expect the landscape of AI applications to continue to diversify and expand. The revolutions will continue all around us, in computers and data, as well as sensing.

So, it follows that apps will continue to get more powerful, more knowledgeable, and cover a broader array of domains and tasks.

It also follows that apps will be increasingly data driven, guided by human knowledge. And they will have a lot more data available, as the Internet of Things takes off.

Finally, intelligent assistants will be even more proficient at improving quality of life. The partnership between human and machine is going to be stronger and closer. How will this improve quality of life? Jobs tend to be more satisfying when we humans are able to focus on the *real* work we set out to do, not distracted by the low-level clutter that most people are forced to deal with today, because computers aren't powerful enough, or because no attempt has yet been made to automate the jobs people don't want to do. Intelligent assistants will deal with the clutter of low-level tasks, or tasks that require extended concentration, consistency, scale, and so on.

As an example, we see big opportunities with unstructured data. It will no longer be lost in the cloud — whether the corporate cloud or the Internet cloud. We will have the tools to find it and unlock its connections. We will also have the tools to extract the essential information from the cluttered real-time data streams that overwhelm us today.

As the developers of today and tomorrow address the new opportunities, the history of IAAI conferences offers lessons in how to build successful deployed AI applications. We have attempted to distill these lessons to increase the chances of future success. In these concluding remarks, we have just a few final bits of advice.

It is prudent for AI researchers to pay attention to what is being learned through engineering practice — deployed applications — as was hoped for from the beginning of IAAI. And it is prudent for practitioners to take advantage of opportunities to learn from research, as was hoped for by colocating the AAAI and IAAI conferences, and by adding the Emerging Applications track to the IAAI conference in 1997.

It is also wise to pay attention to what is happening in the rest of the computing, data, and sensing world. Factors external to AI are likely to have the largest impact on what matters, or what is possible, or where opportunities lie. The biggest impact on how we are able to build applications today has come from revolutions that were not of our own making. Watch for signals from the periphery.

And finally, to quote Neil Jacobstein, "AI expands the range of the possible." So keep doing it!

Acknowledgments

In preparing this article, we have drawn

heavily from AI applications observations made by others. We encourage readers to look back at Feigenbaum, McCorduck, and Nii (1988); Feigenbaum (1993); Shrobe (1996); Shrobe (2000;) and Jacobstein (2007).

Thanks to the following people for generously sharing their time and their ideas on high-impact applications and for their observations on what (we thought) we knew then and what (we think) we know now: Bruce Buchanan, Andy Chun, Edward Feigenbaum, Markus Fromherz, Ashok Goel, Mehmet Goker, Haym Hirsh, Neil Jacobstein, Phil Klahr, Alain Rappaport, Nestor Rychtyckyj, Eric Schoen, Ted Senator, Howard Shrobe, David Stracuzzi, Ramasamy Uthurusamy, and Peter Yeh.

Thanks especially to Bruce Buchanan and Ed Feigenbaum for years of guidance ... and patience.

Thanks also to Jon Glick, AITopics pioneer, and to Raj Reddy, godfather of IAAI.

Finally, thanks to our families for their foundational contributions.

This article is based on the IAAI-16 Robert S. Engelmore Memorial Lecture given by the first author at AAAI/IAAI 2016 in honor of Bob Engelmore's extraordinary service to AAAI and his contributions to applied AI.[6]

Notes

1. See Dennis Bode, The Ivy Bridge Test: Intel Core i7-3770K and all i5 models (Hardware LUXX), available at www.hardwareluxx.com/index.php/reviews/hardware/cpu/21569-ivy-bridge-test-intel-core-i7-3770k-and-all-i5-models.html?start=13.

2. For more information about CLIPS see www.clipsrules.net/?q=AboutCLIPS.

3. The Enterprise Search and Findability Survey 2014, by Carl Björnfors and Mattias Ellison, is available at www2.findwise.com/findabilitysurvey2014.

4. Robot Operating System (ROS) is available at www.ros.org.

5. deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial.

6. Available at www.reidgsmith.com/2016-02-15_Engelmore_Lecture.pdf

References

Brooks, R. A. 1991. Intelligence Without Representation. *Artificial Intelligence* 47(1–3): 139–159.

Buchanan, B. G., and Shortliffe, E. H. 1984. *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project* (The Addison-Wesley Series in Artificial Intelligence). Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Campbell, M.; Hoane, J.; and Hsu, F. 1999. Search Control Methods in Deep Blue. In *Search Techniques for Problem Solving Under Uncertainty and Incomplete Information: Papers from the 1999 AAAI Spring Symposium*. Technical Report SS-99-07. Menlo Park, CA: AAAI Press.

Cheetham, W. E. 2004. Tenth Anniversary of the Plastics Color Formulation Tool. In *Proceedings of the Sixteenth Conference on Innovative Applications of Artificial Intelligence,* 770–776. Menlo Park, CA: AAAI Press.

Cheyer, A. 2014. Siri: Back to the Future. Invited presentation at the Twenty-Sixth IAAI Conference / Twenty-Eighth AAAI Conference, Montreal, Quebec, Canada, July.

Chun, A. H. W., and Suen, T. Y. Y. 2014. Engineering Works Scheduling for Hong Kong's Rail Network. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence and the Twenty-Sixth Annual Conference on Innovative Applications of Artificial Intelligence,* 2890–2897. Palo Alto, CA: AAAI Press.

Chun, A. H. W.; Yeung, D. W. M.; Lam, G. P. S.; Lai, D.; Keefe, R.; Lam, J.; and Chan, H. 2005. Scheduling Engineering Works for the MTR Corporation in Hong Kong. In *Proceedings of Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference,* 1467–1474. Menlo Park, CA: AAAI Press.

Clancy, P.; Gerald, H.; and Arnold, S. 1989. An Expert System for Legal Consultation. In *Proceedings of the First Annual Conference on Innovative Applications of Artificial Intelligence,* 125–135. Menlo Park, CA: AAAI Press.

Davis, R.; Shrobe, H.; and Szolovits, P. 1993. What Is a Knowledge Representation? *AI Magazine* 14(1): 17–33.

Diakopoulos, N. 2013. Rage Against the Algorithms: How Can We Know the Biases of a Piece of Software? By Reverse Engineering It, of Course. *The Atlantic*. 3 October. Available at www.theatlantic.com/technology/archive/2013/10/rage-against-the-algorithms/280255

Dzierzanowski, J. M.; Chrisman, K. R.; MacKinnon, G. J.; and Klahr, P. 1989. The Authorizer's Assistant: A Knowledge-Based Credit Authorization System for American Express. In *Proceedings of the First Conference on Innovative Applications of AI,* 168–172. Menlo Park, CA: AAAI Press.

Eckroth, J.; Dong, L.; Smith, R. G.; and Buchanan, B. G. 2012. NewsFinder: Automating an Artificial Intelligence News Service. *AI Magazine* 33(2): 43-54.

Erman, L.; Hayes-Roth, F.; Lesser, V. R.; and Reddy, D. R. 1980. The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. *ACM Computing Surveys* 12(2): 213–253. doi.org/10.1145/356810.356816

Feigenbaum, E.; McCorduck, P.; and Nii, P. 1988. *The Rise of the Expert Company*. New York: Times Books, Random House.

Feigenbaum, E. A. 1993. Tiger in a Cage: The Applications of Knowledge-Based Systems. Joint invited presentation at the 11th National Conference on Artificial Intelligence and the Fifth Conference on Innovative Applications of Artificial Intelligence, Washington, DC, July.

Ferrucci, D.; Brown, E.; Chu-Carroll, J.; Fan, J.; Gondek, D.; Kalyanpur, A. A.; Lally, A.; Murdock, J. W.; Nyberg, E.; Prager, J.; Schlaefer, N.; and Welty, C. 2010. Building Watson: An Overview of the DeepQA Project. *AI Magazine* 31(3): 59–79.

Fox, M. S.; Lowenfeld, S.; and Kleinosky, P. 1983. Techniques for Sensor-Based Diagnosis. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence,* 159–163. Los Altos, CA: William Kaufmann, Inc.

Frantzi, K.; Ananiadou, S.; and Mima, H. 2000. Automatic Recognition of Multi-Word Terms: The C-Value/NC-Value Method. *International Journal on Digital Libraries* 3(2): 115–130.

Hammond, K. 2015. How Artificial Intelligence Explains Analytics. *Computerworld,* June 19. Available at www.computerworld.com/article/2936142/emerging-technology/ how-artificial-intelligence-explains-analytics.html.

Halevy, A.; Norvig, P.; and Pereira, F. 2009. The Unreasonable Effectiveness of Data. *Intelligent Systems* 24(2): 8–12.

Hiatt, J. 2006. *ADKAR: A Model for Change in Business, Government and Our Community*. Fort Collins, CO: Prosci.

Hodson, H. 2014. The AI Boss That Deploys Hong Kong's Subway Engineers. *Technology News,* 2 July. Available at www.newscientist.com/article/mg22329764.000-the-ai-boss-that-deploys-hong-kongs-subway-engineers.

Jacobstein, N. 2007. Innovative Applications of Early Stage AI. Presented at the 2007 Singularity Summit, San Francisco, CA, September. Available at foresight.org/cms/ events_library/267.

Kahneman, D. 2011. *Thinking, Fast and Slow.* New York: Farrar, Straus and Giroux.

Kirkland, J. D.; Senator, T. E.; Hayden, J. J.; Dybala, T.; Goldberg, H. G.; and Shyr, P. 1999. The NASD Regulation Advanced-Detection System (ADS). *AI Magazine* 20(1): 55–67.

Lohr, S. 2015. If Algorithms Know All, How Much Should Humans Help? *New York Times,* April 6. Available at www.nytimes.com/2015/04/07/upshot/if-algorithms-know-all-how-much-should-humans-help.html.

Markoff, J. 2015. *Machines of Loving Grace: The Quest for Common Ground Between Humans and Robots*. New York: Ecco, an imprint of HarperCollins Publishers.

McCarthy, J. 1981. Epistemological Problems of Artificial Intelligence. In *Readings in Artificial Intelligence,* ed. B. Webber and N. Nilsson, 459–465. Los Altos, CA: Tioga.

McCarthy, J. 1960. Programs with Common Sense. In *Proceedings of the Symposium on the Mechanization of Thought Processes,* National Physical Laboratory 1: 77–84. Teddington, UK: Her Majesty's Stationery Office. Available at aitopics.org/publication/programs-common-sense.

Miller, R. A.; Pople, Jr, H. E.; and Myers, J. D. 1982. Internist-I, An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine. *New England Journal of Medicine* 307(8): 468–476.

Minsky, M. 1961. Steps Toward Artificial Intelligence. Proceedings of the IRE, 49(1), 8–30. Reprinted in *Computers & Thought,* ed. E. A. Feigenbaum and J. Feldman, 406–450. Cambridge, MA: The MIT Press.

Montemerlo, M.; Thrun, S.; Dahlkamp, H.; Stavens, D.; and Strohband, S. 2006. Winning the DARPA Grand Challenge with an AI Robot. In *Proceedings of the National Conference on Artificial Intelligence* 21(1): 982–987. Menlo Park, CA: AAAI Press.

Muratore, J. F.; Heindel, T. A.; Murphy, T. B.; Rasmussen, A. N.; and McFarland, R. Z. 1989. Applications of Artificial Intelligence to Space Shuttle Mission Control. In *Proceedings of the Conference on Innovative Applications of Artificial Intelligence,* 15–22. Menlo Park, CA: AAAI Press.

Myers, J. D.; Pople, H. E.; and Miller, R. A. 1982. CADUCEUS: A Computerized Diagnostic Consultation System in Internal Medicine. In *Proceedings, The Sixth Annual Symposium on Computer Applications in Medical Care,* ed. B. I. Blum, 44–47. Los Alamitos, CA: IEEE Computer Society.

O'Brien, J.; Brice, H.; Hatfield, S.; Johnson, W.; Woodhead, R. 1989. The Ford Motor Company Direct Labor Management System. In *Innovative Applications of Artificial Intelligence,* 331–346. Cambridge, MA: AAAI Press / The MIT Press.

Preshing, J. 2012. A Look Back at Single-Threaded CPU Performance. Preshing on Programming Blog, February 8. Available at preshing.com/20120208/a-look-back-at-single-threaded-cpu-performance/.

Rychtyckyj, N. 1999. DLMS: Ten Years of AI for Vehicle Assembly Process Planning. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence,* 821–828. Menlo Park, CA: AAAI Press.

Sandholm, T. 2007. Expressive Commerce and Its Application to Sourcing: How We Conducted $35 Billion of Generalized Combinatorial Auctions. *AI Magazine* 28(3): 45–58.

Schorr, H., and Rappaport, A. T., eds. 1989. *Innovative Applications of Artificial Intelligence*. Cambridge, MA: AAAI Press / The MIT Press.

Senator, T. E.; Goldberg, H. G.; Wooton, J.; Cottini, M. A.; Khan, A. U.; Klinger, C. D.; and Wong, R. W. 1995. The FinCEN Artificial Intelligence System: Identifying Potential Money Laundering from Reports of Large Cash Transactions. In *Proceedings of the 7th Annual Conference on Innovative Applications of Artificial Intelligence,* 156–170. Menlo Park, CA: AAAI Press

Shrobe, H. E. 1996. The Innovative Applications of Artificial Intelligence Conference: Past and Future. *AI Magazine* 17(4): 15–20.

Shrobe, H. E. 2000. What Does the Future Hold? *AI Magazine* 21(4): 41–57.

Simon, H. 1992. What Is an "Explanation" of Behavior? *Psychological Science* 3(3): 150–161.

Smith, R. G., and Baker, J. D. 1983. The Dipmeter Advisor System: A Case Study in Commercial Expert System Development. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*-Volume 1, 122–129. Los Altos, CA: William Kaufmann, Inc.

Sutcliffe, G., and Pelletier, F. J. 2016. Hoping for the Truth — A Survey of the TPTP Logics. In *Proceedings of the Twenty-Ninth International Flairs Conference,* 110–115. Palo Alto, CA: AAAI Press.

Thibodeau, P. 2008. Computer Science Graduating Class of 2007 Smallest This Decade. *Computerworld* Online. March 5. Available at www.computerworld.com/article/2537436/it-careers/computer-science-graduating-class-of-2007-smallest-this-decade.html.

Thompson, E. D.; Frolich, E.; Bellows, J. C.; Bassford, B. E.; Skiko, E. J.; and Fox, M. S. 2015. Process Diagnosis System (PDS) — A 30 Year History. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence and the 27th Annual Conference on Innovative Applications of Artificial Intelligence,* 3928–3933. Palo Alto, CA: AAAI Press.

Thrun, S. 2006. A Personal Account of the Development of Stanley, the Robot That Won the DARPA Grand Challenge. *AI Magazine* 27(4): 69–82.

Urmson, C.; Baker, C.; Dolan, J.; Rybski, P.; Salesky, B.; Whittaker, W.; and Darms, M. 2009. Autonomous Driving in Traffic: Boss and the Urban Challenge. *AI Magazine* 30(2): 17–28.

Weng, G. K.; Seng, G. K.; Whye, L. C.; Hwa, T. B.; and Kiang, T. E. 1995. Scheduling of Port of Singapore Authority: A Total Approach. In *Proceedings of the Seventh Conference on Innovative Applications of Artificial Intelligence,* 62–69. Menlo Park, CA: AAAI Press.

Winograd, T. 1972. Understanding Natural Language. *Cognitive Psychology* 3(1): 1–191.

Woods, W. A., and Schmolze, J. G. 1992. The KL-ONE Family. *Computers & Mathematics with Applications* 23(2–5): 133–177.

Wu, J.; Williams, K.; Chen, H.-H.; Khabsa, M.; Caragea, C.; Ororbia, A.; Jordan, D.; and Giles, C. L. 2014. CiteSeerX: AI in a Digital Library Search Engine. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence and the Twenty-Sixth Annual Conference on Innovative Applications of Artificial Intelligence,* 2930–2937. Palo Alto, CA: AAAI Press

Zilis, S. 2015. The Current State of Machine Intelligence 2.0. *O'Reilly Media on Our Radar,* December 10. Available at www.oreilly.com/ideas/the-current-state-of-machine-intelligence-2-0.

Zweben, S., and Bizot, B. 2015. Taulbee Survey. *Computing Research News* 27(5): 2–51.

Reid G. Smith is cofounder and CEO of i2k Connect. Formerly, he was vice president of research and knowledge management at Schlumberger, enterprise content management director at Marathon Oil, and senior vice president at Medstory. He holds a Ph.D. in electrical engineering from Stanford University and is a Fellow of AAAI. He is coeditor of AITopics.

Joshua Eckroth is an assistant professor at Stetson University and chief architect of i2k Connect. His research interests lie in abductive reasoning and belief revision as well as computer science pedagogy. Eckroth holds a Ph.D. in computer science from the Ohio State University. He is coeditor of AITopics.

PAWS — A Deployed Game-Theoretic Application to Combat Poaching

Fei Fang, Thanh H. Nguyen, Robert Pickles, Wai Y. Lam, Gopalasamy R. Clements, Bo An, Amandeep Singh, Brian C. Schwedock, Milind Tambe, Andrew Lemieux

■ Poaching is considered a major driver for the population drop of key species such as tigers, elephants, and rhinos, which can be detrimental to whole ecosystems. While conducting foot patrols is the most commonly used approach in many countries to prevent poaching, such patrols often do not make the best use of the limited patrolling resources.

This article presents PAWS, a game-theoretic application deployed in Southeast Asia for optimizing foot patrols to combat poaching. In this article, we report on the significant evolution of PAWS from a proposed decision aid introduced in 2014 to a regularly deployed application. We outline key technical advances that lead to PAWS's regular deployment: (1) incorporating complex topographic features, for example, ridgelines, in generating patrol routes; (2) handling uncertainties in species distribution (game-theoretic payoffs); (3) ensuring scalability for patrolling large-scale conservation areas with fine-grained guidance; and (4) handling complex patrol scheduling constraints.

Poaching is a serious threat to wildlife conservation and can lead to the extinction of species and destruction of ecosystems. For example, poaching is considered a major driver (Chapron et al. 2008) of why tigers are now found in less than 7 percent of their historical range (Sanderson et al. 2006), with three out of nine tiger subspecies already extinct (IUCN 2015). As a result, efforts have been made by law enforcement agencies in many countries to protect endangered animals from poaching. The most direct and commonly used approach is conducting foot patrols. However, given their limited human resources and the vast area in need of protection, improving the efficiency of patrols remains a major challenge.

Game theory has become a well-established paradigm for addressing complex resource allocation and patrolling problems in security and sustainability domains. Models and algorithms have been proposed and studied extensively in the past decade, forming the general area of security games (Tambe 2011). Furthermore, several security-game-based decision support systems have previously been successfully deployed in protecting critical infrastructure such as airports, ports, and metro trains (Pita et al. 2008; Shieh et al. 2012; Yin et al. 2012). Inspired by the success of these deployments, researchers have begun applying game theory to generating effective patrol strategies in green security domains such as protecting wildlife (Yang et al. 2014; Fang, Stone, and Tambe 2015), preventing overfishing (Haskell et al. 2014, Qian et al. 2014), and illegal logging (Johnson, Fang, and Tambe 2012).

Among these prior works, a novel emerging application called PAWS (protection assistant for wildlife security) (Yang et al. 2014) was introduced as a game-theoretic decision aid to optimize the use of human patrol resources to combat poaching. PAWS was the first of a new wave of proposed applications in the subarea now called *green security games* (Fang, Stone, and Tambe 2015; Kar et al. 2015). Specifically, PAWS solves a repeated Stackelberg security game, where the patrollers (defenders) conduct randomized patrols against poachers (attackers) while balancing the priorities of different locations with different animal densities. Despite its promise, the initial PAWS effort did not test the concept in the field.

This article reports on PAWS's significant evolution over the last two years from a proposed decision aid to a regularly deployed application. We report on the innovations made in PAWS and lessons learned from the first tests in Uganda in spring 2014, through its continued evolution since then, to current deployments in Southeast Asia and plans for future worldwide deployment. In this process, we have worked closely with two nongovernment organizations (Panthera and Rimba) and incorporated extensive feedback from professional patrolling teams. Indeed, the first tests revealed key shortcomings in PAWS's initial algorithms and assumptions (we will henceforth refer to the initial version of PAWS as PAWS-Initial, and to the version after our enhancement as PAWS). First, a major limitation was that PAWS-Initial ignored topographic information. Second, PAWS-Initial assumed animal density and relevant problem features at different locations to be known, ignoring the uncertainty. Third, PAWS-Initial could not scale to provide detailed patrol routes in large conservation areas. Finally, PAWS-Initial failed to consider patrol-scheduling constraints.

In this article, we outline novel research advances that remedy the aforementioned limitations, making it possible to deploy PAWS on a regular basis. First, we incorporate elevation information and land features and use a novel hierarchical modeling approach to building a virtual street map of the conservation area. This virtual street map helps scale-up while providing fine-grained guidance and is an innovation that would be useful in many other domains requiring patrolling of large areas. Essentially, the street map connects the whole conservation area through easy-to-follow route segments such as ridgelines, streams, and river banks. The rationale for this comes from the fact that animals, poachers, and patrollers all use these features while moving. To address the second and third limitations, we build on the street map concept with a novel algorithm that uniquely synthesizes two threads of prior work in the security games literature; specifically, the new PAWS algorithm handles payoff uncertainty using the concept of minimax regret (Nguyen et al. 2015), while simultaneously ensuring scalability — using our street maps — through the cutting plane framework (Yang et al. 2013). Finally, we incorporate into PAWS the ability to address constraints such as patrol time limits and starting and ending at the base camp. In the final part of the article, we provide detailed information about the regular deployment of PAWS.

Background and Related Work

Criminologists have worked on the problem of combating poaching, from policy design to illegal trade prevention (Lemieux 2014). Geographic information systems (GIS) experts (Hamisi 2008) and wildlife management staff (Wato, Wahungu, and Okello 2006) have carefully studied the identification of poaching hotspots. In recent years, software tools such as SMART[2] and MIST (Stokes 2010) have been developed to help conservation managers record data and analyze patrols retrospectively. We work on a complementary problem of optimizing the patrol planning of limited security staff in conservation areas.

In optimizing security resource allocation, previous work on Stackelberg security games (SSGs) has led to many successfully deployed applications for the security of airports, ports, and flights (Pita et al. 2008; Fang, Jiang, and Tambe 2013). Based on the early work on SSGs, recent work has focused on green security games (Kar et al. 2015), providing conceptual advances in integrating learning and planning (Fang, Stone, and Tambe 2015) and the first application to wildlife security, PAWS-Initial. PAWS-Initial (Yang et al. 2014) models the interaction between the patroller (defender) and the poacher (attacker) who places snares in the conservation area (see figure 1) as a basic green security game, that is, a repeated SSG, where every few months, poaching data is analyzed, and a new SSG is set up enabling improved patrolling strategies. The deployed version of PAWS adopts this framework.

We provide a brief review of SSGs, using PAWS as a

key example. In SSGs, the defender protects T targets from an adversary by optimally allocating a set of R resources ($R < T$) (Pita et al. 2008). In PAWS, the defender discretizes the conservation area into a grid, where each grid cell is viewed as a target for poachers, to be protected by a set of patrollers. The defender's pure strategy is an assignment of the resources to targets. The defender can choose a mixed strategy, which is a probability distribution over pure strategies. The defender strategy can be compactly represented as a coverage vector $\mathbf{c} = \langle c_i \rangle$ where c_i is the coverage probability, that is, the probability that a defender resource is assigned to be at target i (Korzhyk, Conitzer, and Parr 2010). The adversary observes the defender's mixed strategy through surveillance and then attacks a target. An attack could refer to the poacher, a snare, or some other aspect facilitating poaching (for example, poaching camp). Each target is associated with payoff values that indicate the reward and penalty for the players. If the adversary attacks target i, and i is protected by the defender, the defender gets reward $U^d_{r,i}$ and the adversary receives penalty $U^a_{p,i}$. Conversely, if not protected, the defender gets penalty $U^d_{p,i}$ and the adversary receives reward $U^a_{r,i}$. $U^a_{r,i}$ is usually decided by animal density — higher animal density implies higher payoffs. Given a defender strategy $\mathbf{c}$ and the penalty and reward values, we can calculate the players' expected utilities U^a_i and U^d_i when target i is attacked accordingly.

In SSGs, the adversary's behavior model decides his response to the defender's mixed strategy. Past work has often assumed that the adversary is perfectly rational, choosing a single target with the highest expected utility (Pita et al. 2008). PAWS is the first deployed application that relaxes this assumption in favor of a bounded rationality model called SUQR, which models the adversary's stochastic response to defender's strategy (Nguyen et al. 2013). SUQR was shown to perform the best in human subject experiments when compared with other models. SUQR predicts the adversary's probability of attacking i based on a linear combination of three key features at the targets, including the coverage probability c_i, the attacker's reward $U^a_{r,i}$ and penalty $U^a_{p,i}$. A set of parameters (w_1, w_2, w_3) are used for combining the features. They indicate the importance of the features and can be learned from data.

First Tests and Feedback

We first tested PAWS-Initial (Yang et al. 2014) at Uganda's Queen Elizabeth National Park (QENP) for three days. Subsequently, with the collaboration of Panthera and Rimba, we started working in forests in Malaysia in September 2014[1]. These protected forests are home to endangered animals such as the Malayan tiger and Asian elephant but are threatened by poachers. One key difference of this site compared to QENP is that there are large changes in elevation, and the terrain is much more complex. The first four-day patrol in Malaysia was conducted in November 2014. For this test, we set the value of $U^a_{r,i}$ (input for PAWS-Initial) by aggregating observation data recorded during April 2014–September 2014. We first set the importance value of each cell as a weighted sum of the observed counts of different types of animals and different human activities. We then dilute the importance value of available cells to nearby cells by applying a 5 by 5 Gaussian kernel for a convolution operation so as to estimate the value for cells with no data recorded.

Figure 1. Snares Found by Patrollers.

These initial tests revealed four areas of shortcomings, which restricted PAWS-Initial from being used regularly and widely. The first limitation, which was surprising given that it has received no attention in previous work on security games, is the critical importance of topographic information that was ignored in PAWS-Initial. Topography can affect patrollers' speed in key ways. For example, lakes are inaccessible for foot patrols. Not considering such information may lead to failure to complete the patrol route. Figure 2 shows one patrol route during the test in Uganda. The suggested route (orange straight line) goes across the water body (lower right part of figure), and hence the patrollers decided to walk along the water body (black line). Also, changes in elevation require extra patrol effort, and extreme changes may stop the patrollers from following a route. For example, in figure 3a, PAWS-Initial planned a route on a 1 kilometer by 1 kilometer grid (straight lines), and suggested that the patrollers walk to the north area (row 1, column 3) from the south side (row 2, column 3). However, such movement was extremely difficult because of the changes in elevation. So patrollers decided to head toward the

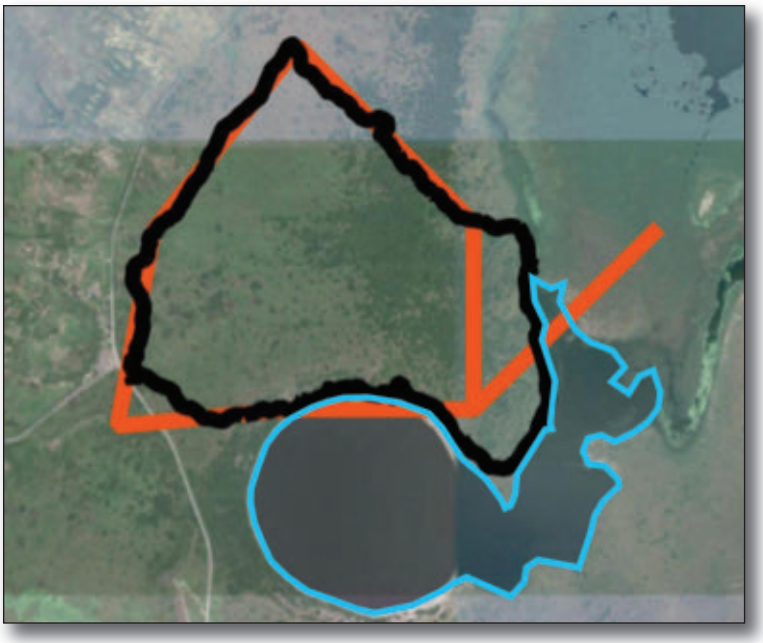


Figure 2. One Patrol Route During the Test in Uganda.

northwest area as the elevation change is more gentle. In addition, it is necessary to focus on terrain features such as ridgelines and streams (figure 3b) when planning routes for three reasons:

First, they are important conduits for certain mammal species such as tigers; hence, second, poachers use these features for trapping and moving about in general; and third, patrollers find it easier to move around here than on slopes. Figure 4a shows a prominent ridgeline.

The second limitation is that PAWS-Initial assumes the payoff values of the targets — for example, $U^{a}_{r,i}$ — are known and fixed. In the domain of wildlife protection, there can be uncertainties due to animal movement and seasonal changes. Thus, considering payoff uncertainty is necessary for optimizing patrol strategy.

The third limitation is that PAWS-Initial cannot scale to provide detailed patrol routes in large conservation areas, which is necessary for successful deployment. Detailed routes require fine-grained discretization, which leads to an exponential number of routes in total.

The fourth limitation is that PAWS-Initial considers covering individual grid cells, but not feasible routes. In practice, the total patrolling time is limited, and the patrollers can move to nearby areas. A patrol strategy for implementation should be in the form of a distribution over feasible patrol routes satisfying these constraints. Without taking these scheduling (routing) constraints into account, the optimal coverage probabilities calculated by PAWS-Initial may not be implementable. Figure 4b shows an example area that is discretized into four cells and the base camp is located in the upper left cell. There are three available patrol routes, each protecting two targets. The coverage probabilities shown in figure 4c cannot be achieved by a randomization over the three routes because the coverage of the upper left cell (Target 1) should be no less than the overall coverage of the remaining three cells since all routes start from the base camp.

PAWS Overview

Figure 5 provides an overview of the deployed version of PAWS. PAWS first takes the input data and estimates the animal distribution and human activity distribution. Based on this information, an SSG-based game model is built, and the patrol strategy is calculated. In wildlife protection, there is repeated interaction between patrollers and poachers. When patrollers execute the patrol strategy generated by PAWS over a period (for example, three months), more information is collected and can become part of the input in the next round.

PAWS provides significant innovations in addressing the aforementioned limitations of PAWS-Initial. In building the game model, PAWS uses a novel hierarchical modeling approach to building a virtual street map, while incorporating detailed topographic information. PAWS models the poachers bounded rationality as described by the SUQR model and considers uncertainty in payoff values. In calculating the patrol strategy, PAWS uses the ARROW (Nguyen et al. 2015) algorithm to deal with payoff uncertainty and adopts cutting plane approach and column generation to address the scalability issue introduced by scheduling constraints.

Input and Initial Analysis

The input information includes contour lines that describe the elevation, terrain information such as lakes and drainage, base camp locations, previous observations (animals and human activities), as well as previous patrol tracks. However, the point detections of the presence of animal and human activity are not likely to be spatially representative. As such, it is necessary to predict the animal and human activity distribution over the entire study area. To this end, we used (1) JAGS (Plummer 2003) to produce a posterior predictive density raster for tigers (as a target species) derived from a spatially explicit capture-recapture analysis conducted in a Bayesian framework; and (2) MaxEnt (Phillips, Anderson, and Schapire 2006) to create a raster of predicted human activity distribution based on meaningful geographical covariates (for example, distance to water, slope, elevation) in a maximum entropy modeling framework.

Build Game Model

Based on the input information and the estimated distribution, we build a game model abstracting the

strategic interaction between the patroller and the poacher as an SSG. Building a game model involves defender action modeling, adversary action modeling, and payoff modeling. We will discuss all three parts but emphasize defender action modeling since this is one of the major challenges to bring PAWS to a regularly deployed application. Given the topographic information, modeling defender actions in PAWS is far more complex than any other previous security game domain.

Defender Action Modeling

Based on the feedback from the first tests, we aim to provide detailed guidance to the patrollers. If we use a fine-grained grid and treat every fine-grained grid cell as a target, computing the optimal patrolling strategy is exceptionally computationally challenging due to the large number of targets and the exponential number of patrol routes. Therefore, a key novelty of PAWS is to provide a hierarchical modeling solution, the first such model in security game research. This hierarchical modeling approach allows us to attain a good compromise between scaling up and providing detailed guidance. This approach would be applicable in many other domains for large open area patrolling where security games are applicable, not only other green security games applications, but others including patrolling of large warehouse areas or large open campuses by robots or UAVs.

More specifically, we leverage insights from hierarchical abstraction for heuristic search such as path planning (Botea, Müller, and Schaeffer 2004) and apply two levels of discretization to the conservation area. We first discretize the conservation area into 1 kilometer by 1 kilometer grid cells and treat every grid cell as a target. We further discretize the grid cells into 50 meters by 50 meters raster pieces and describe the topographic information such as elevation in 50-meter scale. The defender actions are patrol routes defined over a virtual "street map" — which is built in terms of raster pieces while aided by the grid cells in this abstraction as described below. With this hierarchical modeling, the model keeps a small number of targets and reduces the number of patrol routes while allowing for details at the 50-meter scale.

The street map is a graph consisting of nodes and edges, where the set of nodes is a small subset of the raster pieces, and edges are sequences of raster pieces linking the nodes. We denote nodes as key access points (KAPs) and edges as route segments. The street map not only helps scalability but also allows us to focus patrolling on preferred terrain features such as ridgelines. The street map is built in three steps: (1) determine the accessibility type for each raster piece, (2) define KAPs, and (3) find route segments to link the KAPs.

In the first step, we check the accessibility type of every raster piece. For example, raster pieces in a lake are inaccessible, whereas raster pieces on ridgelines or previous patrol tracks are easily accessible. Ridgelines and valley lines are inferred from the contour lines using existing approaches in hydrology (Tarboton, Bras, and Rodriguez-Iturbe 2007).

The second step is to define a set of KAPs, through which patrols will be routed. We want to build the street map in such a way that each grid cell can be reached. So we first choose raster pieces that can serve as entries and exits for the grid cells as KAPs, that is, the ones that are on the boundary of grid cells and are easily accessible according to the accessibility type calculated in the first step. When there are

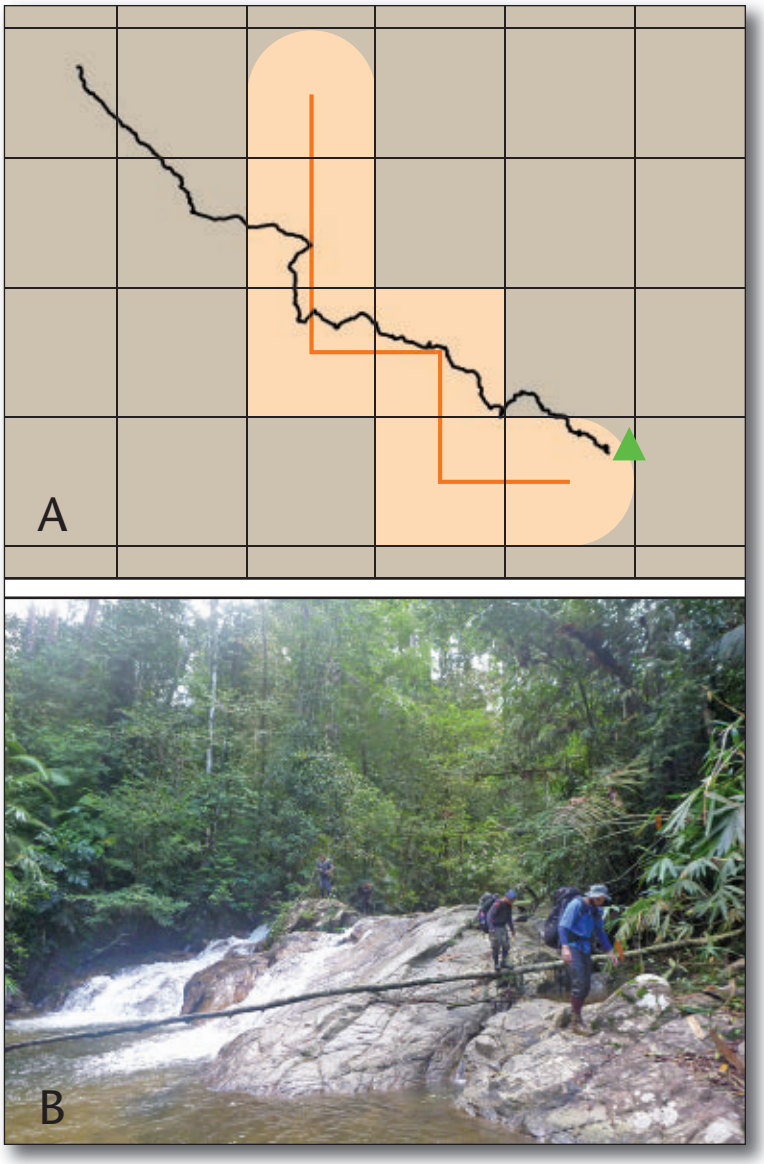


Figure 3. First Four-Day Patrol in Malaysia.

Figure 3a shows one suggested route (orange straight lines) and the actual patrol track (black line). Figure 3b shows the patrollers walking along the stream during the patrol.

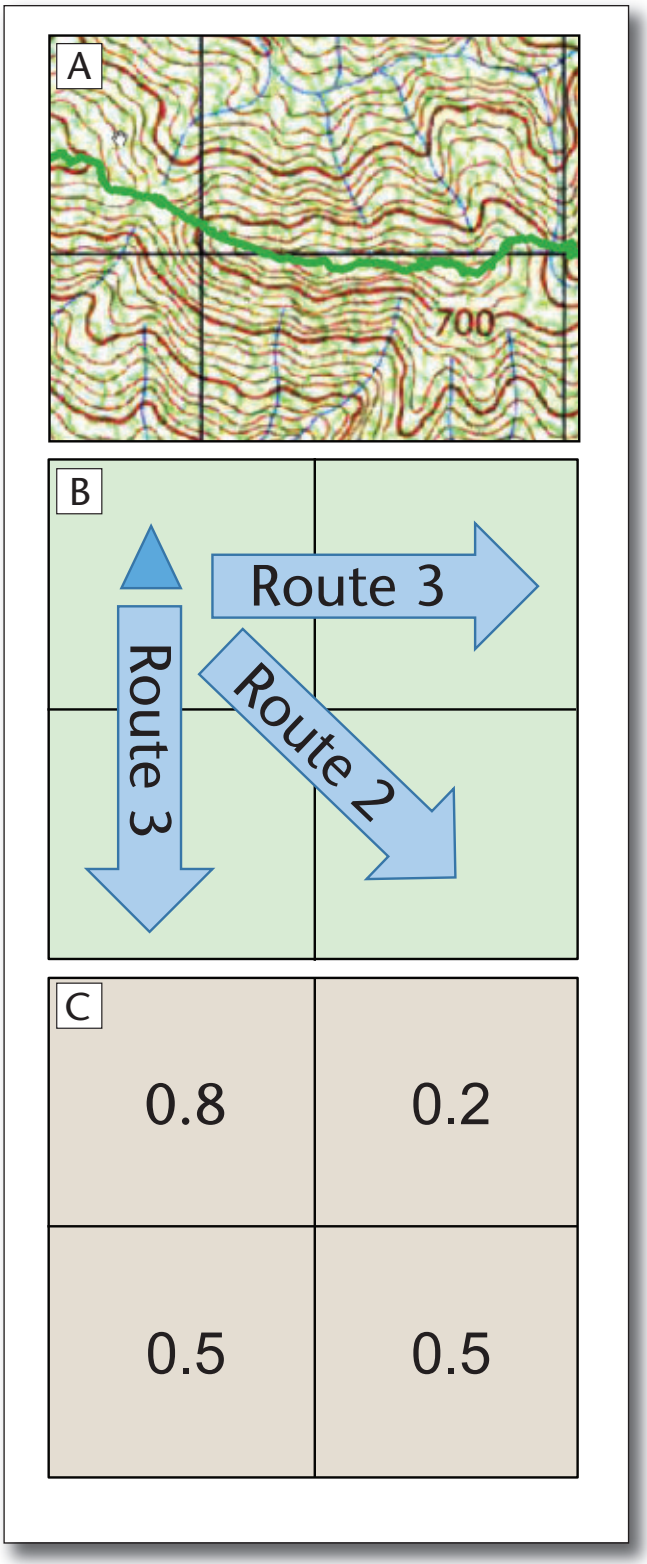


Figure 4. Illustrative Examples.
a. Ridgeline. b. Feasible routes. c. Coverage

multiple adjacent raster pieces that are all easily accessible, we add the midpoint as a KAP. If there are no easily accessible raster pieces on one side of the boundary, we choose the raster piece with the lowest slope as a KAP. In addition, we consider existing base camps as KAPs as they are key points in planning the patroller's route. We choose additional KAPs to ensure KAPs on the boundary of adjacent cells are paired. Figure 6 shows identified KAPs and easily accessible pieces (black and gray raster pieces respectively).

The last step is to find route segments to connect the KAPs. Instead of inefficiently finding route segments to connect each pair of KAPs on the map globally, we find route segments locally for each pair of KAPs within the same grid cell, which is sufficient to connect all the KAPs. When finding the route segment, we design a distance measure that estimates the actual patrol effort and also gives high priority to the preferred terrain features. The effort needed for three-dimensional movement can be interpreted as the equivalent distance on flat terrain. For example, for gentle slopes, equivalent "flat-terrain" distance is obtained by adding eight kilometers for every one kilometer of elevation ascent according to Naismith's rule (Thompson 2011). In PAWS, we apply Naismith's rule with Langmuir corrections (Langmuir 1995) for gentle slopes (< 20°) and apply Tobler's hiking speed function (Tobler 1993) for steep slopes (≥ 20°). Very steep slopes (> 30°) are not allowed. We penalize not walking on preferred terrain features by adding extra distance. Given the distance measure, the route segment is defined as the shortest distance path linking two KAPs within the grid cell.

The defender's pure strategy is defined as a patrol route on the street map, starting from the base camp, walking along route segments and ending with base camp, with its total distance satisfying the patrol distance limit (all measured as the distance on flat terrain). The patroller confiscates the snares along the route and thus protects the grid cells. More specifically, if the patroller walks along a route segment that covers a sufficiently large portion (for example, 50 percent of animal distribution) of a grid cell, the cell is considered to be protected. The defender's goal is to find an optimal mixed patrol strategy — a probability distribution over patrol routes.

Poacher Action Modeling and Payoff Modeling

The poacher's actions are defined over the grid cells to aid scalability. In this game, we assume the poacher can observe the defender's mixed strategy and then chooses one target (a grid cell) and places snares in this target. Following earlier work, the poacher in this game is assumed to be boundedly rational, and his actions can be described by the SUQR model.

Each target is associated with payoff values indicating the reward and penalty for the patrollers and the poachers. As mentioned earlier, PAWS models a zero-sum game and the reward for the attacker (and

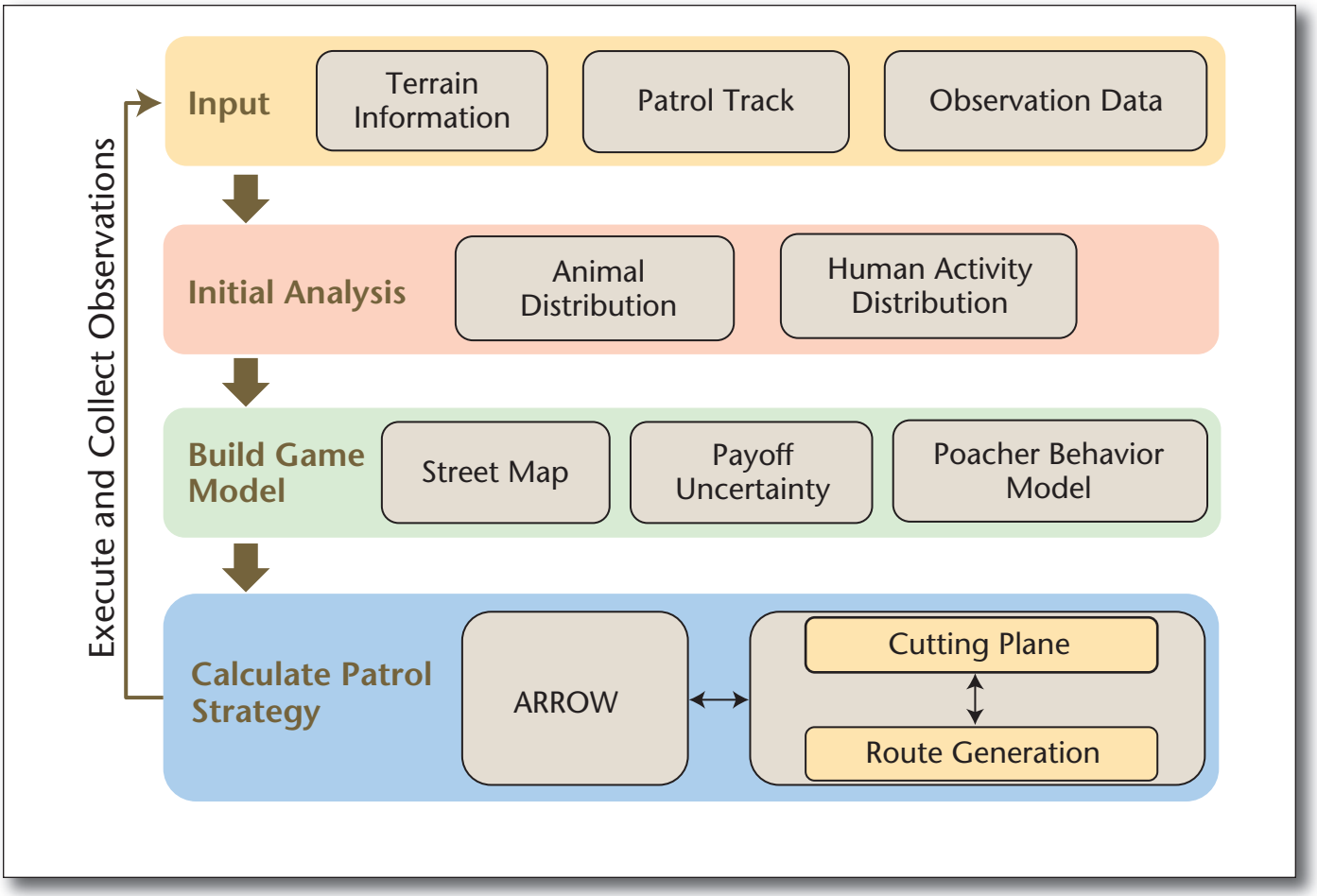


Figure 5. PAWS Overview.

the penalty for the defender) is decided by the animal distribution. However, in this game model, we need to handle uncertainty in the players' payoff values since key domain features, such as animal density, that contribute to the payoffs are difficult to precisely estimate. In addition, seasonal or dynamic animal migration may lead to payoffs to become uncertain in the next season. We use intervals to represent payoff uncertainty in PAWS; the payoffs are known to lie within a certain interval whereas the exact values are unknown. Interval uncertainty is, in fact, a well-known concept to capture uncertainty in security games (Nguyen et al. 2014, 2015). We determine the size of the payoff intervals at each grid cell based on patrollers' patrol efforts at that cell. If the patrollers patrol a cell more frequently, there is less uncertainty in the players' payoffs at that target and thus a smaller size of the payoff intervals.

Calculate Patrol Strategy

We build on algorithms from the rich security game literature to optimize the defender strategy. However, we find that no existing algorithm directly fits our needs as we need an algorithm that can scale up to the size of the domain of interest, where (1) we must generate patrol routes over the street map over the entire conservation area region, while (2) simultaneously addressing payoff uncertainty and (3) bounded rationality of the adversary. While the ARROW (Nguyen et al. 2015) algorithm allows us to address (2) and (3) together, it cannot handle scale-up over the street map. Indeed, while the (virtual) street map is of tremendous value in scaling up as discussed earlier, scaling up given all possible routes (approximately equal to 1012 routes) on the street map is still a massive research challenge. We, therefore, integrate ARROW with another algorithm BLADE (Yang et al. 2013) for addressing the scalability issue, resulting in a novel algorithm that can handle all the three aforementioned challenges. The new algorithm is outlined in figure 7. In the following, we explain how ARROW and BLADE are adapted and integrated.

ARROW attempts to compute a strategy that is robust to payoff uncertainty given that poachers' responses follow SUQR. The concept of minimizing maximum regret is a well-known concept in AI for

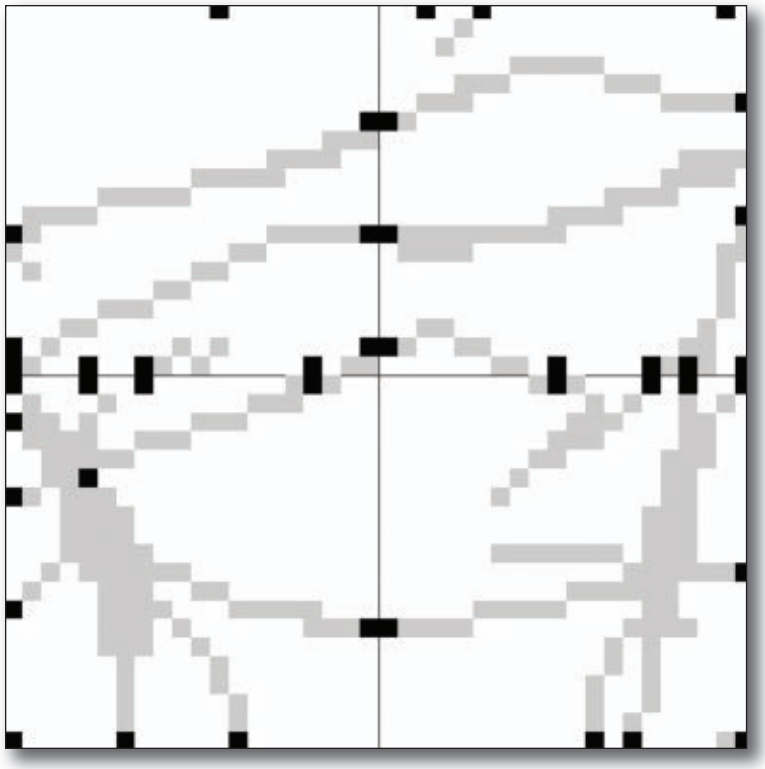


Figure 6. KAPs (Black) for 2 by 2 Grid Cells.

decision making under uncertainty (Wang and Boutilier 2003). ARROW uses the solution concept of behavioral minimax regret to provide the strategy that minimizes regret or utility loss for the patrollers in the presence of payoff uncertainty and bounded rational attackers. In small-scale domains, ARROW could be provided all the routes (the defender's pure strategies), on the basis of which it would calculate the PAWS solution — a distribution over the routes. Unfortunately, in large-scale domains like ours, enumerating all the routes is infeasible. We must, therefore, turn to an approach of incremental solution generation, which is where it interfaces with the BLADE framework.

More specifically, for scalability reasons, ARROW first generates the robust strategy for the patrollers in the form of coverage probabilities over the grid cells without consideration of any routes. Similar to BLADE, a separation oracle is then called to check if the coverage vector is implementable. If it is implementable, the oracle returns a probability distribution over patrol routes that implements the coverage vector, which is the desired PAWS solution. If it is not implementable (see figure 4c for an example of a coverage vector that is not implementable) the oracle returns a constraint (cutting plane) that informs ARROW why it is not. For the example in figure 4b and 4c, if ARROW generates a vector as shown in figure 4c, the constraint returned could be

$$c_1 \geq \sum_{i=2}^{4} c_i$$

since all implementable coverage vectors should satisfy this constraint. This constraint helps ARROW refine its solution. The process repeats until the coverage vector generated by ARROW is implementable.

As described in BLADE (Yang et al. 2013), to avoid enumerating all the feasible routes to check whether the coverage vector is implementable, the separation oracle iteratively generates routes until it has just enough routes (usually after a small number of iterations) to match the coverage vector probabilities or get the constraint (cutting plane). At each iteration of route generation (shown in the bottommost box in figure 7), the new route is optimized to cover targets of high value. However, we cannot directly use any existing algorithm to find the optimal route at each iteration due to the presence of our street map. But we note similarities to the well-studied orienteering problem (Vansteenwegen, Souffriau, and Oudheusden 2011) and exploit the insight of the S-algorithm for orienteering (Tsiligiridis 1984).

In particular, in this bottommost box in figure 7, to ensure each route returned is of high quality, we run a local search over a large number of routes and return the one with the highest total value. In every iteration, we start from the base KAP and choose which KAP to visit next through a weighted random selection. The next KAP to be visited can be any KAP on the map, and we assume the patroller will take the shortest path from the current KAP to the next KAP. The weight of each candidate KAP is proportional to the ratio of the additional target value that can be accrued and distance from the current KAP. We set the lower bound of the weight to be a small positive value to make sure every feasible route can be chosen with positive probability. The process continues until the patroller has to go back to the base to meet the patrol distance limit constraint. Given a large number of such routes, our algorithm returns a route close to the optimal solution.

Integrating all these algorithms, PAWS calculates the patrol strategy consisting of a set of patrol routes and the corresponding probability for taking them.

Addressing Additional Practical Challenges

We have introduced the technical innovations that lead to PAWS's deployment. In addition to these innovations, we have addressed a number of practical constraints to make the strategy suggested by PAWS easier to follow by human patrollers. In this section, we summarize these challenges and our solutions to them.

First, mountaintops should be considered as key points in the patrol route. In PAWS, we require that the patrollers always move between KAPs, which are located at the boundary of the grid cells or the base

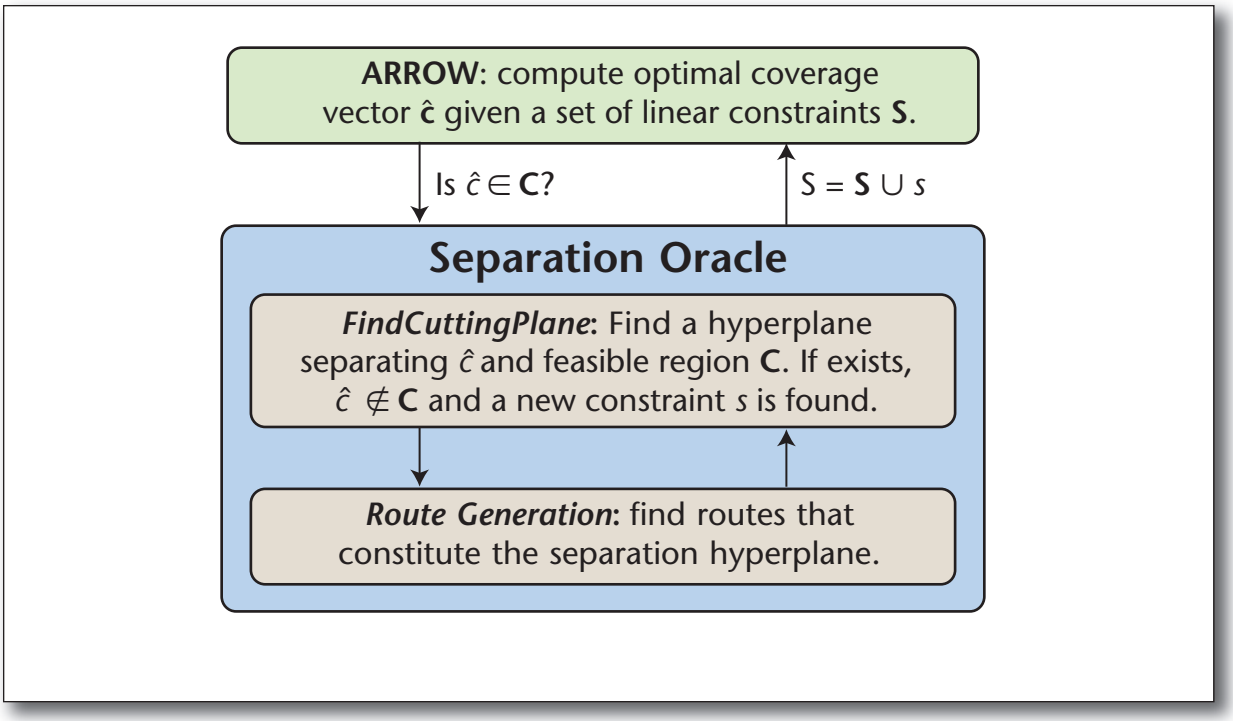


Figure 7. New Integrated Algorithm.

camps. Therefore, in some suggested patrol routes, the patroller is asked to go to a mountaintop and go downhill for a short distance and then backtrack. However, this kind of short downhill followed by returning uphill will annoy the patrollers, and they will naturally ignore the downhill part. We address this problem by considering mountaintops as KAPs when building the street map. With these additional KAPs, patrollers are not forced to take the short downhill unless necessary (that is, when the short downhill covers areas with high animal density).

Second, there is a limit on working time in addition to the limit on walking distance. It takes less time for the patroller to go to an area and then backtrack than to take a loop even if the walking distance is the same. The reason is the patrollers need to spend the time to record what they observe, including animal signs and human activity signs. If the patrollers walk along the same ridgeline twice in a day, they only need to record the signs once. Therefore, in designing the patrol routes, we should consider the total working time in addition to the total distance. This is implemented in PAWS by adding additional constraints in route generation.

Third, not all terrain features should be treated in the same way. In building the street map, we give preference to terrain features like ridgelines by designing a distance measure that penalizes not walking along the terrain feature. However, how much priority should be given to the terrain features depends on the cost of the alternative routes, or how much easier it is compared to taking other routes. On the one hand, in a very hilly region of the area where there are large elevation changes, the patrollers would highly prefer the terrain features as it is much easier to walk along them than taking an alternative route. On the other hand, if the elevation change in the region is small, the effort of taking a ridgeline for unit distance is comparable to that of taking an alternative route. To differentiate these different cases, we use secondary derivatives to check how important the ridgeline is. Instead of penalizing not walking along the terrain features, we can use a discount factor for taking the preferred route, and assign a higher discount factor for terrain features with a higher (regarding absolute value) secondary derivative.

Finally, additional factors such as slope should be considered when evaluating the walking effort. In the distance measure introduced in the previous section, elevation change and terrain features have been considered, but there are other factors that contribute to the walking effort. For example, walking along the contour line will lead to zero elevation change along the way, but the effort needed highly depends on the slope. Walking along the hillside of a steep slope takes much more effort than walking on the flat terrain. Therefore, in the distance measure, we penalize walking along the hillside and assign a higher penalty factor for a higher slope.

Trading Off Exploration and Exploitation

Building on the PAWS framework, we provide a vari-

Average # of Reachable Raster Pieces	9066.67
Average # of Reachable Grid Cells (Targets)	22.67
Average # of Reachable KAPs	194.33

Table 1. Problem Scale for PAWS Patrols.

Average Trip Length	4.67 Days
Average Number of Patrollers	5
Average Patrol Time Per Day	4.48 hours
Average Patrol Distance Per Day	9.29 km

Table 2. Basic Information of PAWS Patrols.

ation of PAWS (denoted as PAWS-EvE) that offers the option of assigning a probability range for selecting an explorative route. Explorative routes are those that cover a significant portion of previously unpatrolled land, while exploitative routes are those that cover a significant portion of land previously patrolled.

The major advantage of selecting exploitative routes is that patrollers are familiar with those patrol routes. Having experience with the routes, patrollers also require less effort on following the routes as they would with unfamiliar territory, enabling them to better cover the area they patrol. Further, the explorative routes may require more effort on checking the map, finding water refill points, and figuring out the best way around unexpected obstacles. Some of these tasks require experienced patrollers and additional equipment, which are not available for every patrol. However, if patrollers were to only take exploitative routes, then poachers would easily be able to observe such a strategy and then focus on targeting other areas. With the objective of PAWS to minimize poaching activity, it is necessary to also take explorative routes. Therefore, offering the option of setting a probability range for selecting an explorative route would be helpful for practical use. The range is set before generating the patrols based on these practical concerns.

To implement the functionality of PAWS-EvE, we modify the previously mentioned separation oracle by introducing two sets of routes, the explorative set and the exploitative set. Two new user-defined parameters θ and δ are introduced and we add two new constraints to make sure the probability of assigning a patrol route from the explorative set lies within a δ margin of θ. Also, in route generation, we check if adding a route from the explorative set would improve the solution, and then we also check if adding a route from the exploitative set would improve the solution.

Deployment and Evaluation

PAWS patrols are now regularly deployed at a conservation area in Malaysia. This section provides details about the deployment and both subjective and objective evaluations of PAWS patrols.

PAWS patrol aims to conduct daily patrols from base camps. Before the patrol starts, PAWS generates the patrol strategy starting from the base camp selected by the patrol team leader. The patrol distance limit considered by PAWS is 10 kilometer per day (equivalent flat terrain). As shown in table 1, this leads to about 9000 raster pieces to be considered. Thus, it is impossible to consider each raster piece as a separate target or consider all possible routes over the raster pieces. With the two-level discretization and the street map, the problem scale is reduced, with 8.57(= 194.33/22.67) KAPs and 80 route segments in each grid cell on average, making the problem manageable. The strategy generated by PAWS is a set of suggested routes associated with probabilities and the average number of suggested routes associated with probability > 0.001 is 12.

Each PAWS patrol lasts for 4–5 days and is executed by a team of 3–7 patrollers. The patrol planner will make plans based on the strategy generated by PAWS. After reaching the base camp, patrollers execute daily patrols, guided by PAWS's patrol routes. Table 2 provides a summary of basic statistics about the patrols. During the patrol, the patrollers are equipped with a printed map, a hand-held GPS, and data-recording booklet. They detect animal and human activity signs and record them with detailed comments and photos. After the patrol, the data manager will put all the information into a database, including patrol tracks recorded by the hand-held GPS, and the observations recorded in the logbook.

Figure 8 shows various types of signs found during the patrols. Table 3 summarizes all the observations. These observations show that there is a serious ongoing threat from the poachers. Column 2 shows results for all PAWS patrols. Column 3 shows results for explorative PAWS patrols, the (partial) patrol routes, which go across areas where the patrollers have never been before. To better understand the numbers, we show in column 4 the statistics about early-stage non-PAWS patrols in this conservation area, which were deployed for a tiger survey. Although it is not a fair comparison as the objectives of the non-PAWS patrols and PAWS patrols are different, comparing columns 2 and 3 with column 4 indicates that PAWS patrols are effective in finding human activity signs and animal signs. Finding the human activity signs is important to identify hotspots of poaching activity,

Courtesy Uda Alok

Figure 8. Various Signs Recorded During PAWS Patrols.

Patrol Type	All PAWS Patrol	Explorative PAWS Patrol	Previous Patrol for Tiger Survey
Total Distance (kilometers)	130.11	20.1	624.75
Average Number of Human Activity Signs per kilometer	0.86	1.09	0.57
Average Number of Animal Signs per kilometer	0.41	0.44	0.18

Table 3. Summary of Observations.

and patrollers' presence will deter the poachers. Animals signs are not a direct evaluation of PAWS patrols, but they indicate that PAWS patrols prioritize areas with higher animal density. Finding these signs is aligned with the goal of PAWS — combat poaching to save animals — and thus is a proof for the effectiveness of PAWS. Comparing column 3 with column 2, we find the average number of observations made along the explorative routes is comparable to and even higher than that of all PAWS patrol routes. The observations on explorative routes are important as they lead to a better understanding of the unex-

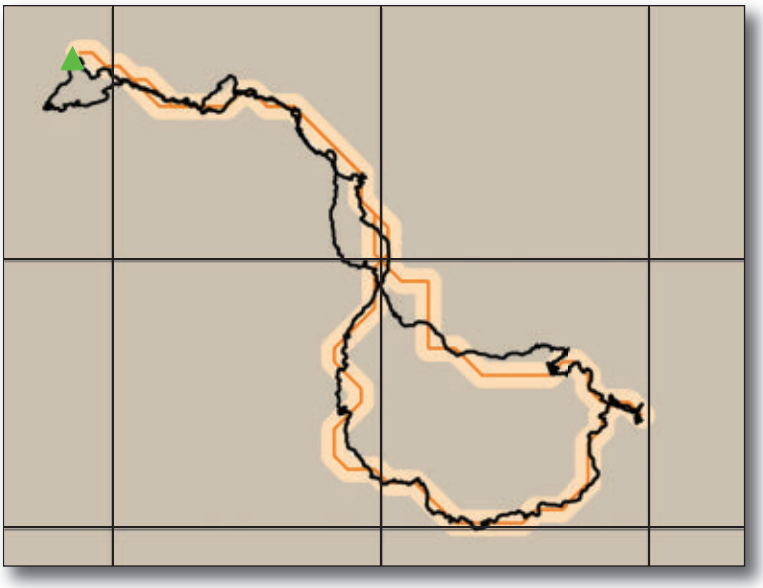


Figure 9. One Daily PAWS Patrol Route in August 2015.

plored area. These results show that PAWS can guide the patrollers toward hotspots of poaching activity and provide valuable suggestions to the patrol planners.

Along the way of PAWS deployment, we have received feedback from patrol planners and patrollers. The patrol planners mentioned that the top routes in PAWS solution (routes with the highest probability) come close to an actual planner's routes, which shows PAWS can suggest feasible routes and potentially reduce the burden of the planning effort. As we deploy PAWS in the future at other sites, the cumulative human planners' effort saved by using PAWS will be a considerable amount. In addition, patrollers commented that PAWS was able to guide them toward poaching hotspots. The fact that they found multiple human signs along the explorative PAWS patrol routes makes them believe that PAWS is good at finding good ridgelines that are taken by animals and humans. Patrollers and patrol planners also agree that PAWS generates detailed suggested routes, which can guide the actual patrol. Patrollers commented that the suggested routes were mostly along the ridgeline, which is easier to follow, compared with the routes from the first trial by PAWS-Initial. Figure 9 shows one suggested route (orange line) and the actual patrol track (black line) during PAWS patrol in August 2015 (shown on a 1 kilometer by 1 kilometer grid). Due to the precision of the contour lines we get, we provide a 50-meter buffer zone (light orange polygon) around the suggested route (orange/light-gray line). The patrollers started from the base camp (the green or shaded triangle) and headed to the southeast. The patrollers mostly followed PAWS's suggested route, indicating that the route generated by PAWS is easy to follow (contrast with PAWS-Initial as shown in figure 3a). Finally, the power of randomization in the PAWS solution can be expected in the long term.

Lessons Learned

During the development and deployment process, we faced several challenges, and here we outline some lessons learned.

First, firsthand immersion in the security environment of concern is critical to understanding the context and accelerating the development process. The authors (from USC and NTU) intentionally went for patrols in the forest with the local patrolling team to familiarize themselves with the area. The firsthand experience confirmed the importance of ridgelines, as several human and animal signs were found along the way, and also confirmed that extreme changes in elevation require a considerable extra effort of the patrollers. This gave us the insight for building the street map.

Second, visualizing the solution is important for communication and technology adaptation. When we communicate with domain experts and human planners, we need to effectively convey the game-theoretic strategy generated by PAWS, which is a probability distribution over routes. We first visualize the routes with probability > 0.01 using ArcGIS so that they can be shown on the topographic map and the animal distribution map. Then for each route, we provide detailed information that can assist the human planners' decision making. We not only provide basic statistics such as probability to be taken and total distance, but also estimate the difficulty level for patrol, predict the probability of finding animals and human signs, and provide an elevation chart that shows how the elevation changes along the route. Such information can help planners' understanding of the strategy, and also help the planner assign patrol routes to the appropriate team of patrollers, as some patrollers may be good at long-distance walking with flat terrain while others would prefer short-distance hiking with high elevation change.

Third, minimizing the need for extra equipment/effort would further ease PAWS future deployment, that is, patrollers would prefer having a single hand-held device for collecting patrol data and displaying suggested patrol routes. If PAWS routes could be embedded in the software that is already in use for collecting data in many conservation areas, for example, SMART, it would reduce the effort required of planners. This is one direction for future development.

Summary

PAWS is a first deployed green security game application to optimize human patrol resources to combat

poaching. We provided key research advances to enable this deployment; this has provided a practical benefit to patrol planners and patrollers. The deployment of PAWS patrols will continue at the site in Malaysia. Panthera has seen the utility of PAWS, and we are taking steps to expand PAWS to its other sites. This future expansion and maintenance of PAWS will be taken over by Armorway,[3] a security games company (starting in spring 2016); Armorway has significant experience in supporting security-games-based software deployments.

Acknowledgement

This research was supported by MURI Grant W911NF-11-1-0332. We also thank our partners in the field who made these tests possible and subcontract from Cornell University for NSF Grant CCF-1522054.

Notes

1. For the security of animals and patrollers, no latitude/longitude information is presented in this article.

2. The Spatial Monitoring and Reporting Tool (SMART), www.smartconservationsoftware.org.

3. The company has now changed its name. See avataai.com.

References

Botea, A.; Müller, M.; and Schaeffer, J. 2004. Near Optimal Hierarchical Path-Finding. *Journal of Game Development* 1(1): 7–28.

Chapron, G.; Miquelle, D. G.; Lambert, A.; Goodrich, J. M.; Legendre, S.; and Clobert, J. 2008. The Impact on Tigers of Poaching Versus Prey Depletion. *Journal of Applied Ecology* 45(6): 1667–1674.

Fang, F.; Jiang, A. X.; and Tambe, M. 2013. Optimal Patrol Strategy for Protecting Moving Targets with Multiple Mobile Resources. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems,* 957–964. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Fang, F.; Stone, P.; and Tambe, M. 2015. When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI).* Palo Alto, CA: AAAI Press.

Hamisi, M. 2008. Identification and Mapping Risk Areas for Zebra Poaching: A Case of Tarangire National Park, Tanzania. Ph.D. Dissertation, Thesis, ITC Faculty of Geo-Information Science and Earth Observation of the University of Twente, Enschede, The Netherlands.

Haskell, W.; Kar, D.; Fang, F.; Tambe, M.; Cheung, S.; and Denicola, E. 2014. Robust Protection of Fisheries with Compass. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence and the Twenty-Sixth Innovative Applications of Artificial Intelligence Conference,* 2978–2983. Palo Alto, CA: AAAI Press.

IUCN. 2015. IUCN Red List of Threatened Species. Version 2015.2. Gland, Switzerland: International Union for Conservation of Nature. (www.iucnredlist.org)

Johnson, M. P.; Fang, F.; and Tambe, M. 2012. Patrol Strategies to Maximize Pristine Forest Area. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI),* 295–301. Palo Alto, CA: AAAI Press.

Kar, D.; Fang, F.; Fave, F. D.; Sintov, N.; and Tambe, M. 2015. A Game of Thrones: When Human Behavior Models Compete in Repeated Stackelberg Security Games. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Korzhyk, D.; Conitzer, V.; and Parr, R. 2010. Complexity of Computing Optimal Stackelberg Strategies in Security Resource Allocation Games. In *Proceedings of the 24th National Conference on Artificial Intelligence,* 805–810. Palo Alto, CA: AAAI Press.

Langmuir, E. 1995. *Mountaincraft and Leadership: A Handbook for Mountaineers and Hillwalking Leaders in the British Isles*. Manchester, UK: Mountain Leader Training Board.

Lemieux, A. M., ed. 2014. *Situational Prevention of Poaching*. Crime Science Series. New York: Taylor & Francis Group / Routledge.

Nguyen, T. H.; Fave, F. M. D.; Kar, D.; Lakshminarayanan, A. S.; Yadav, A.; Tambe, M.; Agmon, N.; Plumptre, A. J.; Driciru, M.; Wanyama, F.; and Rwetsiba, A. 2015. Making the Most of Our Regrets: Regret-Based Solutions to Handle Payoff Uncertainty and Elicitation in Green Security Games. In *Decision and Game Theory for Security: 6th International Conference*. Berlin: Springer.

Nguyen, T. H.; Yadav, A.; An, B.; Tambe, M.; and Boutilier, C. 2014. Regret-Based Optimization and Preference Elicitation for Stackelberg Security Games with Uncertainty. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press.

Nguyen, T. H.; Yang, R.; Azaria, A.; Kraus, S.; and Tambe, M. 2013. Analyzing the Effectiveness of Adversary Modeling in Security Games. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press.

Phillips, S. J.; Anderson, R. P.; and Schapire, R. E. 2006. Maximum Entropy Modeling of Species Geographic Distributions. *Ecological Modelling* 190(3–4): 231–259.

Pita, J.; Jain, M.; Marecki, J.; Ordóñez, F.; Portway, C.; Tambe, M.; Western, C.; Paruchuri, P.; and Kraus, S. 2008. Deployed ARMOR Protection: The Application of a Game Theoretic Model for Security at the Los Angeles International Airport. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems:* Industrial Track, AAMAS 2008, 125–132. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Plummer, M. 2003. JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling. Paper presented at the 3rd International Workshop on Distributed Statistical COmputing, March 20–22, Vienna, Austria.

Qian, Y.; Haskell, W. B.; Jiang, A. X.; and Tambe, M. 2014. Online Planning for Optimal Protector Strategies in Resource Conservation Games. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems* (AAMAS'14). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Sanderson, E.; Forrest, J.; Loucks, C.; Ginsberg, J.; Dinerstein, E.; Seidensticker, J.; Leimgruber, P.; Songer, M.; Heydlauff, A.; OBrien, T.; Bryja, G.; Klenzendorf, S.; and Wikramanayake, E. 2006. Setting Priorities for the Conservation and Recovery of Wild Tigers: 2005–2015. In *Tigers of the*

World: The Science, Politics, and Conservation of Panthera tigris, chapter 9, 143–161, ed. R. Tilson and P. J. Nyhus. Amsterdam, The Netherlands: Elsevier.

Shieh, E.; An, B.; Yang, R.; Tambe, M.; Baldwin, C.; DiRenzo, J.; Maule, B.; and Meyer, G. 2012. PROTECT: A Deployed Game Theoretic System to Protect the Ports of the United States. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems,* Volume 1, AAMAS'12, 13–20. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Stokes, E. J. 2010. Improving Effectiveness of Protection Efforts in Tiger Source Sites: Developing a Framework for Law Enforcement Monitoring Using Mist. *Integrative Zoology* 5(4): 363–377.

Tambe, M. 2011. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge, UK: Cambridge University Press.

Tarboton, D. G.; Bras, R. L.; and Rodriguez-Iturbe, I. 2007. On the Extraction of Channel Networks from Digital Elevation Data. *Hydrologic Processes* 5(1): 81–100.

Thompson, S. 2011. *Unjustifiable Risk?: The Story of British Climbing*. Cumbria, UK: Cicerone Press.

Tobler, W. 1993. Three Presentations on Geographical Analysis and Modeling. Nonisotropic Geographic Modeling: Speculations on the Geometry of Geography, and Global Spatial Analysis. Technical Report 93-1, National center for Geographic Information and Analysis. Santa Barbara, CA: University of California, Santa Barbara.

Tsiligiridis, T. 1984. Heuristic Methods Applied to Orienteering. *The Journal of the Operational Research Society* 35(9): 797–809.

Vansteenwegen, P.; Souffriau, W.; and Oudheusden, D. V. 2011. The Orienteering Problem: A Survey. *European Journal of Operational Research* 209(1): 1–10.

Wang, T., and Boutilier, C. 2003. Incremental Utility Elicitation with the Minimax Regret Decision Criterion. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence,* 309–318. San Francisco: Morgan Kaufmann Publishers.

Wato, Y. A.; Wahungu, G. M.; and Okello, M. M. 2006. Correlates of Wildlife Snaring Patterns in Tsavo West National Park, Kenya. *Biological Conservation* 132(4): 500–509.

Yang, R.; Ford, B.; Tambe, M.; and Lemieux, A. 2014. Adaptive Resource Allocation for Wildlife Protection Against Illegal Poachers. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Yang, R.; Jiang, A. X.; Tambe, M.; and Ordéñez, F. 2013. Scaling-Up Security Games with Boundedly Rational Adversaries: A Cutting-Plane Approach. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press.

Yin, Z.; Jiang, A. X.; Johnson, M. P.; Kiekintveld, C.; Leyton-Brown, K.; Sandholm, T.; Tambe, M.; and Sullivan, J. P. 2012. TRUSTS: Scheduling Randomized Patrols for Fare Inspection in Transit Systems. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence and the Twenty-Fourth Conference on Innovative Applications of Artificial Intelligence,* 2348–2355. Palo Alto, CA: AAAI Press.

Fei Fang is a postdoctoral fellow at the Center for Research on Computation and Society (CRCS) at Harvard University and an adjunct assistant professor at Carnegie Mellon University. Her research lies in the field of artificial intelligence and multiagent systems, focusing on computational game theory with applications to security and sustainability domains.

Thanh H. Nguyen is a postdoctoral researcher at the University of Michigan. Her research is in the field of artificial intelligence, multiagent systems, and machine learning, focusing on game theory with connections to optimization, behavioral modeling, and operations research.

Robert Pickles is a monitoring specialist and the head of monitoring in the Tiger Program in Panthera. He is also a postdoctoral research fellow at the University of Trent. He gained his Ph.D. from the University of Kent and the Zoological Society of London in 2010.

Wai Y. Lam is the technical advisor to Rimba's Project Harimau Selamanya, as well as a Tiger Program scholar with Panthera and a master of science student at Universiti Malaysia Terengganu. Her work involves research on spatiotemporal patterns of people and wildlife in protected areas for antipoaching efforts, camera trap studies, and development of conservation technology in database management.

Gopalasamy R. Clements is an associate professor at the Kenyir Research Institute in Universiti Malaysia Terengganu. He is also the cofounder of Rimba (rimbaresearch.org), a nonprofit research group conducting research on threatened species and ecosystems in Malaysia. His project, Harimau Selamanya, aims to improve protection of three large carnivores in the state of Terengganu, Peninsular Malaysia.

Bo An is an assistant professor at the School of Computer Science and Engineering of the Nanyang Technological University. His research interests include artificial intelligence, multiagent systems, computational game theory, and optimization.

Amandeep Singh is a doctoral candidate at the OID Department at Wharton School of Business. His research interests lie in the areas of digital advertising, crowdfunding, and econometric methods.

Brian C. Schwedock is an undergraduate student at the University of Southern California studying computer engineering and computer science. His research with Teamcore USC includes statistical analysis and security game models.

Milind Tambe is founding codirector of CAIS, the USC Center for AI for Society, and Helen N. and Emmett H. Jones Professor in Engineering at the University of Southern California (USC). He is a fellow of AAAI and ACM, as well as recipient of the ACM/SIGART Autonomous Agents Research Award.

Andrew Lemieux is a researcher at the Netherlands Institute for the Study of Crime and Law Enforcement. His main areas of interest are the spatial and temporal distribution of crime, the use of technology to improve law enforcement operations, and the utility of wildlife crime analysis for data-driven antipoaching operations in Africa and Asia.

Deploying nEmesis: Preventing Foodborne Illness by Data Mining Social Media

Adam Sadilek, Henry Kautz, Lauren DiPrete, Brian Labus, Eric Portman, Jack Teitel, Vincent Silenzio

■ *Foodborne illness afflicts 48 million people annually in the US alone. More than 128,000 are hospitalized and 3000 die from the infection. While preventable with proper food safety practices, the traditional restaurant inspection process has limited impact given the predictability and low frequency of inspections, and the dynamic nature of the kitchen environment. Despite this reality, the inspection process has remained largely unchanged for decades. CDC has even identified food safety as one of seven "winnable battles"; however, progress to date has been limited. In this work, we demonstrate significant improvements in food safety by marrying AI and the standard inspection process. We apply machine learning to Twitter data, develop a system that automatically detects venues likely to pose a public health hazard, and demonstrate its efficacy in the Las Vegas metropolitan area in a double-blind experiment conducted over three months in collaboration with Nevada's health department. By contrast, previous research in this domain has been limited to indirect correlative validation using only aggregate statistics. We show that the adaptive inspection process is 64 percent more effective at identifying problematic venues than the current state of the art. If fully deployed, our approach could prevent more than 9000 cases of foodborne illness and 557 hospitalizations annually in Las Vegas alone. Additionally, adaptive inspections result in unexpected benefits, including the identification of venues lacking permits, contagious kitchen staff, and fewer customer complaints filed with the Las Vegas health department.*

Since its inception, social media have been routinely data mined for marketing consumer goods. Starting around 2010, researchers began to realize that the same techniques could be used for influenza surveillance (Culotta 2010). Since then, social media analytics for public health has been expanded to monitor a variety of conditions, including cholera (Chunara, Andrews, and Brownstein 2012), mental health (Golder and Macy 2011), and diet (Widener and Li 2014). This body of work has shown that social media can be a useful complement to traditional methods, such as surveys of medical providers or individuals, for gathering aggregate public health statistics. Our work extends the social media analytics approach to a new domain, foodborne illness. Our most important contribution, however, is that we go beyond simply monitoring population-level prevalence. Our system, nEmesis, provides specific, actionable information, which is used to support effective public health interventions.

The fight against foodborne illness is complicated by the fact that many cases are not diagnosed or traced back to specific sources of contaminated food. In a typical US city, if a food establishment passes its routine inspection, it may not see the health department again for up to a year. Food establishments can roughly predict the timing of their next inspection and prepare for it. Furthermore, the kitchen environment is dynamic, and ordinary inspections merely provide a snapshot view. For example, the day after an inspection, a contagious cook or server could come to work or a refrigerator could break, either of which can lead to food poisoning. Unless the outbreak is massive, the illness is unlikely to be traced back to the venue.

CDC has identified food safety as one of seven "winnable battles,"[1] along with vehicle accidents and HIV, but progress to date on eradicating the disease has been limited. Our work adds to the arsenal of tools we as humanity can use to fight disease.

We present a novel method for detecting problematic venues quickly — before many people fall ill. We use the term *adaptive inspections* for prioritizing venues for inspection based on evidence mined from social media. Our system (nEmesis) applies machine learning to real-time Twitter data — a popular microblogging service where people post message updates (tweets) that are at most 140 characters long. A tweet sent from a smartphone is usually tagged with the user's precise GPS location. We infer the food venues each user visited by "snapping" his or her tweets to nearby establishments (figure 1). We develop and apply an automated language model that identifies Twitter users who indicate they suffer from foodborne illness in the text of their public online communication. As a result, for each venue, we can estimate the number of patrons who fell ill shortly after eating there. In this paper, we build on our prior work, where we showed a correlation between the number of "sick tweets" attributable to a restaurant and its historic health inspection score (Sadilek et al. 2013). In this paper, we deploy an improved version of the model and validate its predictions in a controlled experiment.

The Southern Nevada Health District (SNHD) conducted a three-month controlled experiment with nEmesis beginning January 2, 2015. Venues with the highest predicted risk on any given day were flagged and subsequently verified through a thorough inspection by an environmental health specialist. For each adaptive inspection, we perform a paired control inspection independent of the online data to ensure full annual coverage required by law and to compensate for the geographic bias of Twitter data. During the first three months, the environmental health specialists inspected 142 venues, half using nEmesis and half following the standard protocol. The latter set of inspections constitutes our control group. The inspectors were not told whether the venue comes from nEmesis or control.

nEmesis downloads and analyzes all tweets that originate from Las Vegas in real time. To estimate visits to restaurants, each tweet that is within 50 meters of a food venue is automatically "snapped" to the nearest one as determined by the Google Places API. We used Google Places to determine the locations of establishments because it includes latitude/longitude data that is more precise than the street address of licensed food venues. As we will see, this decision allowed nEmesis to find problems at unlicensed venues.

For this snapping process, we only consider tweets that include GPS coordinates. Cell phones determine their location through a combination of satellite GPS, WiFi access point fingerprinting, and cell-tower triangularization (Lane et al. 2010). Location accuracy typically ranges from 9 meters to 50 meters and is highest in areas with many cell towers and Wi-Fi access points. In such cases, even indoor localization (for example, within a mall) is accurate.

Once nEmesis snaps a user to a restaurant, it collects all of his or her tweets for the next five days, including tweets with no geo-tag and tweets sent from outside of Las Vegas. This is important because most restaurant patrons in Las Vegas are tourists, who may not show symptoms of illness until after they leave the city. nEmesis then analyzes the text of these tweets to estimate the probability that the user is suffering from foodborne illness.

Determining if a tweet indicates foodborne illness of the user is more complex than simply scanning for a short list of key words. By its nature, Twitter data is noisy. Even a seemingly explicit message, such as "I just threw up," is incomplete evidence that the author of the tweet has a foodborne illness. By using a language model rather than relying on individual key words, our method is able to better model the meaning behind the tweet and is therefore able to capture even subtle messages, such as "have to skip work tomorrow" or "I need to go to a pharmacy." Figure 1 lists the 20 most significant positive and negative language features that contribute to the score.

nEmesis then associates the individual sickness scores to the food venues from which the users originally tweeted. Each snapped twitter user is a proxy for an unknown number of patrons that visited but did not tweet. Since contracting foodborne illness and tweeting at the right times and places is a relatively rare occurrence, even a single ill individual can be a strong evidence of a problem. The web interface (figure 2) is used by the managing health specialist to sort venues by the number of sick users and to dispatch inspectors.

Figure 3 illustrates the full nEmesis process. On a typical day we collect approximately 15,900 geotagged tweets from 3600 users in the Las Vegas area. Approximately 1000 of these tweets, written by 600

Positive Feature		Negative Features	
Feature	**Weight**	**Feature**	**Weight**
stomach	1.7633	think i'm sick	– 0.8411
stomachache	1.2447	i feel soooo	– 0.7156
nausea	1.0935	f--k i'm	– 0.6393
tummy	1.0718	@ID sick to	– 0.6212
#upsetstomach	0.9423	sick of being	– 0.6022
nauseated	0.8702	ughhh cramps	– 0.5909
upset	0.8213	cramp	– 0.5867
naucious	0.7024	so sick omg	– 0.5749
ache	0.7006	tired of	– 0.5410
being sick man	0.6859	cold	– 0.5122
diarrhea	0.6789	burn sucks	– 0.5085
vomit	0.6719	course i'm sick	– 0.5014
@ID i'm getting	0.6424	ifi'm	– 0.4988
#tummyache	0.6422	is sick	– 0.4934
#stomachache	0.6408	so sick and	– 0.4904
i've never been	0.6353	omg i am	– 0.4862
threw up	0.6291	@LINK	– 0.4744
i'm sick great	0.6204	@ID sick	– 0.4704
poisoning	0.5879	if	– 0.4695
feel better tomorrow	0.5643	i feel better	– 0.4670

Figure 1. The Top 20 Most Significant Negatively and Positively Weighted Features in Our Language Model.

unique users, snap to a food venue. nEmesis then tracks these 600 users and downloads all their subsequent tweets for the following five days. These subsequent tracked tweets are then scored by the language model. Finally, venues are ranked based on the number of tweets with sickness score exceeding the threshold of 1.0 determined on a withheld validation set. During the experiment, nEmesis identified on average 12 new tweets per day that were strongly indicative of foodborne illness. Figure 4 shows a distribution over health scores inferred by nEmesis.

Significance of Results

To the best of our knowledge, this is the first study that directly tests the hypothesis that social media provide a signal for identifying specific sources of any disease through a controlled, double-blind experiment during a real-world deployment. By contrast, prior work has been anecdotal, limited to finding correlations, and/or didn't include a control group.

Related Work

Since the famous cholera study by John Snow (1855), much work has been done in capturing the mechanisms of epidemics. There is ample previous work in computational epidemiology on building relatively coarse-grained models of disease spread through differential equations and graph theory (Anderson and May 1979, Newman 2002), by harnessing simulated

Figure 2. nEmesis Web Interface.

The top window shows a portion of the list of food venues ranked by the number of tweeted illness self-reports by patrons. The bottom window provides a map of the selected venue, and allows the user to view the specific tweets that were classified as illness self-reports.

populations (Eubank et al. 2004), and by analysis of official statistics (Grenfell, Bjornstad, and Kappey 2001). Such models are typically developed for the purposes of assessing the impact a particular combination of an outbreak and a containment strategy would have on humanity or ecology (Chen, David, and Kempe 2010).

However, the above works focus on aggregate or simulated populations. By contrast, we address the problem of predicting the health of real-world populations composed of individuals embedded in a social structure and geo-located on a map.

Most prior work on using data about users' online behavior has estimated aggregate disease trends in a large geographical area, typically at the level of a state or large city. Researchers have examined influenza tracking (Culotta 2010; Achrekar et al. 2012; Sadilek and Kautz 2013; Broniatowski and Dredze 2013; Brennan, Sadilek, and Kautz 2013), mental health and depression (Golder and Macy 2011; De Choudhury et al. 2013), as well as general public health across a broad range of diseases (Brownstein, Freifeld, and Madoff 2009; Paul and Dredze 2011b).

Some researchers have begun modeling health and contagion of specific individuals by leveraging fine-grained online social and web search data (Ugander et al. 2012; White and Horvitz 2008; De Choudhury et al. 2013). For example, in Sadilek, Kautz, and Silenzio (2012) we showed that Twitter users exhibiting symptoms of influenza can be accurately detected using a model of language of Twitter posts. A detailed epidemiological model can be subsequently built by following the interactions between sick and healthy individuals in a population, where physical encounters are estimated by spatiotemporal colocated tweets.

Our earlier work on nEmesis (Sadilek et al. 2013) scored restaurants in New York City by their number of sick tweets using an initial version of the language model described here. We showed a weak but significant correlation between the scores and published NYC Department of Health inspection scores. Although the data came from the same year, many months typically separated the inspections and the tweets.

Other researchers have recently tried to use Yelp restaurant reviews to identify restaurants that should be inspected (Harrison et al. 2014). Key words were used to filter 294,000 Yelp reviews for New York City to 893 possible reports of illness. These were manually screened and resulted in the identification of 3 problematic restaurants.

Background: Foodborne Illness

Foodborne illness, known colloquially as food poisoning, is any illness that results from the consumption of contaminated food, pathogenic bacteria, viruses, or parasites that contaminate food, as well as the consumption of chemical or natural toxins such as poisonous mushrooms. The US Centers for Disease Control and Prevention (CDC) estimates that 47.8 million Americans (roughly 1 in 6 people) are sickened each year by foodborne disease. Of that total, nearly 128,000 people are hospitalized, while just over 3000 die of foodborne diseases (CDC 2013).

CDC classifies cases of foodborne illness according to whether they are caused by one of 31 known foodborne illness pathogens or by unspecified agents. These 31 known pathogens account for 9.4 million (20 percent of the total) cases of food poisoning each year, while the remaining 38.4 million cases (80 percent of the total) are caused by unspecified agents.

Food poisoning episodes associated with these 31 known pathogens account for an estimated 44 percent of all hospitalizations resulting from foodborne illness, as well as 44 percent of the deaths. Of these 31 known pathogens, the top five (*Norovirus*, *Salmonella*, *Clostridium perfringens*, *Campylobacter* species, and *Staphylococcus aureus*) account for 91 percent of the cases of foodborne illness, 88 percent of the cases that require hospitalization, and 88 percent of the cases that result in death. The economic burden of health losses resulting from foodborne illness are staggering. One recent study estimated the aggregated costs in the United States alone to be $77.7 billion annually (Scharff 2012).

Despite the variability in the underlying etiology of foodborne illness, the signs and symptoms of disease overlap considerably. The most common symptoms include vomiting, diarrhea (occasionally bloody), abdominal pain, fever, and chills. These symptoms can be mild to serious, and may last from hours to several days. Some pathogens can also cause symptoms of the nervous system, including headache, numbness or tingling, blurry vision, weakness, dizziness, and even paralysis. The gastrointestinal fluid losses can commonly result in dehydration, leading to secondary symptoms such as excessive thirst, infrequent urination, dark-colored urine, lethargy, and lightheadedness. Typically, symptoms appear within hours, but may also occur days to even weeks after exposure to the pathogen (Morris and Potter 2013). According to the US Food and Drug Administration (FDA), the vast majority of these symptoms will occur within three days (FDA 2012).

Public health authorities use an array of surveillance systems to monitor foodborne illness. In the United States, the CDC relies heavily on data from state and local health agencies, as well as more recent systems such as sentinel surveillance systems and national laboratory networks, which help improve the quality and timeliness of data (CDC 2013). An example of the many systems in use by CDC would include the Foodborne Diseases Active Surveillance Network, referred to as FoodNet. FoodNet is a sentinel surveillance system using information provided from sites in 10 states, covering about 15 percent of the US population, to monitor illnesses caused by seven bacteria or two parasites commonly transmitted through food. Other systems include the National Antimicrobial Resistance Monitoring System (NARMS), the National Electronic Norovirus Outbreak Network (CaliciNet), and the National Molecular Subtyping Network for Foodborne Disease Surveillance (PulseNet), among many others.

A major challenge in monitoring foodborne illness is in capturing actionable data in real time. Like all disease surveillance programs, each of the systems currently in use by CDC to monitor foodborne illness can entail significant time lags between when cases are identified and the data is analyzed and reported.

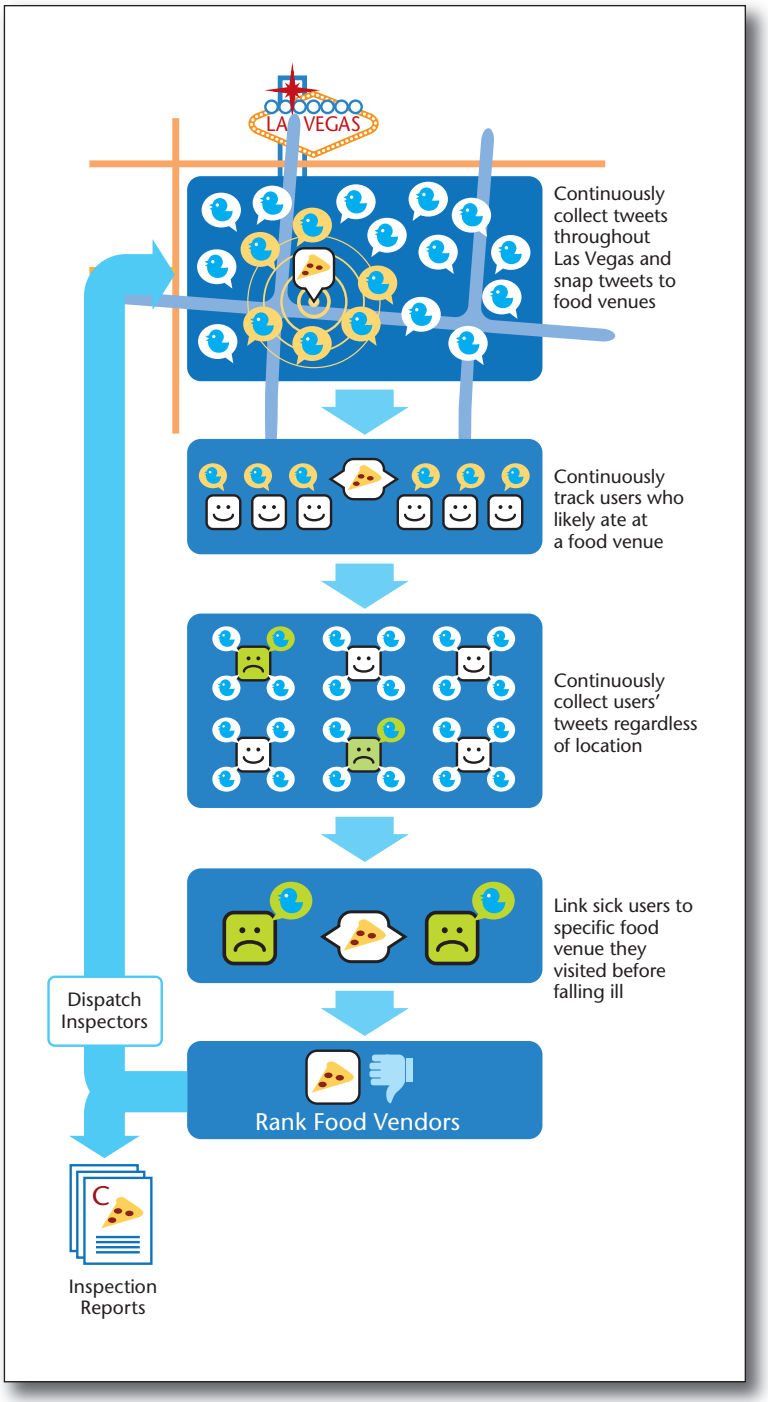


Figure 3. Adaptive Inspection Process.

Starting from the top: All tweets geo-tagged in the Las Vegas area are collected. Tweets geo-tagged within 50 meters of a food venue are snapped to that venue, and the Twitter IDs of the users are added to a database of users to be tracked. All tweets of tracked users are collected for the next five days, whether or not the users remain in Las Vegas. These tweets are evaluated by the language model to determine which are self-reports of symptoms of foodborne illness. Venues are ranked according to the number of patrons who later reported symptoms. Health department officials use the nEmesis web interface to select restaurants for inspection. Inspectors are dispatched to the chosen restaurants, and findings reported.

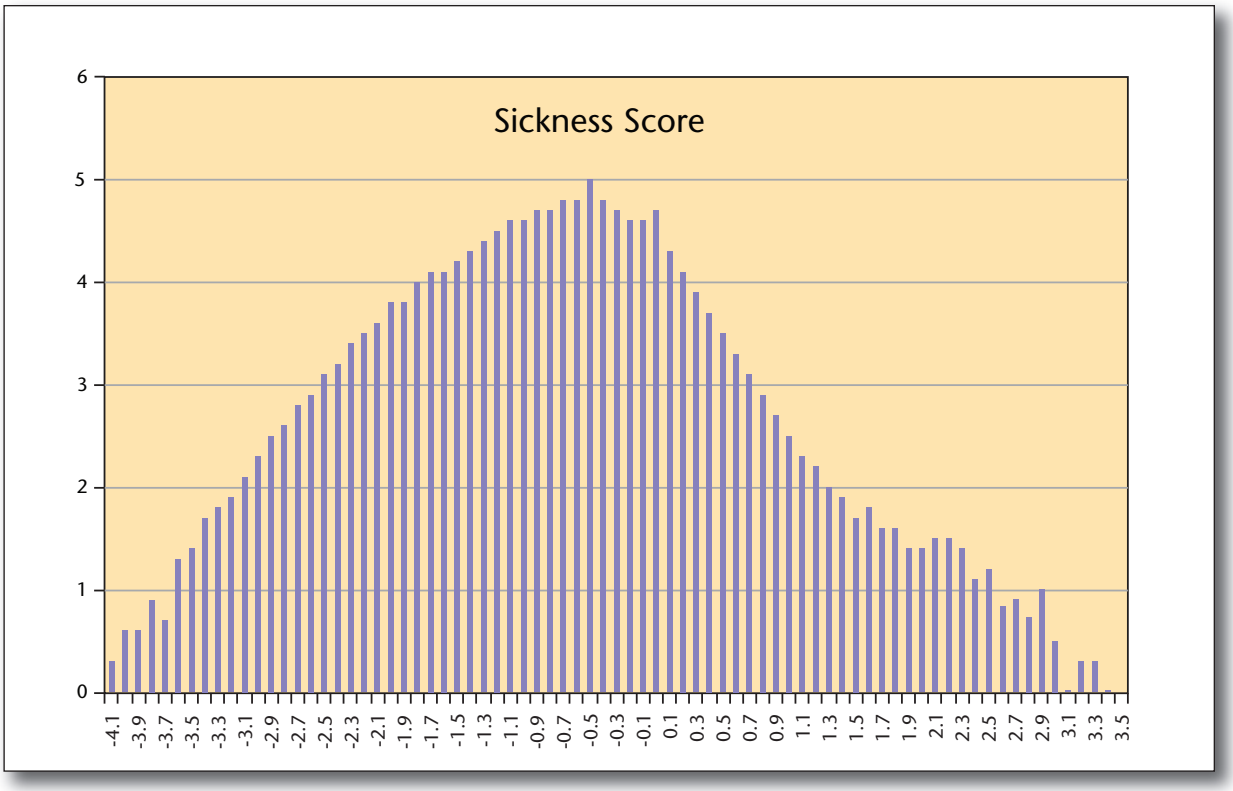


Figure 4. Distribution of Inferred Health Scores (Horizontal Axis) for One Week's Worth of Tweets.

The vertical axis shows the common logarithm of the number of messages with a particular health score. Higher scores indicate increased probability of being sick. Note that a tiny proportion of tweets (scores larger than 1.0) confidently show a foodborne illness.

Whereas this is not as important a limitation in terms of epidemiological surveillance, using surveillance data to actively intervene in outbreaks of foodborne illnesses can be challenging when surveillance data may not infrequently identify cases after the window of opportunity needed to prevent additional cases (Heymann 2004).

Methods

There are three general types of restaurant inspections conducted by health departments. First, restaurants are inspected prior to receiving a permit to ensure that the facility is designed and constructed in a way that allows food to be handled, prepared, and served in a safe manner. For example, inspections would ensure that food contact surfaces were durable and able to be easily cleaned, backflow prevention devices were installed in the plumbing system, and that commercial-grade appliances were installed. Once this type of inspection is completed for a facility, it would not be conducted again unless the facility was renovated.

The second, and most common, type of inspections are routine inspections. Routine inspections are not driven by the occurrence of problems, but are conducted periodically to prevent foodborne illness by ensuring that the facility is operating in accordance with good food-handling practices. Nevada law requires that these types of inspections happen at least annually. A routine inspection is a risk-based process addressing a food establishment's control over the five areas of risk for foodborne illness: personal hygiene, approved food source, proper cooking temperatures, proper holding times and temperatures, and sources of contamination.

A third type of inspection is a complaint-driven inspection initiated by either consumer complaints or the identification of a foodborne illness occurrence that may be associated with the facility. These inspections have a narrow focus but look in depth at a problem. For example, an inspection based on a complaint of improper handwashing at a restaurant would result in the inspector evaluating the handwashing facilities (that is, the availability of hand sinks, hot water, soap, and paper towels) and observing employees as they wash their hands, but would not result in a complete inspection of the facilities. If the inspection were related to foodborne illness, the

inspection would focus on the preparation of the particular foods consumed and the risk factors for the contamination, proliferation or amplification, and survival of the causative organism. This type of inspection is reactive in nature, and while it may prevent additional disease, problems in the facility have already occurred. The ultimate goal of all of these types of inspections is to prevent foodborne illness. Historically, there has been no way to easily identify restaurants having a decline in food handling practices and easily prevent illness, as inspections are based largely on the elapsed time from a previous inspection. As a result, these types of inspections represent the bulk of inspection activities but tend to be rather inefficient in identifying problem facilities. Complaint-driven inspections, while important, identify the problems after they have occurred, which is too late to prevent disease. More importantly, foodborne illnesses are frequently underdiagnosed and underreported (Scallan et al. 2011), preventing public health officials from identifying the source of illness for most foodborne infections.

Clark County, Nevada, is home to more than 2 million people and hosts over 41 million annual visitors to the Las Vegas metropolitan area. The Southern Nevada Health District (SNHD) is the governmental agency responsible for all public health matters within the county and is among the largest local health departments in the United States by population served. In 2014, SNHD conducted 35,855 food inspections (of all types) in nearly 16,000 permitted facilities. In Southern Nevada, inspection violations are weighted based on their likelihood to directly cause a foodborne illness and are divided into critical violations at 5 demerits each (for example, food handlers not washing hands between handling raw food and ready to eat food), to major violations at 3 demerits each (hand sink not stocked with soap), to good food management practices with no demerit value (leak at the hand sink). Demerits are converted to letter grades, where 0–10 is an A, 11–20 is a B, 21–39 is a C, and 40+ is an F (immediate closure). A repeated violation of a critical or major item causes the letter grade to drop to the next lower rank. A grade of C or F represents a serious health hazard.

Controlled Experiment: Adaptive Inspections

During the experiment, when a food establishment was flagged by nEmesis in an inspector's area, he was instructed to conduct a standard, routine inspection on both the flagged facility (adaptive inspection) and also a provided control facility (routine inspection). Control facilities were selected according to their location, size, cuisine, and their permit type to pair the facilities as closely as possible. The inspector was blind as to which facility was which, and each facility received the same risk-based inspection as the other.

Labeling Data at Scale

To scale the laborious process of labeling training data for our language model, we turn to Amazon's Mechanical Turk.[2] Mechanical Turk allows requesters to harness the power of the crowd in order to complete a set of human intelligence tasks (HITs). These HITs are then completed online by hired workers (Mason and Suri 2012).

We formulated the task as a series of short surveys, each 25 tweets in length. For each tweet, we ask "Do you think the author of this tweet has an upset stomach today?" There are three possible responses ("Yes," "No," "Can't tell"), out of which a worker has to choose exactly one (figure 5). We paid the workers 1 cent for every tweet evaluated, making each survey 25 cents in total. Each worker was allowed to label a given tweet only once. The order of tweets was randomized. Each survey was completed by exactly five workers independently. This redundancy was added to reduce the effect of workers who might give erroneous or outright malicious responses. Inter-annotator agreement measured by Cohen's κ is 0.6, considered a moderate to substantial agreement in the literature (Landis and Koch 1977). Responses from workers who exhibit consistently low annotator agreement with the majority were eliminated.

Workers were paid for their efforts only after we were reasonably sure their responses were sincere based on inter-annotator agreement. For each tweet, we calculate the final label by adding up the five constituent labels provided by the workers (Yes = 1, No = –1, Can't tell = 0). In the event of a tie (0 score), we consider the tweet healthy in order to obtain a high-precision data set.

Designing HITs to elicit optimal responses from workers is a difficult problem (Mason and Suri 2012). Pricing HITs poorly can lead to workers not even considering a task; HITs that are too long can cause worker attrition, poorly or ambiguously worded HITs will lead to noisy data. Worker satisfaction is also an important "latent" factor, which should not be taken lightly. Many Mechanical Turk workers are members of communities that offer requester reviews, very similar to Amazon's product review system. As a result, requesters who are unresponsive or opportunistic will soon find it hard to get any HIT completed.

Given that tweets indicating foodborne illness are relatively rare, learning a robust language model poses considerable challenges (Japkowicz et al. 2000; Chawla, Japkowicz, and Kotcz 2004). This problem is called class imbalance and complicates virtually all machine learning. In the world of classification, models induced in a skewed setting tend to simply label all data as members of the majority class. The problem is compounded by the fact that the minority class members (sick tweets) are often of greater interest than the majority class.

We overcome class imbalance faced by nEmesis

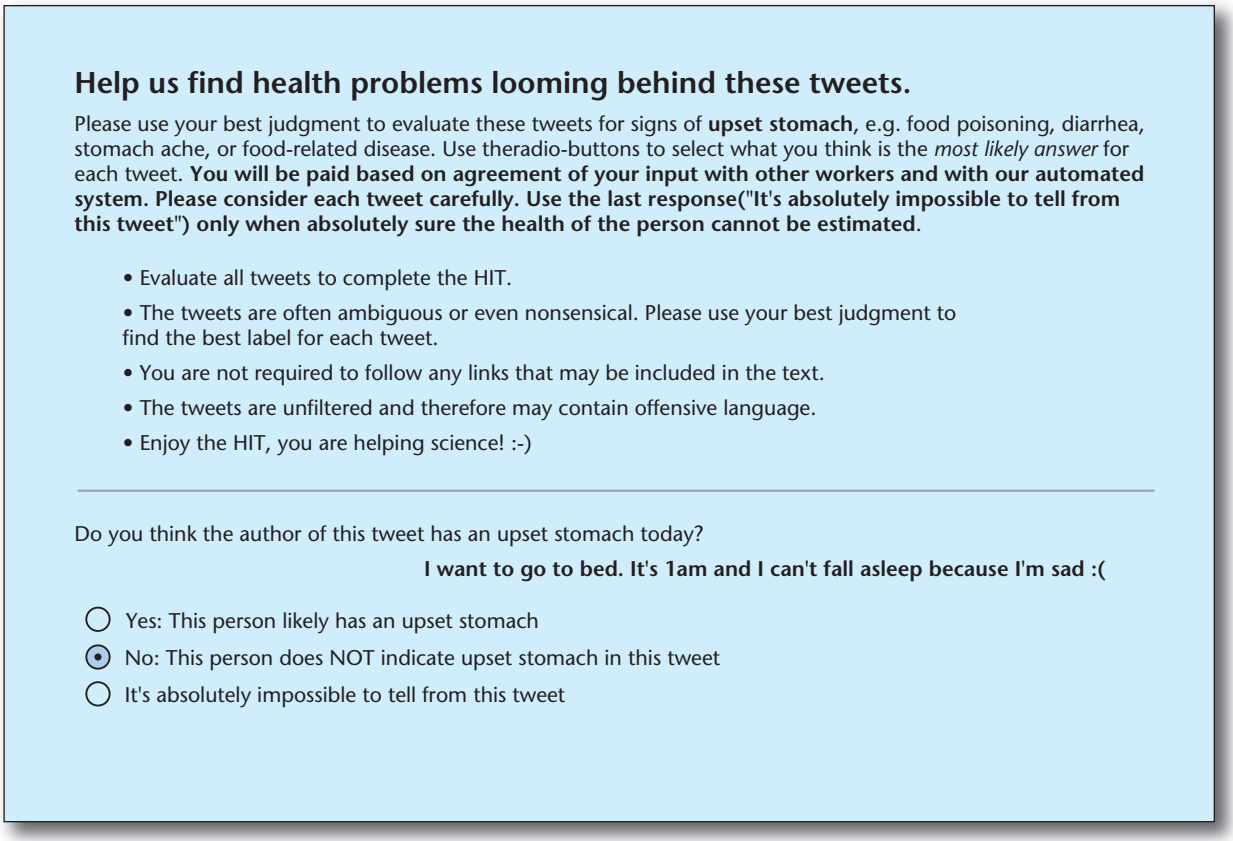


Figure 5. Example of a Mechanical Turk Task.

In this task, online workers are asked to label a given tweet. While tweets are often ambiguous, we encouraged workers to use their best judgment and try to polarize their answers. We found that when workers are presented with too many options, they tend to select "Can't tell" even when the text contains a strong evidence of illness.

through a combination of two techniques: human guided active learning, and learning a language model that is robust under class imbalance. We cover the first technique in this section and discuss the language model induction in the following section.

Previous research has shown that under extreme class imbalance, simply finding examples of the minority class and providing them to the model at learning time significantly improves the resulting model quality and reduces human labeling cost (Attenberg and Provost 2010). In this work, we leverage human guided machine learning — a novel learning method that considerably reduces the amount of human effort required to reach any given level of model quality, even when the number of negatives is many orders of magnitude larger than the number of positives (Sadilek et al. 2013). In our domain, the ratio of sick to healthy tweets is roughly 1 : 2500.

In each human guided learning iteration, nEmesis samples representative and informative examples to be sent for human review. As the focus is on the minority class examples, we sample 90 percent of tweets for a given labeling batch from the top 10 percent of the most likely sick tweets (as predicted by our language model). The remaining 10 percent is sampled uniformly at random to increase diversity. We use the HITs described above to obtain the labeled data.

In parallel with this automated process, we hire workers to actively find examples of tweets in which the author indicates he or she has an upset stomach. We asked them to paste a direct link to each tweet they find into a text box. Workers received a base pay of 10 cents for accepting the task, and were motivated by a bonus of 10 cents for each unique relevant tweet they provided. Each wrong tweet resulted in a 10 cent deduction from the current bonus balance of a worker. Tweets judged to be too ambiguous were neither penalized nor rewarded. Overall, we have posted 50 HITs that resulted in 1971 submitted tweets (mean of 39.4 per worker). Removing duplicates yielded 1176 unique tweets.

As a result, we employ human workers that "guide" the classifier induction by correcting the system when it makes erroneous predictions, and proactively seek-

ing and labeling examples of the minority classes. Thus, people and machines work together to create better models faster. This combination of human guided learning and active learning in a loop with a machine model has been shown to lead to significantly improved model quality (Sadilek et al. 2013).

In a postmortem, we have manually verified submitted tweets and 97 percent were correct sick tweets. This verification step could also be crowdsourced. Since searching for relevant tweets is significantly more time consuming than simply deciding if a given tweet contains a good example of sickness, future work could explore multitiered architecture, where a small number of workers acting as "supervisors" verify data provided by a larger population of "assistants." Supervisors as well as assistants would collaborate with an automated model, such as the support vector machine (SVM) classifier described in this paper, to perform search and verification tasks.

Language Model

Harnessing human and machine intelligence in a unified way, we develop an automated language model that detects individuals who likely suffer from a foodborne disease, on the basis of their online Twitter communication.

Support vector machines are an established method for classifying high-dimensional data (Cortes and Vapnik 1995). We train a linear binary SVM by finding a hyperplane with the maximal margin separating the positive and negative data points. Class imbalance, where the number of examples in one class is dramatically larger than in the other class, complicates virtually all machine learning. For SVMs, prior work has shown that transforming the optimization problem from the space of individual data points to one over pairs of examples yields significantly more robust results (Joachims 2005).

We use the trained SVM language model to predict how likely each tweet indicates foodborne illness. The model is trained on 8000 tweets, each independently labeled by five human annotators as described above. As features, the SVM uses all uni-gram, bigram, and tri-gram word tokens that appear in the training data at least twice. For example, a tweet "My tummy hurts" is represented by the following feature vector:

{my, tummy, hurts, my tummy, tummy hurts, my tummy hurts}

Prior to tokenization, we convert all text to lower case and strip punctuation. Additionally, we replace mentions of user identifiers (the "@" tag) with a special @ID token, and all web links with a @LINK token. We do keep hashtags (such as #upsetstomach), as those are often relevant to the author's health state, and are particularly useful for disambiguation of short or ill-formed messages.

Training the model associates a real-valued weight to each feature. The score the model assigns to a new tweet is the sum of the weights of the features that appear in its text. There are more than 1 million features; figure 2 lists the 20 most significant positive and negative features. While tweets indicating illness are sparse and our feature space has a very high dimensionality, with many possibly irrelevant features, support vector machines with a linear kernel have been shown to perform very well under such circumstances (Joachims 2006, Sculley et al. 2011, Paul and Dredze 2011a). Evaluation of the language on a held-out test set of 10,000 tweets shows 0.75 precision and 0.96 recall. The high recall is critical because evidence of illness is very scarce.

System Architecture

nEmesis consists of several modules that are depicted at a high-level in figure 3. Here we describe the architecture in more detail. We implemented the entire system in Python, with NoSQL data store running on Google Cloud Platform. Most of the code base implements data download, cleanup, filtering, snapping (for example, "at a restaurant"), and labeling ("sick" or "healthy"). There is also a considerable model-learning component described in the previous two sections.

Downloader

This module runs continuously and asynchronously with other modules, downloading all geo-coded tweets based upon the bounding box defined for the Las Vegas Metro area. These tweets are then persisted to a local database in JSON format.

Tracker

For each unique Twitter user that tweets within the bounding box, this module continues to download all of their tweets for two weeks, independent of location (also using the official Twitter API). These tweets are also persisted to local storage in JSON format.

Snapper

The responsibility of this module is to identify Las Vegas area tweets that are geo-coded within 50 meters of a food establishment. It leverages the Google Places API, which serves precise location for any given venue. We built an in memory spatial index that included each of those locations (with a square boundary based on the target distance we were looking for). For each tweet, nEmesis identifies a list of Google Places in the index that overlapped with the tweet based on its lat/long. If a given tweet had one or more location matches, the matching venues are added as an array attribute to the tweet.

Labeler

Each tweet in the data store is piped through our SVM model that assigns it an estimate of probability of foodborne illness. All tweets are annotated and saved back into the data store.

Aggregation Pipelines

We use Map Reduce framework on Google App

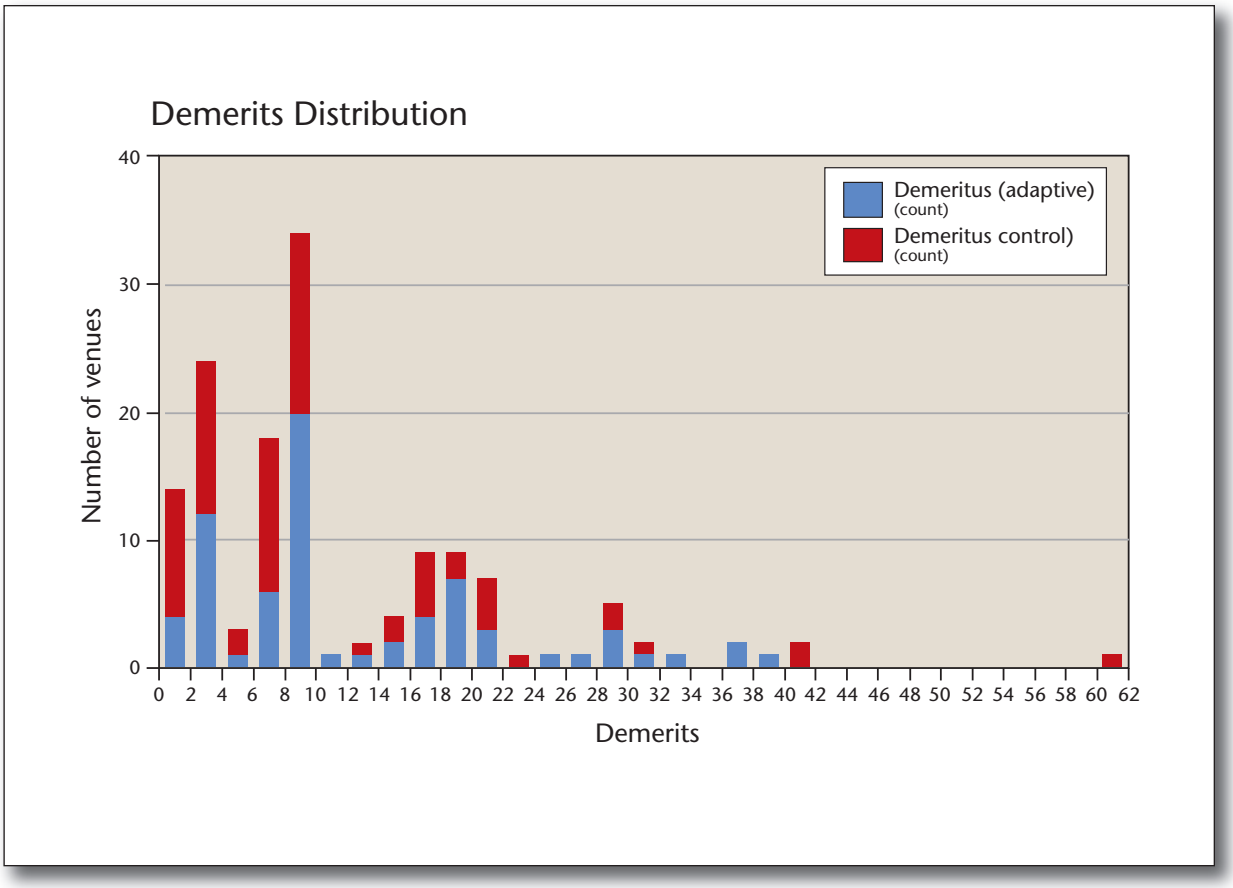


Figure 6. Histogram of the Inspection Results.

The adaptive inspections are blue (light gray), and the control inspections are red (dark gray). The horizontal axis is the number of demerits where the bucket size is 2, and the vertical axis is the number of venues.

Engine to support custom aggregation pipeline. It updates statistics about each venue (number of sick tweets associated with that venue, etc.).

Web Interface

The health professionals interact with nEmesis through a web application shown in figure 1. All modules described above work together to produce a unified view that lists most likely offending venues along with supporting evidence. This allows inspectors to make informed decisions how to allocate their resources. The application was written using a combination of Python for the data access layer and AngularJS for the front-end.

Developing the SVM model took 3 engineer-months. The backend modules above (Downloader through Labeler) took 2 engineer-months, and the Web Interface took an additional engineer-month.

Results and Discussion

Figure 6 is a histogram of the inspection results. There are clearly more control restaurants (red) that passed inspection with flying colors — zero or one demerit. The adaptive inspections (blue) appear to cluster toward the right — more demerits — but a careful statistical analysis is necessary to determine if this is really the case. We use paired Mann-Whitney-Wilcoxon tests to calculate the probability that the distribution of demerits for adaptive inspection is stochastically greater than the control distribution (Mann and Whitney 1947). This test can be used even if the shapes of the distributions are nonnormal and different, which is the case here. The test shows that adaptive inspections uncover significantly more demerits: nine versus six per inspection (p-value of 0.019).

Note that the result would have been even stronger if not for an outlier in the control group, a single control restaurant that received a score of 62 for egregious violations. Even including this outlier, however, we have very strong statistical evidence that adaptive inspections are effective.

Chi-squared test at the level of discrete letter grades (as noted earlier, 0–10 is an A, 11–20 is a B, 21–39 is a C, and 40+ is an F), also show a significant skew

toward worse grades in adaptive inspections. The most important distinction, however, is between restaurants with minor violations (grades A and B) and those posing considerable health risks (grade C and worse). nEmesis uncovers 11 venues in the latter category, whereas control finds only 7, a 64 percent improvement.

All of our data, suitably anonymized to satisfy Twitter's terms of use, is available upon request to other researchers for further analysis.

CDC studies show that each outbreak averages 17.8 afflicted individuals and 1.1 hospitalizations (CDC 2013). Therefore we estimate that adaptive inspections saved 71 infections and 4.4 hospitalizations over the three-month period. Since the Las Vegas health department performs more than 35,000 inspections annually, nEmesis can prevent over 9126 cases of foodborne illness and 557 hospitalizations in Las Vegas alone. This is likely an underestimate as an adaptive inspection can catch the restaurant sooner than a normal inspection. During that time, the venue continues to infect customers.

Adaptive inspections yield a number of unexpected benefits. nEmesis alerted SNHD to an unpermitted seafood establishment. This business was flagged by nEmesis because it uses a comprehensive list of food venues independent of the permit database. An adaptive inspection also discovered a food handler working while sick with an influenza-like disease. Finally, we observed a reduced amount of foodborne illness complaints from the public and subsequent investigations during the experiment. Between January 2, 2015, and March 31, 2015, SNHD performed 5 foodborne illness investigations. During the same time frame the previous year, SNHD performed 11 foodborne illness investigations. Over the last 7 years, SNHD averaged 7.3 investigations during this three-month time frame. It is likely that nEmesis alerted the health district to food safety risks faster than traditional complaint channels, prior to an outbreak.

Given the ambiguity of online data, it may appear hopeless to identify problematic restaurants fully automatically. However, we demonstrate that nEmesis uncovers significantly more problematic restaurants than current inspection processes. This work is the first to directly validate disease predictions made from social media data. To date, all research on modeling public health from online data measured accuracy by correlating aggregate estimates of the number of cases of disease based on online data and aggregate estimates based on traditional data sources (Grassly, Fraser, and Garnett 2005; Brownstein, Wolfe, and Mandl 2006; Ginsberg et al. 2008; Golder and Macy 2011; Sadilek et al. 2013). By contrast, each prediction of our model is verified by an inspection following a well-founded professional protocol. Furthermore, we evaluate nEmesis in a controlled double-blind experiment, where predictions are verified in the order of hours.

Finally, this study also showed that social-media-driven inspections can discover health violations that could never be found by traditional protocols, such as unlicensed venues. This fact indicates that it may be possible to adapt the nEmesis approach for identifying food safety problems in noncommercial venues, ranging from school picnics to private parties. Identifying possible sources of foodborne illness among the public could support more targeted and effective food safety awareness campaigns.

The success of this study has led the Southern Nevada Health District to win a CDC grant to support the further development of nEmesis and its permanent deployment statewide.

Acknowledgements

This research was partly funded by NSF grants 1319378 and 1516340; NIH grant 5R01GM108337-02; and the Intel ISTC-PC.

References

Achrekar, H.; Gandhe, A.; Lazarus, R.; Yu, S.; and Liu, B. 2012. Twitter Improves Seasonal Influenza Prediction. *Proceedings of the Fifth Annual International Conference on Health Informatics*. Setubal, Portugal: Institute for Systems and Technologies of Information, Control and Communication.

Anderson, R., and May, R. 1979. Population Biology of Infectious Diseases: Part I. *Nature* 280(5721): 361.

Attenberg, J., and Provost, F. 2010. Why Label When You Can Search?: Alternatives to Active Learning for Applying Human Resources to Build Classification Models Under Extreme Class Imbalance. In *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining,* 423–432. New York: Association for Computing Machinery.

Brennan, S.; Sadilek, A.; and Kautz, H. 2013. Towards Understanding Global Spread of Disease from Everyday Interpersonal Interactions. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press

Broniatowski, D. A., and Dredze, M. 2013. National and Local Influenza Surveillance Through Twitter: An Analysis of the 2012–2013 Influenza Epidemic. *PLoS ONE* 8(12): e83672. doi: 10.1371/journal.pone.0083672.

Brownstein, J.; Wolfe, C.; and Mandl, K. 2006. Empirical Evidence for the Effect of Airline Travel on Inter-Regional Influenza Spread in the United States. *PLoS Medicine* 3(10): e401. dx.doi.org/10.1371/journal.pmed.0030401

Brownstein, J. S.; Freifeld, B. S.; and Madoff, L. C. 2009. Digital Disease Detection — Harnessing the Web for Public Health Surveillance. *New England Journal of Medicine* 260(21): 2153–2157.

CDC. 2013. *Surveillance for Foodborne Disease Outbreaks United States, 2013: Annual Report*. Technical Report, Centers for Disease Control and Prevention National Center for Emerging and Zoonotic Infectious Diseases. Atlanta, GA: Centers for Disease Control and Prevention.

Chawla, N.; Japkowicz, N.; and Kotcz, A. 2004. Editorial: Special Issue on Learning from Imbalanced Data Sets. *ACM SIGKDD Explorations Newsletter* 6(1): 1–6.

Chen, P.; David, M.; and Kempe, D. 2010. Better Vaccination Strategies for Better People. In *Proceedings of the 11th ACM Conference on Electronic Commerce,* 179–188. New York: Association for Computing Machinery.

Chunara, R.; Andrews, J.; and Brownstein, J. 2012. Social and News Media Enable Estimation of Epidemiological Patterns Early in the 2010 Haitian Cholera Outbreak. *The American Journal of Tropical Medicine and Hygiene* 86(1): 39–45.

Cortes, C., and Vapnik, V. 1995. Support-Vector Networks. *Machine Learning* 20(3): 273–297.

Culotta, A. 2010. Towards Detecting Influenza Epidemics by Analyzing Twitter Messages. Paper presented at the First Workshop on Social Media Analytics, July 25–28, Washington DC.

De Choudhury, M.; Gamon, M.; Counts, S.; and Horvitz, E. 2013. Predicting Depression

via Social Media. *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media,* 128–137. Palo Alto, CA: AAAI Press.

Eubank, S.; Guclu, H.; Anil Kumar, V.; Marathe, M.; Srinivasan, A.; Toroczkai, Z.; and Wang, N. 2004. Modelling Disease Outbreaks in Realistic Urban Social Networks. *Natur*e 429(6988): 180–184.

FDA. 2012. *Bad Bug Book*. U.S. Food and Drug Administration, 2nd ed. Silver Spring, MD: U.S. Food and Drug Administration.

Ginsberg, J.; Mohebbi, M.; Patel, R.; Brammer, L.; Smolinski, M.; and Brilliant, L. 2008. Detecting Influenza Epidemics Using Search Engine Query Data. *Nature* 457(7232): 1012–1014.

Golder, S., and Macy, M. 2011. Diurnal and Seasonal Mood Vary with Work, Sleep, and Daylength Across Diverse Cultures. *Science* 333(6051): 1878–1881.

Grassly, N.; Fraser, C.; and Garnett, G. 2005. Host Immunity and Synchronized Epidemics of Syphilis Across the United States. *Nature* 433(7024): 417–421.

Grenfell, B.; Bjornstad, O.; and Kappey, J. 2001. Travelling Waves and Spatial Hierarchies in Measles Epidemics. *Nature* 414(6865): 716–723.

Harrison, C.; Jorder, M.; Stern, H.; Stavinsky, F.; Reddy, V.; Hanson, H.; Waechter, H.; Lowe, L.; Gravano, L.; and Balter, S. 2014. Using a Restaurant Review Website to Identify Unreported Complaints of Foodborne Illness. *Morbidity and Mortality Weekly Report* 63(20): 441–445.

Heymann, D. L. 2004. C*ontrol of Communicable Diseases Manual: A Report of the American Public Health Association* 18th edition. Washington, DC: American Public Health Association.

Japkowicz, N., et al. 2000. Learning from Imbalanced Data Sets: A Comparison of Various Strategies. In *Learning from Imbalanced Data Sets: Papers from the AAAI Workshop*. Technical Report WS-00-05. Palo Alto, CA: AAAI Press.

Joachims, T. 2005. A Support Vector Method for Multivariate Performance Measures. In *Proceedings of the Twenty-Second International Conference on Machine Learning,* 377–384. New York: Association for Computing Machinery.

Joachims, T. 2006. Training Linear Svms in Linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 217–226. New York: Association for Computing Machinery.

Landis, J. R., and Koch, G. G. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33(1): 159–174.

Lane, N. D.; Miluzzo, E.; Lu, H.; Peebles, D.; Choudhury, T.; and Campbell, A. T. 2010. A Survey of Mobile Phone Sensing. *IEEE Communications Magazine* 48(9): 140–150.

Mann, H., and Whitney, D. 1947. On a Test of Whether One of Two Random Variables Is Stochastically Larger Than the Other. *Annals of Mathematics and Statistics* 18(1): 50–60.

Mason, W., and Suri, S. 2012. Conducting Behavioral Research on Amazon's Mechanical Turk. *Behavior Research Methods* 44(1): 1–23.

Morris, J. G., and Potter, M. 2013. *Foodborne Infections and Intoxications,* 4th ed. Amsterdam: Elsevier Science.

Newman, M. 2002. Spread of Epidemic Disease on Networks. *Physical Review* E 66(1): 016128.

Paul, M., and Dredze, M. 2011a. A Model for Mining Public Health Topics from Twitter. Unpublished paper, Johns Hopkins University..

Paul, M., and Dredze, M. 2011b. You Are What You Tweet: Analyzing Twitter for Public Health. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*. Palo Alto, CA: AAAI Press.

Sadilek, A., and Kautz, H. 2013. Modeling the Impact of Lifestyle on Health at Scale. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*. New York: Association for Computing Machinery.

Sadilek, A.; Brennan, S.; Kautz, H.; and Silenzio, V. 2013. nEmesis: Which Restaurants Should You Avoid Today? In *Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing,* 138–146. Palo Alto, CA: AAAI Press.

Sadilek, A.; Kautz, H.; and Silenzio, V. 2012. Predicting Disease Transmission from Geo-Tagged Micro-Blog Data. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press.

Scallan, E.; Hoekstra, R. M.; Angulo, F. J.; Tauxe, R. V.; Widdowson, M. A.; and Roy, S. L. 2011. Foodborne Illness Acquired in the United States — Major Pathogens. *Emerging Infectious Diseases*. 17(1): 7–15. doi: 10.3201/eid1701.P11101

Scharff, R. L. 2012. Economic Burden from Health Losses Due to Foodborne Illness in the United States. *Journal of Food Protection* 75(1): 123–131.

Sculley, D.; Otey, M.; Pohl, M.; Spitznagel, B.; Hainsworth, J.; and Yunkai, Z. 2011. Detecting Adversarial Advertisements in the Wild. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: Association for Computing Machinery.

Snow, J. 1855. *On the Mode of Communication of Cholera.* London: John Churchill.

Ugander, J.; Backstrom, L.; Marlow, C.; and Kleinberg, J. 2012. Structural Diversity in Social Contagion. *Proceedings of the National Academy of Sciences* 109(16): 5962–5966. Washington, DC: National academy of Sciences of the United States of America.

White, R., and Horvitz, E. 2008. Cyberchondria: Studies of the Escalation of Medical Concerns in Web Search. Technical Report MSR-TR-2008-177, Microsoft Research. Appearing in *ACM Transactions on Information Systems,* 27(4), Article 23, November 2009, DOI 101145/1629096. 1629101.

Widener, M. J., and Li, W. 2014. Using Geolocated Twitter Data to Monitor the Prevalence of Healthy and Unhealthy Food References Across the US. *Applied Geography* 54(October): 189–197.

Adam Sadilek is a senior engineer at Google.

Henry Kautz is a professor in the Department of Computer Science at the University of Rochester.

Lauren DiPrete is a senior environmental health specialist at the Southern Nevada Health District.

Brian Labus is a visiting research assistant professor in the School of Community Health Sciences at the University of Nevada, Las Vegas.

Eric Portman is a consulting research engineer in greater Atlanta Georgia.

Jack Teitel is an undergraduate student at the University of Rochester.

Vincent Silenzio, M.D., is an associate professor at the School of Medicine and Dentistry at the University of Rochester.

Ontology Reengineering: A Case Study from the Automotive Industry

Nestor Rychtyckyj, Venkatesh Raman, Baskaran Sankaranarayanan, P. Sreenivasa Kumar, Deepak Khemani

■ *For more than 25 years Ford Motor Company has been utilizing an AI-based system to manage process planning for vehicle assembly at its assembly plants around the world. The scope of the AI system, known originally as the Direct Labor Management System and now as the Global Study Process Allocation System (GSPAS), has increased over the years to include additional functionality on ergonomics and powertrain assembly (engines and transmission plants). The knowledge about Ford's manufacturing processes is contained in an ontology originally developed using the KL-ONE representation language and methodology. To preserve the viability of the GSPAS ontology and to make it easily usable for other applications within Ford, we needed to reengineer and convert the KL-ONE ontology into a semantic web OWL/RDF format. In this article, we will discuss the process by which we reengineered the existing GSPAS KL-ONE ontology and deployed semantic web technology in our application.*

The Direct Labor Management System (DLMS) (Rychtyckyj 1999) was initially developed and deployed in Ford Motor Company's North American assembly plants back in the early 1990s. It was recognized that an ontology and a reasoner were required to represent the complex knowledge in the manufacturing process. This was done by creating an implementation of the KL-ONE language using the LISP programming language and developing a classifier that could reason with the ontology. This implementation turned out to be extremely successful and became the production version as the system was expanded to assembly plants first in Europe and then the rest of the world. Throughout this, the KL-ONE architecture remained in place as the ontology was expanded and maintained through thousands of updates.

As the semantic web architecture and standards were developed, it became obvious that the Global Study Process Allocation System (GSPAS) KL-ONE ontology would be much more usable and of better value to Ford if it could be rewritten into OWL/RDF. An ontology based on modern semantic web standards would be much easier to maintain and could be extended and utilized for other applications in the company. The main issue was in terms of time and resources: GSPAS was a production system with high value to the business customers and it was impossible to spare the people to redo the ontology and keep the existing system in production. An alternative solution was needed and Ford found it by partnering with the Indian Institute of Technology Madras (IITM) in Chennai, India. Ford elected to partner with IITM because the university has an excellent reputation with a strong background in artificial intelligence (Khemani 2013), and moreover, Ford wanted to develop a strong relationship with the university.

The results of this project were very successful. The IITM team delivered a reengineered OWL/RDF ontology that contained the knowledge in the existing KL-ONE ontology. The Ford team validated and updated the ontology to meet Ford's

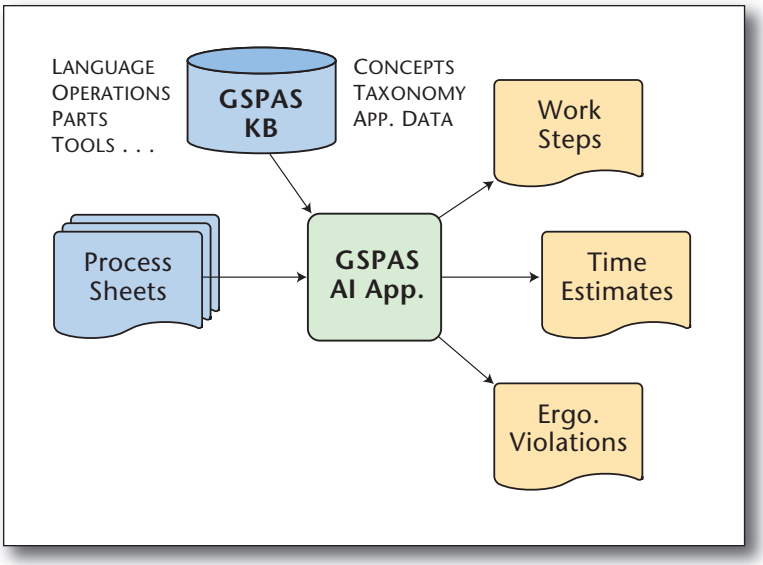


Figure 1. GSPAS AI Application.

TITLE: ASSEMBLE IMMERSION HEATER TO ENGINE

10 OBTAIN ENGINE BLOCK HEATER ASSEMBLY FROM STOCK
20 LOOSEN HEATER ASSEMBLY TURNSCREW USING POWER TOOL
30 APPLY GREASE TO RUBBER O-RING AND CORE OPENING
40 INSERT HEATER ASSEMBLY INTO RIGHT REAR CORE PLUG HOSE
50 ALIGN SCREW HEAD TO TOP OF HEATER

TOOL 20 1 P AAPTCA TSEQ RT ANGLE NUTRUNNER
TOOL 30 1 C COMM TSEQ GREASE BRUSH

Figure 2. Process Sheet.

requirements and has deployed the lexical ontology into the GSPAS application. In the rest of the article we will describe the structure and usage of the existing KL-ONE ontology, and then describe the conversion approach and the conversion process.

In this article, we refer to the GSPAS KL-ONE ontology as GSPAS KB or as GSPAS ontology or as KL-ONE ontology, and refer to the reengineered GSPAS OWL ontology as new ontology or as OWL ontology.

GSPAS and the KL-ONE Ontology

Ford's DLMS was developed to standardize vehicle assembly, improve efficiency, and reduce cost throughout the entire manufacturing process planning system. DLMS was then integrated into Ford's Global Study Process Allocation System, which is currently used across all of Ford's global vehicle assembly and powertrain plants.

Artificial intelligence in GSPAS is used for several different purposes: (1) Validate the correctness of process sheets that describe assembly operations. (2) Develop a list of operator work instructions and associated MODAPTS (modular arrangement of predetermined time standards) codes (Sullivan, Carey, and Farrell 2001) for each assembly operation in the process sheet. (3) Check the process sheet for ergonomic concerns. (4) Translate the process sheets into the language used at a particular assembly plant.

Figure 1 shows the architecture of the GSPAS AI application. Figure 2 shows a sample process sheet with five build steps and two tool specifications; at such granularity, thousands of process sheets are used to document the build steps for a whole vehicle. The core of the GSPAS AI application is an ontology that contains relevant knowledge about Ford's manufacturing processes including the labor requirements for the assembly operations, part and tooling information, workplace ergonomic concerns, linguistic representation of Standard Language (Rychtyckyj 2006) and other concepts. Figure 3 shows how this ontology is used to generate operator work instructions and MODAPTS codes. Each build step in the process sheet is parsed and transformed into a KL-ONE description, which is then classified to find the matching concepts in GSPAS KB. The matching concepts provide meaning to a build step and also supply the necessary work steps and MODAPTS codes.

The ontology was developed inhouse using the KL-ONE knowledge representation language and includes a graphic user interface for ontology editing as well as a classifier. The GSPAS ontology has been updated frequently to keep in sync with all of the changes that have occurred to Ford and the automobile industry in general. The automotive business has evolved dramatically and Ford itself bought and then sold off companies such as Jaguar, Land Rover, and Volvo. The manufacturing process, technology, and tooling have all changed dramatically over the last few years, and all of these changes needed to be reflected in the GSPAS ontology. Technology and parts for new products like electric and hybrid-electric vehicles, in-vehicle infotainment, and aluminum bodies all became part of the Ford manufacturing process and consequently needed to be added into Standard Language and the GSPAS ontology. On the other hand, different concepts in the ontology became obsolete and were no longer needed. Throughout the intervening years and all of the changes, the KL-ONE ontology model and classifier proved to be robust enough to support GSPAS and Ford's manufacturing plants.

Ford adapted the KL-ONE knowledge representation system during its initial development of DLMS. There were no KL-ONE tools or editors available so Ford built both a KL-ONE editor as well as the code for classification and reasoning (Rychtyckyj 1994). The knowledge base update module, an in-house developed graphic user interface, allowed us to maintain the KL-ONE knowledge base and also performed error checking as part of the update process.

The KL-ONE knowledge representation system (Brachman and Schmolze 1985) was first developed in the late 1970s. KL-ONE was selected for use on the DLMS project because of its adaptability as well as the power of the KL-ONE classification algorithm (Lipkis 1981).

The KL-ONE knowledge base as used in DLMS can be described as a network of concepts with the general concepts being closer to the root of the tree and the more specific concepts being the leaves of the tree. A concept in a KL-ONE knowledge base inherits attributes from the nodes that subsume it. The power of the KL-ONE system lies in the classification scheme. The system will place a new concept into its appropriate place in the taxonomy by utilizing the subsumption relation on the concept's attributes. A detailed description of the KL-ONE classification scheme can be found in the papers by Lipkis (1981) and Schmolze and Lipkis (1983).

The existing KL-ONE ontology proved to be very robust and flexible as Ford made hundreds of changes to it on an annual basis. Both the business and the technology changed dramatically, but Ford managed to keep the system fully functional as its scope increased. However, it also became obvious that the KL-ONE framework was limiting the usefulness of the GSPAS ontology. It was difficult to extract and share knowledge with other applications because custom code was needed. The graphic user interface was rewritten several times as the application migrated to new platforms, and maintaining it was time consuming. In the meantime semantic web technology had matured to a point where it was certainly feasible to move into this space. We had previously explored using an automated learning approach to reengineer our KL-ONE ontology, but the results showed that the new ontology was not as intuitive and understandable to users and developers.

Reengineering GSPAS into OWL

The goal is to reengineer the GSPAS ontology into an OWL ontology that will preserve the existing relations and links. This reengineering involves ontology translation, which maps GSPAS ontology to an OWL ontology, and ontology modeling, which identifies a design for the OWL ontology while resolving some of the issues in the existing design.

GSPAS to OWL translation follows a four-layered translation model (Corcho and Gómez-Pérez 2005, Euzenat 2001) consisting of lexical, syntactic, semantic, and pragmatic levels. This model covers all aspects of ontology translation, including semantics preservation, which is a key requirement that is not always easy to satisfy.

In this model, the lexical and syntactic levels deal with the translation of characters, words, values, strings, and sentences between knowledge representation (KR) languages. The semantic level deals with KR framework translation and semantics preservation. The pragmatic level deals with the choice of modeling and encoding that relates to scalability, maintainability, and ontology usage. For example, an entity can be modeled as a class (*red* as a class of *color*) or as an individual *(red* is a *color*). And a binary relation can be modeled as a subclass relation *(obtain* as a class of *verb)* or as a role restriction (*obtain* has part-of-speech some *verb*) or as a role assertion *(obtain* has part-of-speech *verb)*. The choice is between storing information in the taxonomy versus storing it in role links. Further, one can find attach application data to classes, individuals, or roles; and interpret it before or after building the taxonomy.

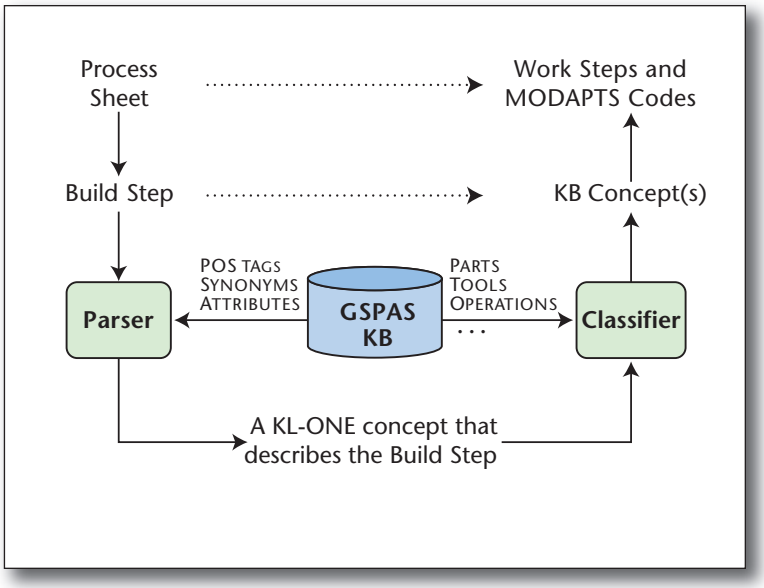


Figure 3. Ontology Use Cases.

As shown in figure 4, our approach to reengineering (modeling and translation) starts with the study phase and works through three levels of abstraction, namely, framework, design, and ontology levels, and finally, ends with the validation phase. We follow a spiral development model, which makes several iterations through the various phases. The framework-mapping and design phases incorporate the semantic and pragmatic aspects from the four-layered model. The ontology conversion tool implements, among other things, the lexical and syntactic translations. The remainder of this section describes the various phases in figure 4.

Study Phase

In this phase, the goal is to study the GSPAS and OWL (Bechhofer et al. 2004) frameworks and the GSPAS ontology and further understand the reengineering problem and identify areas that need improvement.

To accomplish this goal, the IITM team studied the GSPAS, KL-ONE, Description Logics (DL), and OWL

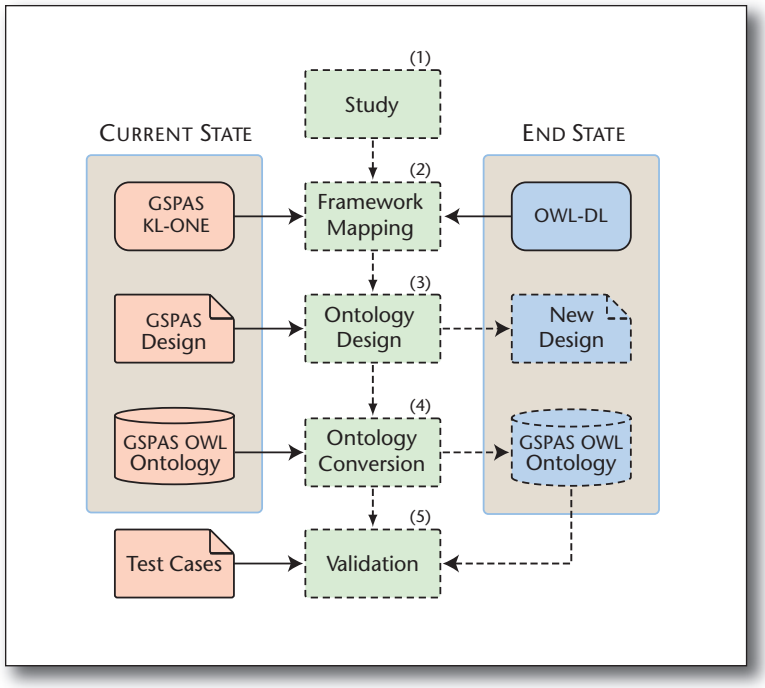


Figure 4. Ontology Reengineering.

The figure shows the current and end states of the ontology, the inputs to reengineering (solid line), and the various phases and deliverables (dashed line).

frameworks, and with the help of the Ford team analyzed the GSPAS ontology. Then the IITM team developed a document that presented (1) their understanding of the KR frameworks, (2) a potential mapping between GSPAS and OWL, (3) their understanding of the design, organization, and use cases of GSPAS ontology, and (4) a high-level approach to GSPAS ontology reengineering.

The Ford team then reviewed the understanding document and worked with the IITM team to validate their understanding of the ontology and to address the questions and fill in the blanks where needed.

Framework Mapping

An ontology describes terms in a domain and captures their association with other terms in that domain. A structure-preserving transformation maps each term and its associations and subsumptions from a source ontology to a term with corresponding associations and subsumptions in a target ontology, and thereby preserves the semantics of these terms.

GSPAS implements a subset of KL-ONE that satisfies Ford's AI needs. We worked with this subset instead of the full KL-ONE. Accordingly, the goal of framework mapping is to create a semantics-preserving mapping between GSPAS (a subset of KL-ONE) and OWL frameworks. This mapping is created for each of vocabulary, representation, and reasoning components of these frameworks.

Vocabulary

GSPAS, KL-ONE, DL, and OWL frameworks, though related, were developed by different groups across space and time. This naturally led to the use of different names to refer to a given idea. Table 1 documents the various vocabularies and their correspondences. It also shows the GSPAS features (un) supported in other frameworks.

Representation

To encode knowledge, the GSPAS ontology uses two kinds of concepts (primitive and defined) and two concept-forming operators (value-restriction and conjunctions), further, it uses classifiable attributes (roles) to define value restrictions, and two kinds of nonclassifiable attributes (nondefinitional roles) to store application data, where one is inherited by subclasses and the other is noninheritable. In KL-ONE and so in GSPAS, a primitive-concept provides necessary conditions for membership, whereas a defined-concept provides both necessary and sufficient conditions for membership. And a value restriction restricts all fillers of a role to a given type or concept, and allows us to describe concepts based on these restrictions, like *things whose tires are slick*. Consider the statement, *Formula One car has slick tires*. If this is taken to provide a necessary condition about F1 cars (a primitive concept) then it states that *tires of F1 car are slick tires*. Instead, if it is taken to provide both necessary and sufficient condition about F1 cars (a defined concept) then it states that *tires of F1 car are slick tires, and things whose tires are slick are F1 cars*. In short, the GSPAS KR language permits the following:

$A \sqsubseteq C$ (primitive concepts); $A \equiv C$ (defined concepts)

where A is any concept name, and C is a concept forming expression which can be a concept-name or a value-restriction or a conjunction, as shown below. Here A_1, A_2 are concept names, R is role name, and C_1, C_2 are concept forming expressions.

$C \rightarrow A_1 \quad C \rightarrow (\forall R.A_2 \sqcap \exists R) \quad C \rightarrow C_1 \sqcap C_2$

Using this notation, we can describe F1-Car as a primitive concept: F1-Car $\sqsubseteq$ Car $\sqcap$ ($\forall$tire.Slick-Tire $\sqcap$ $\exists$tire), which states that F1-Car is a car and all its tires are slick tires and has some tires. See how the textual description resembles the expression.

For a lossless translation, we have to map the GSPAS KR language to OWL constructs that will preserve the meaning of domain terms and their subsumptions. One such mapping (table 2) is discussed next.

First, the primitive concepts are mapped to partial concepts in OWL and are encoded as subclass axioms. And defined concepts are mapped to complete concepts in OWL and are encoded as class-equivalence axioms. Further, concept names and concept conjunctions are mapped, respectively, to class names and class intersections in OWL. These four mappings are exact.

Next, GSPAS roles are mapped to object properties.

	GSPAS	KL-ONE	DL	OWL
1	THING	THING	Top concept 'T'	owl:Thing
2	Concept	Concept	Concept	Class
3	Primitive Concept	Primitive Concept	Atomic Inclusion	Partial Concept
4	Generic Concept	Defined Concept	Definition	Complete Concept
5	Individual	Individual Concept	Individual	Object
6	Role Restriction	Role Restriction	Role Restriction	Property Restriction
7	Value Restriction	Value Restriction	Value Restriction	Value Restricti
8	Number Restriction	Number Restriction	Number Restriction	Cardinality restriction
9	Classifiable Attribute	Role	Role	Object Property
10	Nondefinitional Attribute	Nondefinitional Role	n/a	Annotation Property
11	Nondefinitional Inheritable Attribute	n/a	n/a	n/a
12	Classifier	Classifier	Reasoner	Reasoner

Table 1. Vocabulary Mapping.

	GSPAS	KL-ONE	DL*	OWL
1	Primitive Concept	Primitive Concept	A ! C	rdfs:subClassOf
2	Generic Concept	Defined Concept	A " C	owl:equivalentClass
3	Value Restriction (#R.A $ ∃R)	Value Restriction	∃R.A	owl:someValuesFrom
4	Conjunction	Conjunction	C_1 $ C_2	owl:intersectionOf
5	Classifiable Attribute	Role	Role	owl:ObjectProperty
6	Nondefinitional Attribute	Non-definitional Role	n/a	owl:AnnotationProperty
7	Nondefinitional Inheritable Attribute	n/a	n/a	n/a

Table 2. GSPAS KR Primitives (Modeling Elements) and Their OWL Translation.

*In DL expression, A is concept name; C, C_1, C_2 are concept forming expressions; R is role name.

And nondefinitional attributes are mapped to annotation properties. The inheritable nondefinitional attributes (not supported in KL-ONE and OWL) are modeled as annotation properties and the attribute inheritance is handled in the application. These three mappings are lossless, and the logical (roles and concepts) versus nonlogical (annotation properties and application data) separation remains intact.

Finally, GSPAS value restriction (∀R.A ⊓ ∃R), which restricts all fillers of R to concept A, is remodeled as existential restriction ∃R.A in the OWL ontology, which restricts R to have some fillers from concept A and, optionally, other fillers from other concepts. It is our observation that, in the GSPAS ontology, concepts that are best modeled using existential restriction are modeled using value restriction.

Observe that (∀R.A ⊓ ∃R) is a subclass of ∃R.A, and so, the existential restriction admits more models than the corresponding value restriction. This is a widening or relaxing transformation that preserves subsumption structure (subclass or *is-a* relation). We will justify this for both assertion and inference links.

Consider two value restrictions in figure 5, and their translation given by *is-*a_1 and *is-*a_2. If *is-*a_3 is asserted in the GSPAS ontology then *is-*a_4 will be asserted during ontology conversion. By *is-*a_3 and *is-*a_1 all individuals of (∀R.A_2 ⊓ ∃R) will belong to ∃R.A_1, making *is-*a_5 true. By similar argument, *is-*a_2 and *is-*a_4 also make *is-*a_5 true. As a result, the asserted *is-*a_4 agrees with the assertion *is-*a_3 (figure 5).

The sufficient conditions for inferring *is-a* link between a concept *Sub* and a concept *Super* is stated in Lipkis (1981). Two of the relevant conditions are (1) Each role of *Super* is modified by a role of *Sub*. (2) Each value description of each role of *Super* subsumes a value description of the corresponding role of *Sub*. Accordingly, if A_1 subsumes A_2, then *is-*a_3 will be inferred, and correspondingly, *is-*a_4 will be inferred in OWL. Therefore *is-*a_3 (be it an assertion or an inference) will have a corresponding *is-a* in the OWL ontology, and thus subsumption links will be preserved.

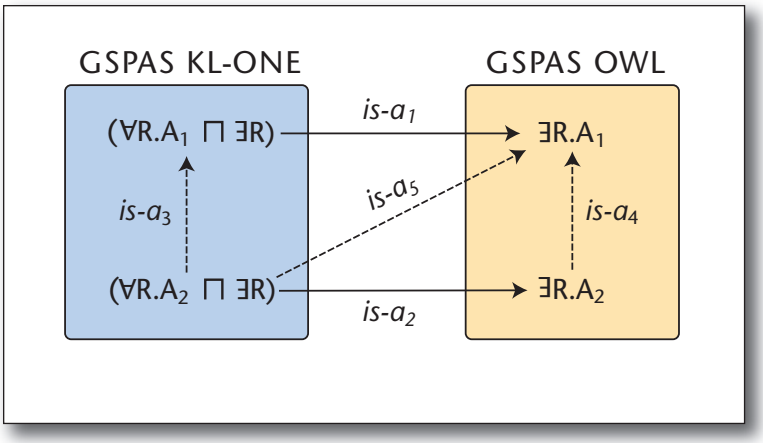


Figure 5. Translation of Value Restriction.

Next, we offer three reasons for choosing existential restrictions over value restrictions: (1) Between KL-ONE and OWL there is a paradigm shift. OWL ontologies use variants of existential restriction to model common use cases found in real-world ontologies. (2) It reduces the computational complexity of the resulting ontology. (3) It tends to reduce the number of base terms in the ontology. For example, we can model car owners in two ways. First, using value restriction a car owner is someone whose owns-car role is filled only by cars (∀owns-car.Car ⊓ ∃owns-car), and second, using existential restriction it is someone whose owns role is filled with a car (∃owns.Car). For ship owners, we get (∀owns-ship.Ship ⊓ ∃owns-ship) and (∃owns.Ship), respectively. The first model uses different roles to describe different owner concepts, whereas the second model uses just one role (owns) for that purpose.

In this section, we have presented a structure-preserving mapping between GSPAS and OWL primitives. Based on this mapping, the KR language of the new OWL ontology is:

$A \sqsubseteq C \quad A \equiv C$
$C \rightarrow A_1 \quad C \rightarrow \exists R.A_2 \quad C \rightarrow C_1 \sqcap C_2$

Reasoning

The GSPAS classifier, a derivative of the KL-ONE classifier (Lipkis 1981), uses structure matching to compute subsumptions, whereas OWL reasoners use logic-based tableau algorithms for this purpose. It is known that structural subsumption is sound but incomplete with respect to logical subsumption (Baader et al. 2003); in fact, structure matching is complete only for a small subset of OWL-DL (Khemani 2013; Brachman and Levesque 2004); that is, for a given knowledge base, logical subsumption will find all inferences that structural subsumption can find and possibly more. Moreover, the mapping from GSPAS KR language to the new KR language preserves subsumption links. Therefore, we conclude that each subsumption link in the GSPAS ontology will have a corresponding link in the OWL ontology. Further, a GSPAS concept will be a subclass of the corresponding OWL class.

Furthermore, the new OWL ontology allows domain restriction, range restriction, and subroles:

$domain(R) \sqsubseteq A1 \quad range(R) \sqsubseteq A2 \quad R \sqsubseteq S$

where R, S are role names, and A_1, A_2 are concept names. Now, the profile of the new OWL ontology is a subset of $\mathcal{EL}^{++}$ profile (Baader, Brandt, and Lutz 2008; Motik et al. 2012), which in turn is a subset of OWL-DL profile. $\mathcal{EL}^{++}$ runs in polynomial time for common reasoning tasks. We experimented with other DL profiles and selected $\mathcal{EL}^{++}$ because it provides a good balance between expressiveness and performance for the GSPAS ontology.

Ontology Design and Organization

The GSPAS ontology supports two use cases (figure 3): to parse build-steps written in Standard Language, and to interpret parsed build steps. As a result, there are two sets of terms in the ontology — one that describes words in the Standard Language and the other that describes build steps, parts, tools, and so on. All terms reside in a common namespace, and a term is identified by its name (label).

Ontology Organization

Each term (concept, individual, role, or attribute) in the new ontology is assigned a namespace, a label, and a unique identifier. The unique identifier[1] is generated from the namespace[2] and label. Namespaces have a hierarchical structure, which allows top-down organization of the ontology to arbitrary depth.

The new ontology is divided into subject areas, namely, language and manufacturing. Each subject area is divided into smaller areas (like verbs, parts, tools, and others), and so on to arbitrary depth. One or more namespaces are used to organize a subject area. Figure 6 shows the differences between the GSPAS ontology and the new ontology.

Ontology Design

The various concept types, role types, and modeling choices (like entity as concept versus individual, binary relation as subclass-relation versus role versus annotation property, and others) and the various hierarchies (lexical hierarchy, operations, parts, tools, and others) in the GSPAS ontology are mostly stable and are retained as such in the new ontology. We reused the working parts of the design and remodeled only the problematic cases. Here, we describe how the new ontology models three interesting problems: homonyms (one-spelling, many-meanings), synonyms (many-spellings, one-meaning), and part-of-speech information.

Homonyms

Terms in the GSPAS ontology reside in a single namespace, and a term is identified by its name (label). As a result, a term like HAMMER that occurs as a lexical term, a tool, and an operation will have a single representation overloaded with three meanings. Such

terms will cause interleaving of unrelated hierarchies and produce spurious inferences. For example, given that HAMMER is a TOOL and HAMMER is also an OPERATION, if POWER-HAMMER is a HAMMER, then POWER-HAMMER becomes a TOOL as well as an OPERATION. The latter inference is spurious.

Homonyms can cause incorrect descriptions; for example, a concept can be either primitive or defined; if HAMMER as a tool is a primitive concept, and as an operation it is a defined concept, then choosing either type will lead to incorrect description.

Homonyms can also cause punning. OWL-DL requires the identifiers of objects, classes, and properties to be mutually disjoint. Punning is the result of violating this constraint. For example, prepositions like USING and WITH occur as concepts in the language ontology and as properties in the manufacturing ontology.

The new ontology adopts the one term, one meaning (OOM) principle, where a new term will carry only one meaning. Therefore, each sense of a homonym will be independently represented. Thus, HAMMER will split into three terms, each with a single meaning and a distinct namespace.

lex:HAMMER opr:HAMMER tool:HAMMER

This solves the homonym problem. Now, homonyms will have matching labels but different IRIs and will not cause spurious inferences.

Synonyms

In the GSPAS ontology, name variations (like synonyms, acronyms, abbreviations, misspellings, regional variations, names given by external sources, and others) are treated as synonyms (call them GSPAS synonyms). GSPAS synonyms are stored as data values in the associated term and so the classifier does not process them. The same approach is used in the new ontology where GSPAS synonyms are stored in OWL annotation property. Next, we present an alternative approach and give reasons for rejecting it.

GSPAS synonyms of classes and objects can be modeled using the predefined properties owl:equivalentClass and owl:sameAs, respectively. Now, GSPAS synonyms become logical terms and the classifier will process them. This has some side effects. First, we cannot tell apart a term and its synonym because both are first-class terms; this is not wrong, but the synonym relation goes out of sight. Second, the synonym relation is neither symmetric nor transitive, but owl:equivalentClass and owl:sameAs are both symmetric and transitive and so will induce spurious synonym relationships. Third, the GSPAS synonyms become new terms and may cause homonym problems. This can be solved at the expense of introducing spurious homonyms (matching labels but different IRIs). For these reasons we reject this approach and treat synonyms as data values.

Part-of-Speech Information

In GSPAS ontology, part-of-speech (POS) information is modeled in two ways: POS tags (like noun, verb, and others) appear as concepts in the taxonomy (so words in Standard Language can specialize them), and POS tags are stored as data values in a nondefinitional attribute. In the new ontology, we model POS tags as concepts in the taxonomy. The tags stored in the attributes are remodeled into the taxonomy by creating suitable POS concepts and subsumption links.

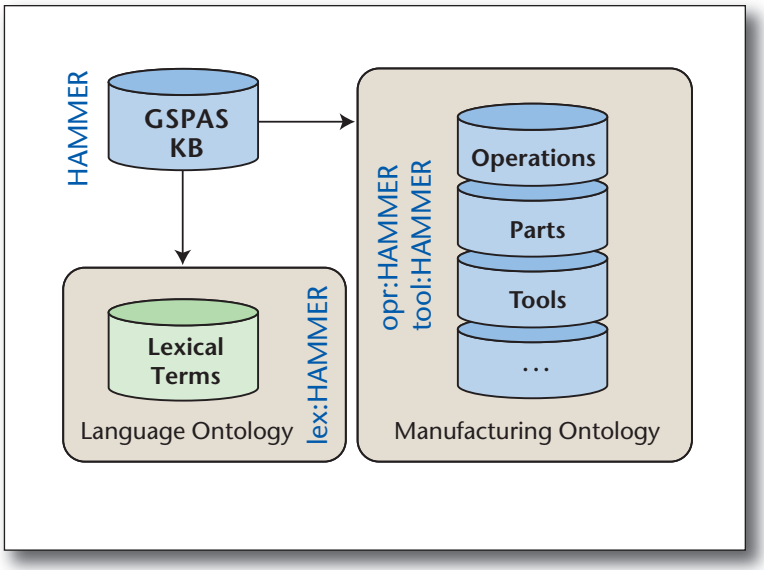


Figure 6. Reengineered Ontology.

Ontology Conversion

HAMMER has three senses: As an OPERATION it operates on an OBJECT restricted to HAMMERABLE type, and as a TOOL its SIZE is restricted to HAMMER-SIZE. In the interest of space we will ignore the lexical sense of HAMMER.

$$\text{HAMMER} \sqsubseteq \text{OPERATION} \sqcap \text{TOOL} \sqcap (\forall \text{OBJECT}.\text{HAMMERABLE} \sqcap \exists \text{OBJECT}) \sqcap (\forall \text{SIZE}.\text{HAMMER-SIZE} \sqcap \exists \text{SIZE})$$

Conceptually, ontology conversion takes a GSPAS term description and creates one or more new descriptions after resolving homonyms and implementing the various design choices. For the case of hammer, our goal is to split its description into two new descriptions:

$$\text{HAMMER}_{\text{opr}} \sqsubseteq \text{OPERATION}_{\text{opr}} \sqcap \exists \text{OBJECT}_{\text{opr}}.\text{HAMMERABLE}_{\text{obj}}$$

$$\text{HAMMER}_{\text{tool}} \sqsubseteq \text{TOOL}_{\text{tool}} \sqcap \exists \text{SIZE}_{\text{tool}}.\text{HAMMER-SIZE}_{\text{tool}}$$

where each new term is assigned a single namespace that is denoted by its subscript, the left side of a description is a name, and the right side is an expression that refers to other term descriptions in the ontology.

Technically, the GSPAS ontology conversion reduces to the problem of assigning one or more

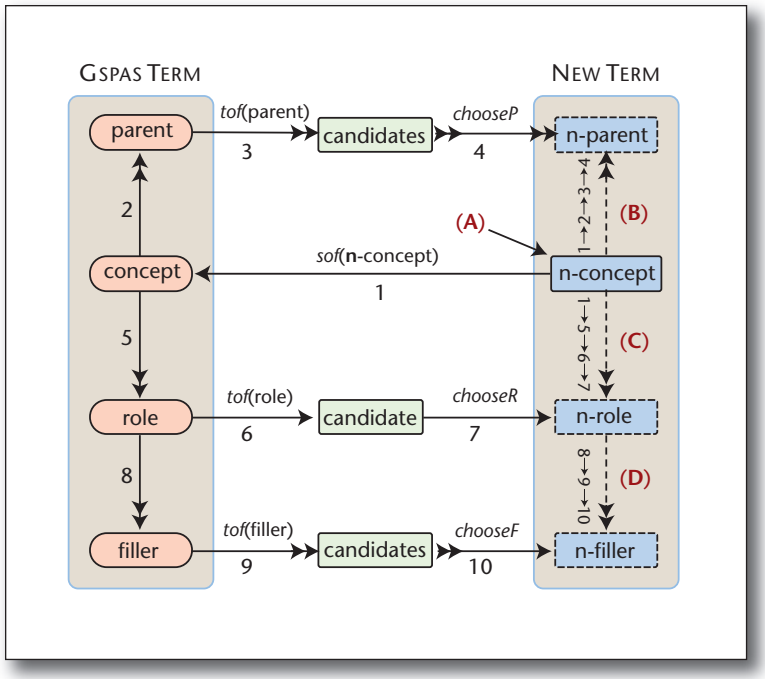


Figure 7. Conversion Work Flow.

Numbers indicate flow sequence. Nodes are sets; edges are functions. A double arrowhead indicates a set valued input/output. The items to be computed are in dashed lines.

namespaces to each name in a description and then extracting new descriptions. The description of HAMMER after namespace assignment is shown below; from this, $HAMMER_{opr}$ and $HAMMER_{tool}$ will be extracted after resolving namespace ambiguity.

$$HAMMER_{opr,tool} \sqsubseteq OPERATION_{opr,lex} \sqcap TOOL_{tool,lex} \sqcap (\forall OBJECT_{opr}.HAMMERABLE_{obj} \sqcap \exists OBJECT_{opr}) \sqcap (\forall SIZE_{tool}.HAMMER\text{-}SIZE_{tool} \sqcap \exists SIZE_{tool})$$

In the presence of namespace ambiguity, ontology conversion becomes an inverse problem and so it has several solutions. The corresponding forward problem is to recover the GSPAS ontology from the new ontology, that is, drop the namespaces and merge the descriptions. The conversion is lossless if the GSPAS ontology can be fully recovered from the new ontology. To choose the correct description of $HAMMER_{opr}$ and $HAMMER_{tool}$, we need a set of rules, also called choice functions, that will depend on the list of homonyms, list of namespaces, and the organization of GSPAS ontology.

In what follows, we describe the conversion process (figure 7) with the help of term-mapping functions and choice functions. In figure 7, *parent* denotes a named parent concept, *role* denotes a role name, and *filler* denotes a value restriction (which is a concept name). For the concept HAMMER, parents are {OPERATION, TOOL}, roles are {OBJECT, SIZE}, and filler of OBJECT is {HAMMERABLE}. The term-mapping functions track the link between GSPAS terms and new terms: *tof* (target-of) maps a GSPAS term to a set of new terms, and *sof* (source-of) maps a new term to a GSPAS term.

$$tof(HAMMER) = \{HAMMER_{opr}, HAMMER_{tool}\}$$
$$sof(HAMMER_{opr}) = HAMMER$$

The choice functions are used to resolve homonyms and select admissible terms. Given a new concept, *chooseP* takes candidate parents and returns the admissible parents; similarly, *chooseR* takes candidate roles and returns the admissible roles, and further, *chooseF* returns the admissible fillers for a new concept-role pair. Given $HAMMER_{opr}$, *chooseP* takes $\{OPERATION_{opr}, OPERATION_{lex}\}$ and returns $\{OPERATION_{opr}\}$, similarly, *chooseR* takes $\{OBJECT_{opr}, SIZE_{tool}\}$ and returns $\{OBJECT_{opr}\}$. Given $HAMMER_{opr}$ and $OBJECT_{opr}$, *chooseF* takes $\{HAMMERABLE_{obj}\}$ and returns $\{HAMMERABLE_{obj}\}$.

Ontology conversion creates new descriptions by making several passes over the GSPAS ontology: (Step A) it first creates new terms, with empty descriptions, (Step B) then adds parents to the newly created terms, (Step C) then adds roles, (Step D) and finally role fillers (value restrictions). (See Listing 1.)

Step A.

To create a new term we need a namespace and a label. First, we identify the namespaces of the new ontology then we assign GSPAS terms to namespaces. Homonyms will show up in multiple namespaces. Now, we create one new term for each GSPAS term and its namespace combination, and we track this association using *sof* and *tof* functions (listing one). At this point we will have new terms with empty descriptions; each new term will link to one GSPAS term, and each GSPAS term will link to one or more new terms. Use *sof* and *tof* to complete the rest of the conversion process.

Step B.

To populate new parents, follow the edges 1, 2, 3, 4 in figure 7. For each new concept and its GSPAS parent, fetch the candidate parents, if a GSPAS parent is a homonym, it will return multiple candidates. Now, select the admissible parents and add them to the new concept (listing one).

Step C.

To populate new roles, follow the edges 1, 5, 6, 7 in figure 7. For each new concept and its GSPAS role, fetch the candidate roles, which will be a singleton set because GSPAS roles have only one meaning. Now, select the admissible role, and add it to the new concept (listing one). Now, populate attributes in a similar manner.

Step D.

To populate role fillers, continue from the previous step and follow the edges 8, 9, 10 in figure 7. For a GSPAS role and its GSPAS filler, fetch the candidate fillers. Now, select the admissible fillers, and add it to the new role in new concept (listing one). Add selected fillers to new concept. Now populate attribute fillers in a similar manner.

At the end of step D, all term descriptions are complete and we have a reengineered namespace-aware ontology that is ready for lexical and syntactic translation.

In the conversion process, namespace assignment and the choice functions are two important decision points, and the remaining is routine processing. The choice functions use a set of cascading rules to disambiguate terms. Given a concept and a set of candidate parents, *chooseP* returns the parents from the concept's namespace; otherwise it returns the parents that have a preference to children from the concept's namespace, and otherwise it returns the candidate set.

For each role, its namespace and the namespaces in which it can be used are determined during the design phase. Also, its domain and range are predetermined. Given a concept and a candidate role, *chooseR* returns the role if it is admissible in that concept's namespace.

Given a concept, a role, and a set of candidate fillers, *chooseF* filters the candidate list progressively until only one candidate is left. First, it selects fillers that are subtypes of the role's range, next it selects fillers from the concept's namespace, and finally it selects fillers from the role range's namespace.

The choice functions and their rules were determined by profiling the GSPAS ontology and by experimentation. These rules are specific to GSPAS ontology, its design and organization, and the choice of namespaces and homonyms. These rules were tuned to the ontology instance that was used for final conversion and testing.

```
    // Step A: Create new terms.
1   for each ns in Namespaces
2       ns-terms = identify all terms that belong to ns
3       for each term in ns-terms
4           n-term = create-new-term(ns, term)
5           sof (n-term) = term
6           tof (term) = tof(term) ∪ {n-term}

    // Step B: Populate new parents.
7   for each n-concept
8       concept = sof(n-concept)                    // 1
9       for each parent of concept                  // 2
10          candidates = tof(parent)                // 3
11          n-parents = chooseP(candidates)         // 4
12          add n-parents to n-concept

    // Step C: Populate new roles.
13  for each n-concept
14      concept = sof(n-concept)                    // 1
15      for each role of concept                    // 5
16          candidates = tof(role)                  // 6
17          n-role = chooseR(candidates)            // 7
18          add n-role to n-concept

            // Step D: Populate new fillers.
19          for each filler of role                 // 8
20              candidates = tof(filler)            // 9
21              n-fillers = chooseF(candidates)     // 10
22              add n-fillers to n-role of n-concept
```

Listing 1.

Verification

Verification is done at three levels: framework level, ontology level, and application level.

At the framework level, (1) we verified the correctness of framework mapping (table 2) by first comparing the asserted hierarchies of the new and GSPAS ontologies, and then by comparing the respective inferred hierarchies. The new asserted hierarchy had four missing subsumption links (out of 12,600+ direct links); these were manually added to the OWL ontology. Next, we manually compared the inferred hierarchies; most of the hierarchy matched; there were about 20 cases where a subconcept became equivalent to its parent. These cases were manually corrected in the new ontology. (2) Further, we verified the profile of the new ontology. We used Pellet info tool to compute OWL and DL profiles of the new ontology. It turned out to be OWL 2 EL and $\mathcal{EL}^{++}$ (see table 4) as expected.

At the ontology level, (1) we verify that every GSPAS term has a representation in the new ontology and that every new term description is part of some GSPAS term description. This is done by a reverse transformation from the new ontology to GSPAS ontology, by dropping the namespaces and merging terms. We manually compared the two versions of GSPAS ontology and found no significant differences. This verification alone does not establish the validity of the new ontology, but checks whether the conversion is lossless. It is a good first line of defense and helps in accounting for terms in the new ontology. (2) Further, we checked for the case of punning using the Pellet lint tool, and found one violation, which was fixed manually.

The application-level verification provides the final validation of the new ontology. It is discussed in the Deployment and Maintenance section.

Performance Testing

In the GSPAS ontology, all terms are modeled as concepts, but primitive concepts that occur as leaves in the taxonomy, and without any role restriction, qualify as individuals. To explore alternate models of GSPAS ontology, qualifying individuals in the part-of-speech hierarchy and object hierarchy were modeled as individuals.

We created five OWL ontologies from GSPAS ontology (see table 3). Each differs in the number of individuals it models. (1) LEX_1 is the language ontology where leaves are individuals. (2) ONT_1 is the full ontology where all terms are concepts. (3) ONT_2 is ONT_1 with lexical leaves as individuals. (4) ONT_3 is ONT_2 with object leaves as individuals. (5) ONT_4 is

Case	Individuals	Individuals	Classes
LEX_1	lex leaves	6,780	317
ONT_1	none	0	12,815
ONT_2	lex leaves	5,679	7,136
ONT_3	lex and obj leaves	6,898	5,917
ONT_4	lex leaves minus nominals	5,136	7,679

Table 3. Ontology Test Cases.

	Language Profile		Classification Time		
Case	OWL	DL*	FacT++	HermiT	Pellet
LEX_1	OWL 2 EL	AL	0.2	0.8	0.7
ONT_1	OWL 2 EL	ALEH	1.6	12	4
ONT_2	OWL 2 EL	ALEHO	2.3	74	564
ONT_3	OWL 2 EL	ALEHO	2.7	352	716
ONT_4	OWL 2 EL	ALEH	1.7	13	4

Table 4. Classification Time (in Seconds).

In DL profile, AL stands for attributive language, E for existential restriction, H for subrole, and O for nominals.

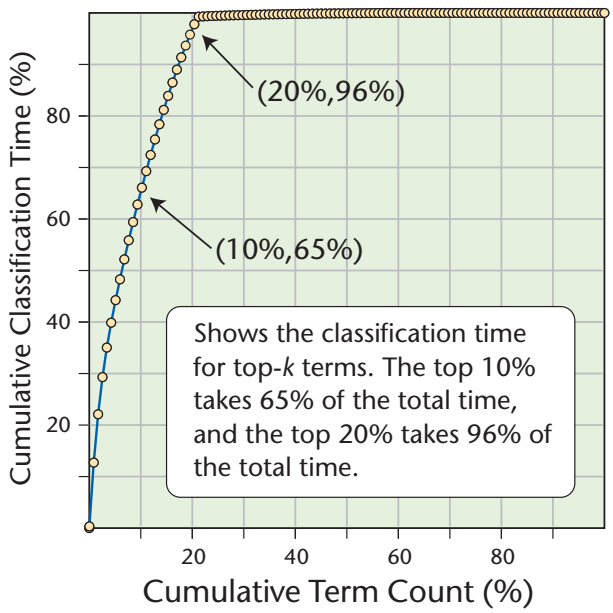


Figure 8. Pareto Chart. Time Versus Terms for ONT_3.

ONT_2 with nominals rolled back to concepts. The first four cases were created for performance testing, The fifth one was the result of performance tuning.

We tested three reasoners (FacT++ v1.6.3, Pellet v2.2.0, and HermiT v1.3.8.) on the five ontologies using Protégé v4.3.0 on Intel i7-4770 with 16 GB RAM running 64-bit Ubuntu 12.04. The execution times are given in table 4. We make the following observations: (1) Of the reasoners, FacT++ has the best overall performance, followed by HermiT and Pellet. (2) Of the ontologies, LEX_1 has the best overall performance, it has a 1:21 class to individual ratio; and ONT_1 has good overall performance and has no individuals. (3) The performance, though within acceptable limits, begins to degrade for ONT_2 and ONT_3. HermiT and Pellet are up to two orders of magnitude slower than FacT++ for these ontologies.

To understand where the reasoner was spending time, we profiled ONT_3 using Pellet[3] and computed the classification time for each concept. Using this, a Pareto chart was prepared; see figure 8. Observe that 96 percent of the reasoner's time is spent in classifying 20 percent of the terms.

We analyzed these terms and found that most of these had owl:hasValue restriction in its definition. To verify the impact of owl:hasValue on performance, we created ONT_4 from ONT_2 by changing fillers of owl:hasValue into concepts and rewriting owl:hasValue as existential restriction. Now, ONT_4 outperforms ONT_2 and ONT_3, and has a comparable performance to ONT_1 (table 4).

From this we conclude that creation of individuals has less impact on performance, as seen in LEX_1, but using them in owl:hasValue restriction degrades performance, as seen in ONT_2, ONT_3. This is true for HermiT and Pellet. In our test, FacT++ consistently outperforms HermiT and Pellet, and for our ontology FacT++ is unaffected by nominals.

This performance test is solely based on execution time. We did not compare the inferences from these reasoners, so we do not know if there is any qualitative difference in the inferences from these reasoners.

Deployment and Maintenance

We (Ford) verified the completeness of the new OWL ontology by developing a tool to compare it to the KL-ONE version. The delivered OWL ontology needed to be validated and verified as the first step toward deployment. This process consisted of several steps. Initially, the OWL ontology was loaded into an Allegrograph server and we wrote various SPARQL queries to determine if the results returned were as expected. In cases where the results were not satisfactory, we then examined the ontology and made modifications if they were required. This manual validation went on for a period of several weeks until we were certain that the OWL ontology was complete and usable.

The next phase of the validation process utilized

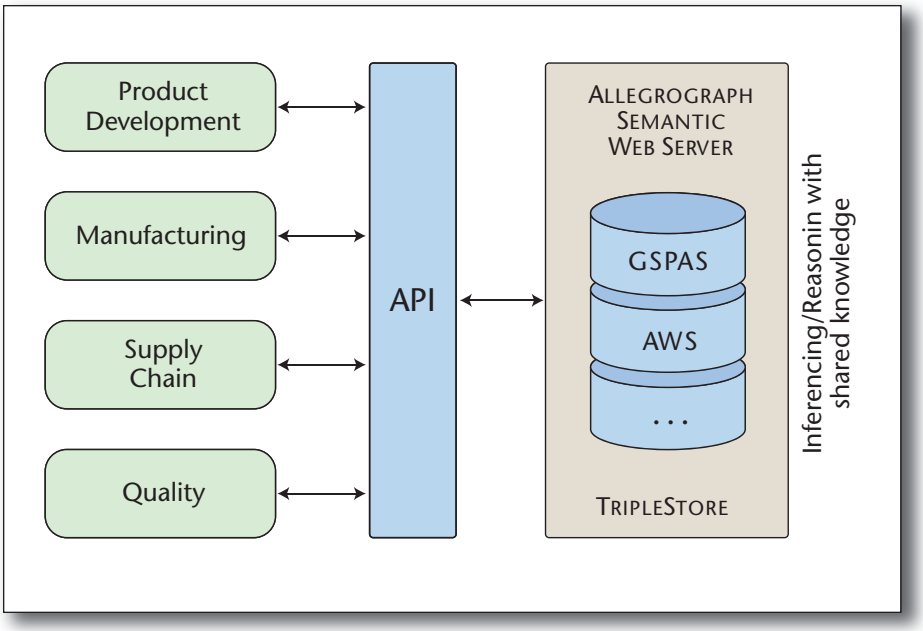


Figure 9. Ford Semantic Web Framework.

an automated set of regression tests that were run against the new OWL ontology. This is a set of more than 1000 use cases that access the OWL ontology to parse and process the assembly build instructions. In this case, we replaced the KL-ONE ontology with the OWL ontology and ran the entire suite of regression tests and compared the results with the baseline. As with the manual tests we found a number of differences that needed to be analyzed and addressed. These differences fell into the following categories. First, OWL representation was different than KL-ONE but was part of the reengineering process. In this case we adjusted the regression tests to reflect how the knowledge was represented in OWL. Second, discrepancies were caused because of formatting, punctuation, special characters, and related syntax errors. In these cases, we wrote a routine that would fix these errors as part of the OWL retrieval process, but our intention is to go back and fix these in OWL. Third, in some cases, the OWL representation was not what we wanted. In this case we went back to OWL and made the appropriate fixes.

At this point we were confident that the lexical ontology was fairly complete and would be usable after the changes made above were completed.

The next step was to build an image using the new OWL ontology and deploy it for user acceptance testing. This testing pointed out some performance issues that were addressed by rewriting the code to make the OWL interface work more efficiently. After these performance issues were fixed the new AI system with the OWL ontology was deployed into the testing environment. No other major issues were discovered during the user-acceptance testing phase and the application with the embedded lexical OWL ontology was deployed for use.

We were able to take advantage of the extensibility of the OWL ontology by developing a script that could load a class of parts known as wire assemblies directly from an external database. This allows us to add additional knowledge into OWL much more quickly. Another of the main advantages of using OWL was the capability to use standard tools for ontology maintenance such as Top Braid Composer, which provides additional capability. The OWL/RDF system has proven to be easier to maintain and utilize for reusing knowledge.

The OWL ontology is also available for use through Allegrograph and is being utilized by other applications that need the information. Figure 9 shows the structure of our semantic web architecture.

Conclusions and Future Steps

In this article we described a project where Ford collaborated with the Indian Institute of Technology Madras to reengineer and convert an existing ontology into a semantic web OWL/RDF architecture. There were a number of compelling reasons that motivated the reengineering of the ontology from KL-ONE to OWL. The most important ones were based on maintainability and extensibility. The original software was written before any software tools for ontology maintenance were available. The KL-ONE ontology could only be maintained using a specialized tool. This tool had to be rewritten several times as operating systems and hardware were being upgraded, and it was becoming a bottleneck for future ontology development. It was extremely tedious and time consuming to manually create reports and to extract knowledge from the KL-ONE ontology. In the meantime business requirements for the ontology were rapidly increasing and the existing architecture could not support them. The conversion of the ontology to OWL was a critical requirement for the future usage of the AI application. Our experience was somewhat unique in that we have been using KL-ONE since the 1990s and much of the work in semantic web had taken place after we had a deployed application.

The conversion from KL-ONE to OWL required a significant amount of work, but the advantages from moving into a semantic web architecture made this a worthwhile investment. It enables us to take advantage of existing tools and processes and to make our ontology reusable and extensible using existing standards. Queries can easily be developed using SPARQL, which allow other applications to access our ontology.

The semantic web infrastructure also gives us the capability to link to other ontologies and take advantage of the linked open data world. Therefore, the return on investment for this project

includes a number of benefits that will pay dividends in the future. The standards and tools built around semantic technologies make our ontology easily accessible to other applications and will reduce future expenses in terms of maintenance and development costs. In addition, this project has helped us build the infrastructure needed to support semantic technology and allow for the development of other projects that could benefit from the semantic web.

Our future work will include the deployment of other ontologies into production as well as the use of semantic web tools and semantic web architecture for ontology development and maintenance. However, the real benefit will occur as we leverage semantic technology across other areas of the company and integrate this into our development and manufacturing processes.

Notes

1. International Resource Identifier (IRI).

2. International Resource Identifier (IRI).

3. In Pellet, concept classification is done by a series of subsumption tests. Pellet reports the execution time for each test, and we sum up these times to compute the classification time for a concept.

References

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge, UK: Cambridge University Press.

Baader, F.; Brandt, S.; and Lutz, C. 2008. Pushing the EL Envelope Further. In Proceedings of the Fourth OWLED Workshop on OWL: Experiences and Directions. *Ceur Workshop Proceedings* Volume 496. Aachen, Germany: RWTH Aachen University.

Bechhofer, S.; van Harmelen, F.; Hendler, J.; Horrocks, I.; McGuinness, D. L.; Patel-Schneider, P. F.; and Stein, L. A. 2004. OWL Web Ontology Language Reference. W3C Recommendation. Cambridge, MA: World Wide Web Consortium, W3C. (www.w3.org/TR/2004/REC-owl-ref-20040210).

Brachman, R., and Levesque, H. 2004. *Knowledge Representation and Reasoning*. San Francisco: Morgan Kaufmann Publishers Inc.

Brachman, R. J., and Schmolze, J. G. 1985. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science* 9(2): 171–216.

Corcho, O., and Gómez-Pérez, A. 2005. A Layered Model for Building Ontology Translation Systems. *International Journal of Semantic Web Information Systems* 1(2): 22–48.

Euzenat, J. 2001. Towards a Principled Approach to Semantic Interoperability. In Proceedings of the IJCAI Workshop on Ontologies and Information Sharing, 19–25. *Ceur Workshop Proceedings* Volume 47. Aachen, Germany: RWTH Aachen University.

Khemani, D. 2013. *A First Course in Artificial Intelligence*. Nolda, India: McGraw Hill Education.

Lipkis, T. A. 1981. A KL-ONE Classifier. In *Proceedings of the Second KL-ONE Workshop*. BBN Technical Report 4842, Cambridge, MA: BBN Laboratories.

Motik, B.; Grau, B. C.; Horrocks, I.; Fokoue, A.; and Wu, Z. 2012. OWL 2 Web Ontology Language Profiles, 2nd ed. W3C Recommendation. Cambridge, MA: World Wide Web Consortium, W3C. (www.w3.org/TR/2012/REC-owl2-profiles-20121211).

Rychtyckyj, N. 1994. Classification in DLMS Utilizing a KL-ONE Representation Language. In *Proceedings of the Sixth International Conference on Tools with Artificial Intelligence,* ICTAI'94, 339–45. Los Alamitos, CA: IEEE Computer Society.

Rychtyckyj, N. 1999. DLMS: Ten Years of AI for Vehicle Assembly Process Planning. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence*, 821–828. Menlo Park, CA: AAAI Press.

Rychtyckyj, N. 2006. Standard Language at Ford Motor Company: A Case Study in Controlled Language Development and Deployment. Paper presented at the 5th International Workshop on Controlled Language Applications, Cambridge, MA, August 12.

Schmolze, J. G., and Lipkis, T. A. 1983. Classification in the KL-ONE Knowledge Representation System. In *Proceedings of the Eighth International Conference on Artificial Intelligence,* 330–332. Los Altos, CA: William Kaufmann, Inc.

Sullivan, B.; Carey, P.; and Farrell, J. 2001. H*eyde's Modapts: A Language of Work*. London: Heyde Dynamics Pty, Ltd.

Nestor Rychtyckyj is a senior analytics scientist for global data insight and analytics at Ford Motor Company in Dearborn, Michigan. His responsibilities include the application of machine learning, natural language processing, semantic computing, and machine translation for manufacturing, quality, customer interaction and cybersecurity. Previously, Rychtyckyj was responsible for the development and deployment of AI-based systems for vehicle assembly process planning and ergonomic analysis in manufacturing. He received his Ph.D. in computer science from Wayne State University in Detroit, Michigan. Rychtyckyj is a senior member of AAAI and IEEE and a member of ACM.

Venkatesh Raman is a senior data analyst for global data insight and analytics at Ford Motor Pvt. Ltd. in Chennai, India. His responsibilities include leveraging the big data platform and tools for analyzing and applying machine learning to connected vehicle data. Previously, Raman was with the Enterprise Technology Research group wherein he was researching the big data domain and evangelizing it. He received his master's degree in computer science from MS University in India.

Baskaran Sankaranarayanan is a researcher in the Department of Computer Science and Engineering, IIT Madras, India. He has more than 10 years of industry experience in designing, developing, and deploying large-scale data cleansing, data integration, and OLAP applications for retail, banking, financial services, health care, credit rating, and magazine domains. He is interested in the application of ontology to real-world problems. His long-term goal is to develop efficient data integration frameworks. He holds a master's degree in structural engineering from IIT Bombay, and a bachelor's degree in civil engineering from University of Madras.

P. Sreenivasa Kumar is a professor in the Department of Computer Science and Engineering (CSE), IIT Madras, India. He was also the head of the Computer Science and Engineering Department during the years 2013–2015. His research interests include database systems, semistructured data and XML, ontologies and semantic web, data mining, graph algorithms, and parallel computing. He earned his bachelor's degree in electronics and communication engineering from the Sri Venkateswara University College of Engineering, Tirupati, India. His master's and Ph.D. degrees are in computer science from the Indian Institute of Science, Bangalore, India.

Deepak Khemani is a professor in the Department of Computer Science and Engineering, IIT Madras, India. His long-term goal is to build articulate problem-solving systems that can interact with humans, currently looking at contract bridge. He works in memory-based reasoning, knowledge representation, planning, constraint satisfaction, and qualitative reasoning. He graduated with three degrees from IIT Bombay, including two in computer science. He is the author of *A First Course in Artificial Intelligence*.

Automated Volumetric Intravascular Plaque Classification Using Optical Coherence Tomography

Ronny Shalev, Daisuke Nakamura, Setsu Nishino, Andrew M. Rollins, Hiram G. Bezerra, David L. Wilson, Soumya Ray

■ *An estimated 17.5 million people died from a cardiovascular disease in 2012, representing 31 percent of all global deaths. Most acute coronary events result from rupture of the protective fibrous cap overlying an atherosclerotic plaque. The task of early identification of plaque types that can potentially rupture is, therefore, of great importance. The state-of-the-art approach to imaging blood vessels is intravascular optical coherence tomography (IVOCT). However, currently, this is an offline approach where the images are first collected and then manually analyzed an image at a time to identify regions at risk of thrombosis. This process is extremely laborious, time consuming, and prone to human error. We are building a system that, when complete, will provide interactive three-dimensional visualization of a blood vessel as an IVOCT is in progress. The visualization will highlight different plaque types and enable quick identification of regions at risk. In this article, we describe our approach, focusing on machine-learning methods that are a key enabling technology. Our empirical results using real OCT data show that our approach can identify different plaque types efficiently with high accuracy across multiple patients.*

Imaging techniques are a key tool in the diagnosis of disease. X-rays, ultrasound, CAT, and PET scans are now routinely used as a preliminary step to determine the extent of a disease and the need for and type of treatment (Tearney et al. 2006). These techniques generate vast quantities of data. The images that are produced must typically be analyzed by trained clinicians. This is extremely labor intensive, expensive, and can be prone to error. Thus, there is a need for, and an opportunity to, improve the quality of healthcare systems by developing automated aids to assist in this process. Given the patient-critical outcomes of the image-analysis process, a human analyst must always remain in the loop. However, it may be possible to reduce the labor involved, and thereby the costs to the patient, using such systems. Further, a well-designed system may also reduce errors, potentially saving lives.

There are rarely well-defined, crisp guidelines that can be used, for example, to separate healthy tissue from diseased. Therefore, such image analysis tasks are often formulated as machine-learning problems. Here, a collection of images annotated by experts is used as data to train a classifier, which is then used to help annotate a new image. Typically, this system will present a rank-ordered list of image regions to the human analyst for verification or correction.

In our work, we focus on image analysis for coronary artery disease (CAD). This is a leading cause of death worldwide. An estimated 17.5 million people died from a cardiovascular disease in 2012, representing 31 percent of all global deaths. Of these deaths, an estimated 7.4 million were due to coronary heart disease and 6.7 million were due to stroke (Mendis, Puska, and Norrving 2011). Although this is such a common disease, the underlying causes are quite complex, and it is only recently that an imaging technique that can help identify the disease mechanism clearly has been developed. This tech-

 ISSN 0738-4602

nique is called intravascular optical coherence tomography (IVOCT). In the following sections, we describe in detail CAD, the prior state of the art in imaging for CAD, and the IVOCT technique.

Like many other imaging techniques, a major issue with IVOCT is that it can produce more than 500 images in a single scan, resulting in an explosion of image data. This can be difficult and labor intensive to analyze manually, taking up to one hour of examination for each image by a trained analyst, of which there are not many, given the recency of the technique (Lu et al. 2012). This often precludes measurements from every frame, and plaque classification is not done because it is infeasible in terms of time. Further, this manual process is also prone to error. In prior work (Lu et al. 2012), our group has found evidence of up to 5 percent intra- and 6 percent inter-rater variability among analysts looking at these images.

The goal of our work is to enable an effective detection and diagnosis of CAD, which is a necessary precursor for effective treatment. We aim to build a tool to do this in three ways: (1) reduce the effort involved, (2) improve the accuracy of disease mechanism identification, and (3) make the diagnosis available as early in the process as possible. The prevalence of CAD means achieving these goals can have a major impact on health worldwide.

We anticipate fulfilling our goals in two steps. In the first step, reported in this article, we develop an automated method to process a single image generated by IVOCT scans. We demonstrate that it is accurate and efficient on real IVOCT data and that analysts can use the output to greatly reduce their annotation effort. In the second step, our goal is to integrate this approach into a real-time visualization that accompanies an IVOCT scan. These images will be annotated with different detected plaque types and will be used for rapid identification of high-risk regions for intervention, management and guidance.

Cardiovascular Artery Disease (CAD)

In this section, we discuss CAD and the state of the art in its diagnosis and treatment.

The underlying disease process in the blood vessels that results in coronary heart disease (heart attack) is known as atherosclerosis. For many years, it was thought that the main cause of a heart attack was the buildup of fatty plaque within an artery leading to the heart. With time, the plaque buildup would narrow the artery so much that the artery would either close off or become clogged by a blood clot (stenosis). The lack of oxygen-rich blood to the heart would then lead to a heart attack. However, these types of blockages cause only about 3 out of 10 heart attacks (Virmani et al. 2000).

Researchers are now finding that many people who have heart attacks do not have arteries severely narrowed by plaque (Falk 1983). In fact, vulnerable plaque may be buried inside the artery wall and may not always bulge out and block the blood flow through the artery. This is why researchers began to look for, potentially, a different cause. What they found was that a thin protective fibrous cap (FC) overlying an atherosclerotic plaque (lipid pool) may rupture, triggering the formation of a blood clot, which may eventually lead to an acute event such as heart attack.

Current state-of-the-art treatment of the disease focuses on blood vessel narrowing by means of percutaneous interventions (PCIs). PCI is a procedure that uses a catheter (a thin flexible tube) to place a "stent" to open up blood vessels in the heart that have been narrowed by plaque buildup (stenosis). A stent is a flexible tube that reinforces the blood vessel wall. This needs significant imaging support to determine how, where, or even if it should be done. For example, the presence of significant calcification in the vessel may prevent the stent from being placed or from functioning as intended, triggering additional procedures to remove the calcium or aborting the procedure. On the other hand, if there is a lipid pool that may rupture, a physician can extend a stent to seal off the affected area or at least avoid placing the stent edge in a lipid region, an occurrence that raises the risk of a tear or damage to the inner wall or lining of an artery. These examples highlight the need for a reliable imaging technique with suitable resolution to identify plaque at high resolution (for example, thickness of vulnerable fibrous cap $<<65\ \mu m$).

The current standard intravascular imaging modality is intravascular ultrasound (IVUS). IVUS is a medical imaging methodology that uses a catheter with a miniaturized ultrasound probe attached to the distal end of the catheter. The proximal end of the catheter is attached to an ultrasound device. The IVUS machine produces a detailed cross-sectional image of the vessel wall and plaque as a gray level intensity image. An example of the IVUS two-dimensional (2D) cross-sectional image is shown in figure 1, which shows the plaque and vessel wall from which the ultrasound wave bounces off.

When analyzing the 2D image generated by the IVUS machine, it is possible to quantify, limited to the IVUS resolution, the lipid plaque and the fibrous plaque. However, quantification of the total amount of vessel calcification by IVUS is problematic in that its resolution is low and it cannot measure the distance between the superficial calcium and the vessel boundary, nor can it assess the thickness of calcium (Mintz et al. 1995).

Intravascular Optical Coherence Tomography (IVOCT)

In this section, we introduce IVOCT and describe its

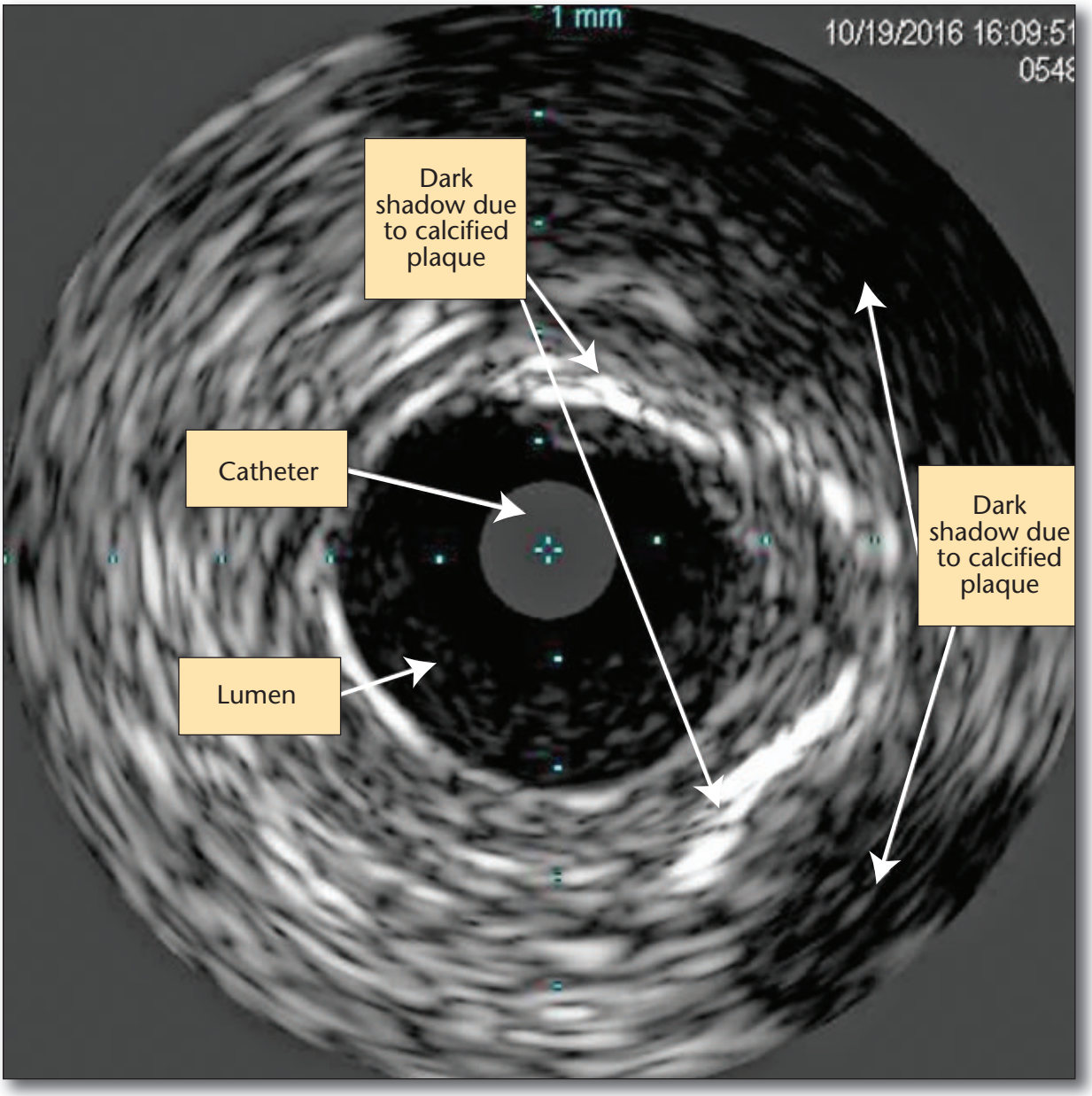


Figure 1. Intravascular Ultrasound Image.

advantages relative to IVUS. Intravascular optical coherence tomography uses the same concept of imaging, only it uses light instead of ultrasound waves. The underlying concept of OCT is similar to that of ultrasound; by measuring the delay time of optical echoes reflected or back scattered from subsurface structures in tissues, we can obtain structural information as a function of depth within the tissue (Tearney et al. 2012).

In IVOCT, we obtain cross-sectional images by inserting a flexible imaging probe (catheter) into the blood vessel to be imaged. The catheter has an optical fiber coupled to a lens and microprism. The microprism reflects the OCT beam perpendicular to the catheter's longitudinal direction and captures the light that is back scattered from that tissue (the reflected beam is referred to as an *A-Line,* figure 2a). The probe is then rotated and pulled back. This pullback creates a two-dimensional image (referred to as polar or r-θ image) by assembling successive A-lines next to each other resulting in an image shown in figure 2b. This image is then transformed to Cartesian coordinates to produce the image shown in figure 2c. A typical pullback contains 271 images covering 54 mm and an image contains 504 A-lines.

Different tissues have different qualities that influence the back reflectance. The longer the distance traveled, the longer the delay in returning to a detec-

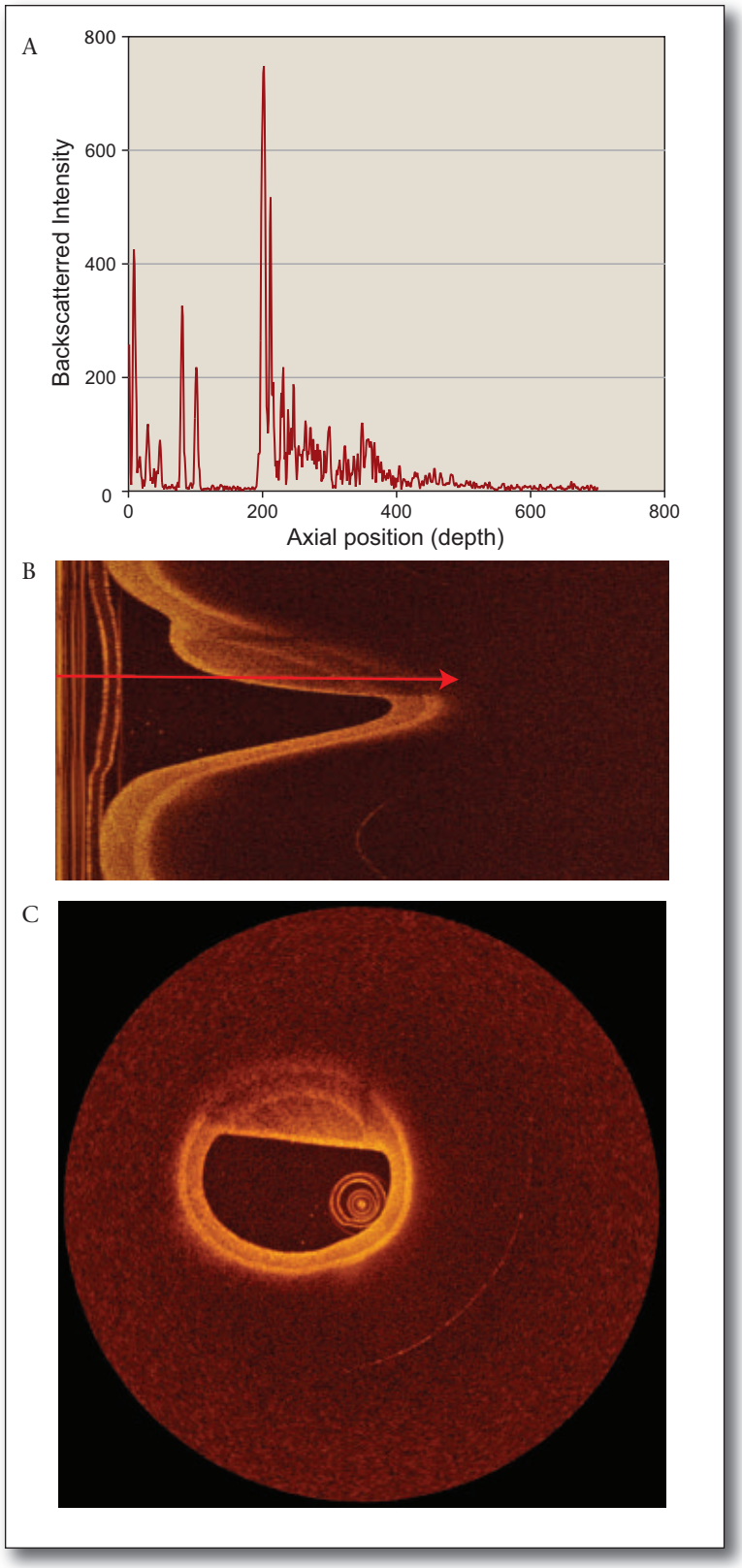


Figure 2. Intravascular OCT Image Generation Process

a) Back-scattered intensity of a single A-line. (b) Polar (r-θ) image (the red line is the A-line in a). (c) The polar image converted to the more human readable x-y.

tor. The delay in the returning light from deeper structures compared with shallow structures is used to reconstruct images.

Since its approval for clinical use, IVOCT has become an invaluable tool for vascular assessments due to its high contrast and microscopic resolution (5–15 µm), which is superior to other in vivo imaging modalities such as IVUS. To exemplify the difference in the output of IVUS and IVOCT, we show in figure 3 what a calcified cross section might look like using both modalities.

A key advantage of IVOCT over IVUS is that it is able to distinguish key types of plaque (Yabushita et al. 2002) and aid in assessment of new coronary artery stent design (Lu et al. 2012, Wang 2012). This is important because the presence of calcium is the strongest factor affecting "stent expansion," a well-documented metric for clinical outcome (Fujimoto, Nakamura, and Yokoi 2012; Nishida et al. 2013). IVOCT provides the location, circumferential extent, and thickness of calcium. Second, there can be a geographic miss, where the stent either misses the lesion along its length or is improperly expanded, affecting its ability to stabilize the lesion and/or provide appropriate drug dosage. This has a well-documented impact on recurrence of narrowing (Costa et al. 2008). Plaque dissections at the edge of a stent (when a stent's length does not fully cover the plaque along the vessel) clearly visible in IVOCT were detected by angiography in only 16 percent of cases (Chamie et al. 2013). Edge dissection (that is, when the edge of the stent lies on top of a plaque) happens almost exclusively in areas where the calcium/lipid plaque is not evenly distributed around the lumen circumference (Chamie et al. 2013), characteristics only available with intravascular imaging. Under IVOCT guidance, one can use a longer stent or apply a second stent to reduce effects of this geographic miss. Third, plaque sealing is the treatment of a lesion that may appear vulnerable and may rupture, under intravascular imaging. Because approximately 50 percent of coronary events after stenting happen at these remote, nonstented sites, plaque sealing is an attractive concept under investigation in trials. IVOCT's high sensitivity for lipid plaque will be advantageous for guidance of plaque sealing.

For the reasons mentioned, we focus on analyzing IVOCT images for CAD. Next, we describe how we represent these images in order to train a classifier from them, followed by a detailed experimental evaluation.

Representing an IVOCT Image

In order to build our system, we need to identify different plaque types in IVOCT images automatically and accurately. In this section, we describe image characteristics that are key to identifying different plaque types. In constructing our features we use the

qualitative description of the different plaques' characteristics in prior work (Yabushita et al. 2002) described below. This also provides the ability to interpret results in a meaningful way.

A *fibrous plaque* (figure 4a) has high back scattering and the region has relatively homogeneous intensity values. We see that the average intensity is high (bright). Likewise, the intensity is not attenuated much along the A-line (Gargesha et al. 2015).

A *lipid plaque* (figure 4b) is a low-intensity region with poorly delineated borders, a fast IVOCT signal drop off, and little or no OCT signal back scattering, within a lesion that is covered by a fibrous cap. We see that the intensity starts very bright and decreases quickly along the A-line (Gargesha et al. 2015).

A *calcified plaque* (figure 4c) appears as a low intensity or heterogeneous region with a sharply delineated border (leading, trailing, and/or lateral edges). Calcium is darker than fibrous plaque with greater variation in intensity level.

Based on this description, we construct a set of eight (real-valued) features for each pixel in the image. We compute these features using a three-dimensional (3D) neighborhood centered on the pixel of interest. The third dimension comes from neighboring images (human analysts will often use adjacent images when annotating an image). In these features, σ represents the standard deviation of the intensity values within a 3D neighborhood.

Distance to Lumen (D_l): This is a measure of the distance of the center pixel from the lumen border (that is, the wall of the blood vessel). This feature helps identify lipid plaques since they are typically within a fibrous plaque.

Beam Penetration (D_d): This is a measure of the length of the beam from the lumen border to the back border (the border beyond which the near infrared beam does not reach and the signal is at baseline). It depends on tissue type, thus can distinguish between plaques. This feature is invariant for pixels across an A-line but varies across A-lines.

Mean Intensity (I): This represents the average signal intensity of the different plaque types within the 3D neighborhood. As can be seen in figure 4, this is a very distinctive feature.

Homogeneity (H): This is a local coefficient of variation, σ /I. It helps in distinguishing between heterogeneous intensity regions and homogeneous intensity regions.

Relative Smoothness of Intensity (S): This is defined as $S = 1 - 1/(1 + \sigma^2)$. *S* is 0 for constant intensity regions and it approaches 1 for large deviations in intensity values.

Entropy (E): Entropy is another measure of the variability of the signal intensity within the respective plaque type regions. To compute it, we construct a histogram of the intensity distribution within a 3D neighborhood, convert it to a probability distribution, and then estimate its information content.

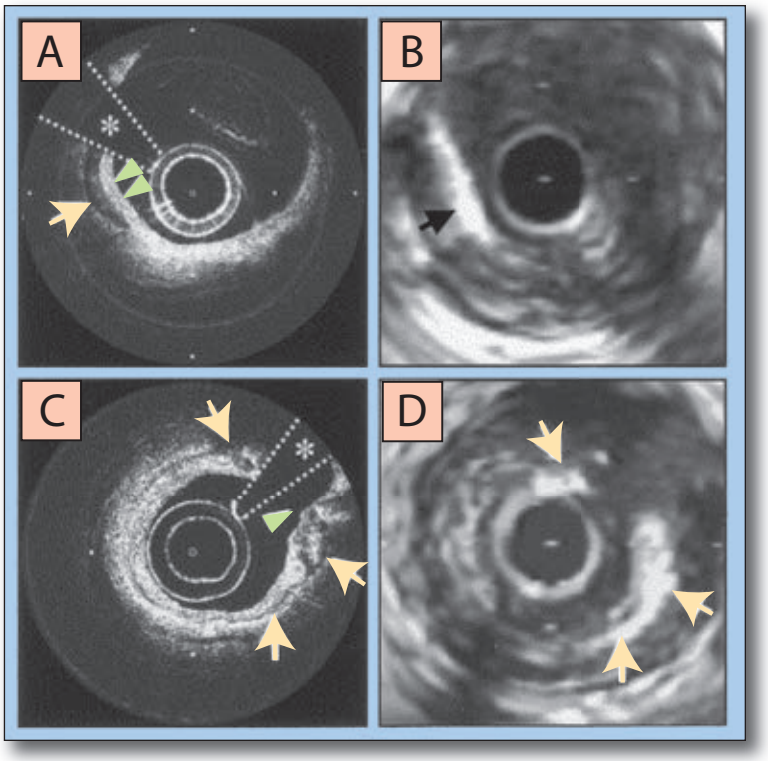


Figure 3. Calcific Coronary Plaques.

Imaged in vivo by optical coherence tomography (OCT) (A, C) and intravascular ultrasound (IVUS) (B, D). (A) This OCT image shows a well delineated, heterogeneous, signal-poor region corresponding to a macrocalcification (A, arrow), also seen in the corresponding IVUS image (B, arrow). A signal-rich fibrous band (A, two arrowheads) overlying the calcification is easily identified in the OCT image but is obscured by a saturation artifact in the IVUS image. (C) A thin layer of circumferential calcification is seen in this OCT image (arrows) as a well-defined, heterogeneous, signal-poor region within the vessel wall. A side-branch (arrowhead) can be seen adjacent to the guidewire artifact (*). (D) The extent of the calcifications (arrows) and their relation to the surrounding fibrous components of the plaque are not as clearly seen in the corresponding IVUS image. The borders of the guidewire (*) artifact are marked by dotted lines in A, C. Tick marks, 1 mm. (*Source:* Jang et al. 2002).

Similar features as these are often used in image-processing applications (Gonzalez, Woods, and Eddins 2009). The final two features we use are optical parameters.

Attenuation Coefficient, μ_t — This feature measures the rate at which the signal intensity drops off within the tissue. Calcified plaque has lower attenuation, and as a result, IVOCT can see deeper into these tissues compared to lipid, where IVOCT does not see as deeply. For this reason, the attenuation coefficient (or penetration depth) gives useful information about plaque types.

Incident Intensity, I_0 — This represents the back scattering characteristics of the plaque at the point where the light touches it. This feature is excellent at distinguishing fibrous plaques, which are very reflective.

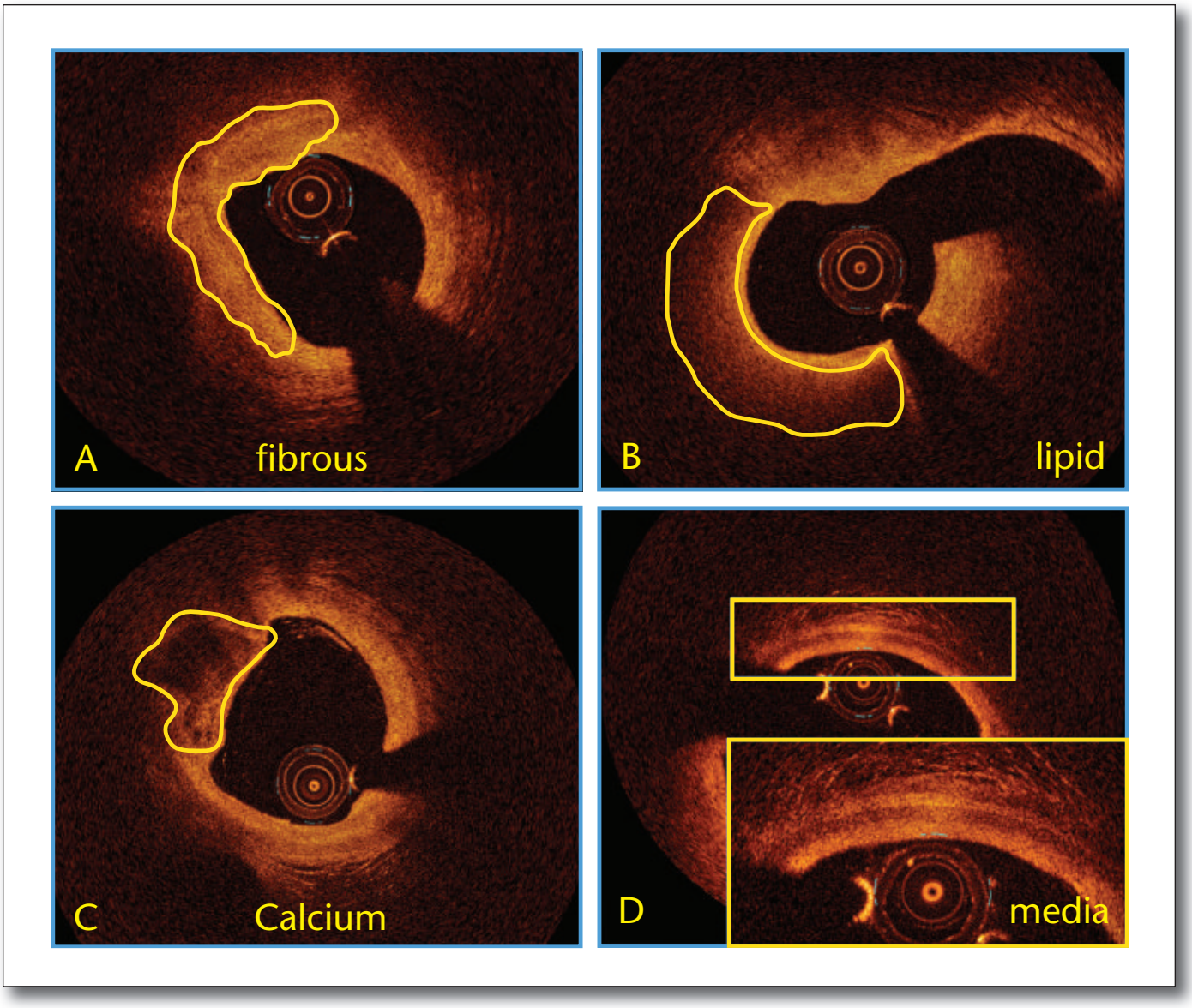


Figure 4. Appearance of Plaque Types in Clinical Images.

A is fibrous, B is lipid, and C is calcium. D shows the appearance of a normal blood vessel wall, which has layered structure.

These features are based on models of light transmission and reflectance. We verified our models by fabricating phantom (realistic imitations) blood vessels with known plaque types and checking the estimates against measured values in these cases.

The Plaque-Type Classifier

After extracting features from pixels in our IVOCT images, we then train a support vector machine (SVM) (Cristianini and Shawe-Taylor 2000) for classification of the individual pixels. The SVM is a state-of-the-art classification method. It is theoretically well founded and robust to noise in the data, which is a desirable property.

A second desirable property of the SVM is its ability to construct nonlinear classifiers through the use of kernel functions. A kernel function implicitly maps the input data to a possibly high dimensional space, where it learns a linear classifier. Since this mapping is done implicitly (that is, we never actually construct the high-dimensional feature vector), the procedure is computationally efficient. In our work, we use a radial basis function (RBF) kernel, which is a commonly used kernel.

The SVM is a binary classifier. Given that we are interested in classifying three different plaque types, we use a one-versus-rest (OVR) approach for multiclass classification. This produces three binary classifiers, one treating each class as positive and the oth-

ers as negative. During classification, each new example is classified by all three classifiers. If more than one classifies the point as positive, it is associated with the label corresponding to the classifier with the maximum margin.

There are two parameters that must be input to the SVM: C, the regularization parameter that trades off margin size and training error, and γ, the RBF kernel's bandwidth. In our experiments, we select these parameters using an internal fivefold stratified cross-validation loop and a two-dimensional grid search.

Image Data Sets

The clinical images (in vivo) that we used for evaluating our approach were selected from a large database of manually analyzed IVOCT images obtained in a clinical setting. Images were collected on the C7-XR system from St. Jude Medical Inc., Westford, MA. It has an OCT swept source that has a 1310 nm center wavelength, 110 nm wavelength range, 50 kHz sweep rate, and ~12 mm coherence length. The pullback speed was 20 mm/s and the pullback length was 54 mm. The images consist of 35 IVOCT pullbacks of the left anterior descending (LAD) and the left circumflex (LCX) coronary arteries of patients acquired prior to stent implantation, with a total of 287 images across 35 patients. An expert cardiologist on our team then labeled volumes of interest (VOIs) as belonging to one of the three plaque types in the images. The expert marked the VOIs of a particular plaque type using freehand brush strokes. On the clinical images the expert annotated 311 VOIs (roughly equal number from each plaque type). VOIs were of various sizes and shapes. Most consisted of 2–5 image frames, 50–200 A-lines, and 20–50 sample points in each A-line.

A concern with the images is that the image annotations we train with are provided by an expert and so could contain errors. To evaluate the performance of the trained classifier on ground truth, we created a second data set using cryo-imaging from cadaver samples (Salvado et al. 2006). The system serially sections and acquires micron-scale color images using different lighting wavelengths (figure 7, depicted later in this article, left column, bottom row shows an example of lipid plaque obtained this way) and autofluorescence microscopy images along the vessel (figure 7 left column, top row shows a calcified lesion obtained this way). Visualization software is then used on the cryo-images to generate microscopic resolution color/fluorescence volume renderings of vessels, in which plaque architecture and components are fully preserved (Nguyen et al. 2008, Prabhu et al. 2016). This provides an accurate depiction of the vessel without the limitations of standard histological fixation and processing (shrinkage, spatial distortion, missing calcifications, missing lipid pools, tears, and so on). Most importantly, this provides 3D validation for volumetric IVOCT pullback. Furthermore, in cases where plaque type may be ambiguous, the system enables acquisition of standard cryo-histology.

We acquired a set of 106 such cryo-images. Note that, since these are ex vivo, we do not use these images for training our classifiers but use them to validate the results. We call these images "cryo-images" below to distinguish them from the previous set.

Empirical Evaluation

We now describe experiments to test our hypothesis that the system we described will be able to accurately and efficiently classify different plaque types from IVOCT images.

We preprocess all images for speckle noise reduction, baseline subtraction, catheter optical system correction, and catheter eccentricity correction. We segment the lumen and the back border using dynamic programming. To do this, we use a cost function from prior work (Wang 2012). An example of the results of the back-border segmentation is shown in figure 5 in both the r-θ view and the x-y view. Segmenting the image in this way is important because (1) the regions of interest are contained between these borders and the rest of the pixels do not contain any relevant information, and (2) it enables us to properly compute the distance to the lumen and the beam penetration depth discussed previously, which are important signals for different plaque types.

Next, we generate features by scanning the annotated VOIs in the image pixel by pixel. For each pixel, we construct a 7 x 11 x 3 neighborhood (0.035mm x 0.055mm x 0.6mm) around it. As long as the neighborhood is within the VOI, the features of the box are computed as explained above and the values are assigned to the pixel. In the cryo-images we generated features for all pixels between border regions in a similar way.

For cross validation we use the processed images with a leave-one-pullback-out strategy. Here, in each iteration, we hold out all the data from one pullback as the test set and use the remaining 34 pullbacks as the training set. This mimics practical usage where the system will operate on novel pullbacks and is more stringent than using random folds. In a second experiment, we ran the trained classifiers on the cryo-images (these were not used at all during training/cross validation). We ran our experiments on a 64-bit Windows 7 machine with third-generation Intel Core i7 and 16 GB RAM.

Results and Discussion

The receiver operating curves (ROC) for each OVR classifier from the cross-validation experiment is shown in figure 6. The summary statistics are shown in table 1, where the accuracy, sensitivity, and specificity are noted at the optimal operating point along

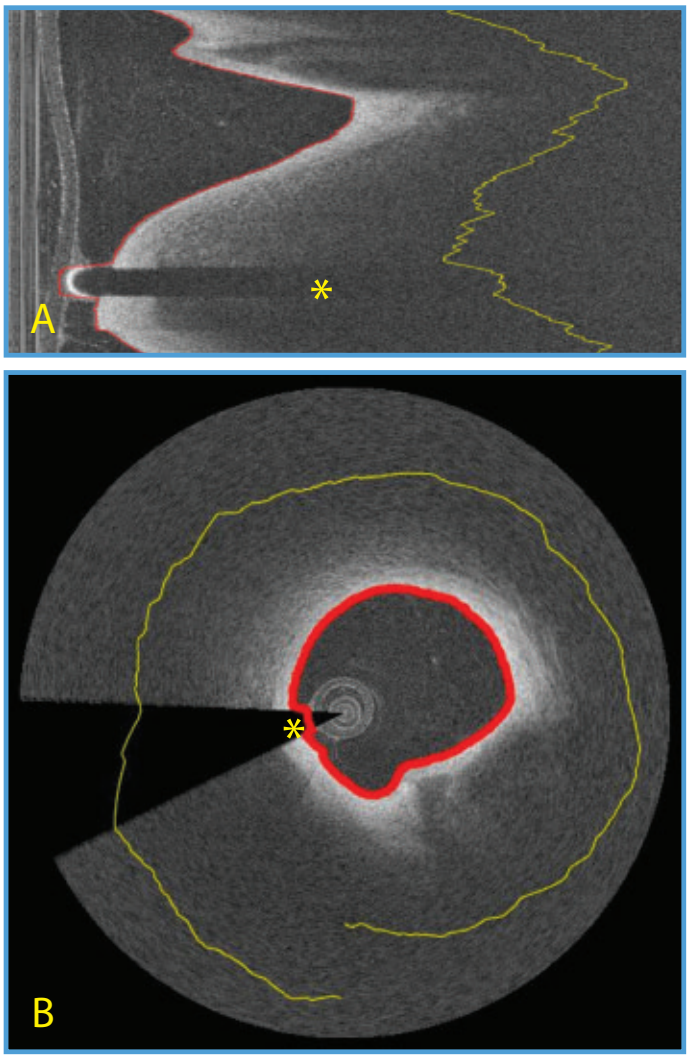


Figure 5. Results of the Back-Border Segmentation.

An illustration of back-border segmentation (yellow line) along with lumen segmentation (red line) in a typical clinical image in both views. (a) is the polar image and (b) is the *x-y* image. The yellow line is broken due to view conversion. Asterisk marks the guide-wire shadowing artifact.

	Calcium	Lipid	Fibrous
Accuracy	92.2 ± 6.28%	96.95 ± 2.79%	96.17 ± 4.0%
Sensitivity	93.0 ± 2.58%	98.95 ± 2.35%	94.28 ± 5.23%
Specificity	96.5 ± 3.39%	93.65 ± 2.77%	95.89 ± 2.18%
AUC	0.9837	0.9947	0.9959

Table 1. Performance Measures.

Area under ROC and the accuracy, sensitivity, and specificity at the optimal operating point on the ROC curves.

the curves. The ROC describes the system's behavior for a range of confidence threshold settings and enables the cardiologist (the end user) to decide on weighting the false positives (FPs) and false negatives (FNs) unequally (a very desirable property according to our expert).

The overall accuracy results, averaged over 35 folds, are shown in table 2. As can be seen from all of these results, our approach has excellent accuracy for all three plaque types. In fact, across the 35 folds, the median accuracy for all three plaque types is 100 percent, indicating that our classifiers are able to perfectly separate the plaque types using the features we designed. In a few folds, the accuracy is lower than 100 percent. We conjecture that this is because some pullbacks have many more images associated with them than others. When such a pullback is held out, the training set size decreases in size and yields a classifier with lower accuracy.

In the second experiment, we ran our trained classifier on the cryo-images. We also ran a baseline approach following Ughi et al. (2013). This approach uses beam-attenuation estimates from a layer model applied to single A-lines and 2D texture and geometric measures as features for classification with the added requirement of manual region of interest selection for analysis. These results are shown in table 3. Here the "Other" row corresponds to pixels in these images that belong to none of the three plaque types. The accuracy of the approach in this case is lower, possibly because these are ex vivo images, which have somewhat different characteristics from the training set. However, our approach still outperforms the state of the art. Further, these values are still at a very useful level according to our expert. In particular, cardiologists now divide an image into quadrants and simply state whether a quadrant contains a certain plaque type. If we use this as a performance measure, our current approach has perfect accuracy on the cryo-images.

The results also indicate that in some cases some plaque types may be confused with others. For example, the average intensity of a lipid region may be very close to that of calcium. However, they may still be separable due to the fact that the lipid's attenuation coefficient is much higher.

To confirm our intuitive understanding of the plaques' characteristics we performed a leave-one-feature-out experiment. In this experiment, we ran the classifier using all of the features and noted the accuracy measures (as shown in table 2). We then removed each feature at a time to see the impact on the accuracy. We found that removing the attenuation parameter had the biggest impact on the lipid accuracy, reducing it down to 92.4 ± 8.87 percent, while removing the average intensity feature had a significant effect on the fibrous' accuracy and uncertainty (down to 95.2 percent ± 10.75).

In addition to high accuracy, our approach is also efficient at classification. Each test fold (on average

200,000 data points) was classified in 0.366 seconds by our implementation. This facilitates future real-time usage.

Finally, we consider whether an automatic classification procedure such as this can be useful in reducing the amount of time taken to process images in a clinical setting. In an initial experiment, we found that cardiologists would spend approximately five hours analyzing a section of a blood vessel. We then created a tool (figure 8) with our classifier built in. The screen of this graphic user interface (GUI) is divided into two main regions. The leftmost region contains the tools provided to the user. There, the user can select which view is most informative, adjust image contrast and/or window level, and so on. The right region is the work area where the user can interact with any of the views, slide along the pullback to focus on the cross section of interest, make measurements, create annotations, and more. The cardiologist would run the classifier for a new image and then, using the interactive tools, analyze the results and correct some of the errors in the predictions.

The process, which the cardiologist follows, can be described by following the process used in order to annotate, classify, validate, and clean classification results as shown in figure 7. In this figure, the leftmost column shows cryo-images (Roy et al. 2009) while the second column from the left shows the IVOCT images. Using the annotation function of the plaque analysis tool, the expert would annotate the image pixels as belonging to either calcium, lipid, fibrous, or something else (used during training). The third column from the left shows the result of this annotation. It shows a mask, the same size as the image itself, that indicates the location of each plaque using colors. The next step includes running the classifier, the results of which are shown in the fourth column from the left. These results after preprocessing to remove isolated artifact predictions are presented to the cardiologist (rightmost column).

We found that this process took at most an hour, a reduction of 80 percent. This effort reduction indicates that improving the tool (figure 8) will make it deployable in the near future.

Conclusion and Future Work

In this article, we have discussed an important emerging application: an automated approach for early plaque detection in blood vessels. Our approach analyzes IVOCT images to solve this task. Using a carefully designed feature set, we show that an SVM with an RBF kernel is a high-accuracy classifier for this task. Our results are of significant impact on this important problem (Wagstaff 2012) with implications for early diagnosis of cardiovascular disease. Now, for the first time, to our knowledge, it is possible to perform complete plaque analysis automatically, enabling not only treatment planning for plaque modification in real time but also to provide enough information to perform studies on the effects of various treatments of vulnerable plaque as well as offline assessment of drug and biologic therapeutics.

In future work we will develop a complete software suite for automated plaque characterization, creating

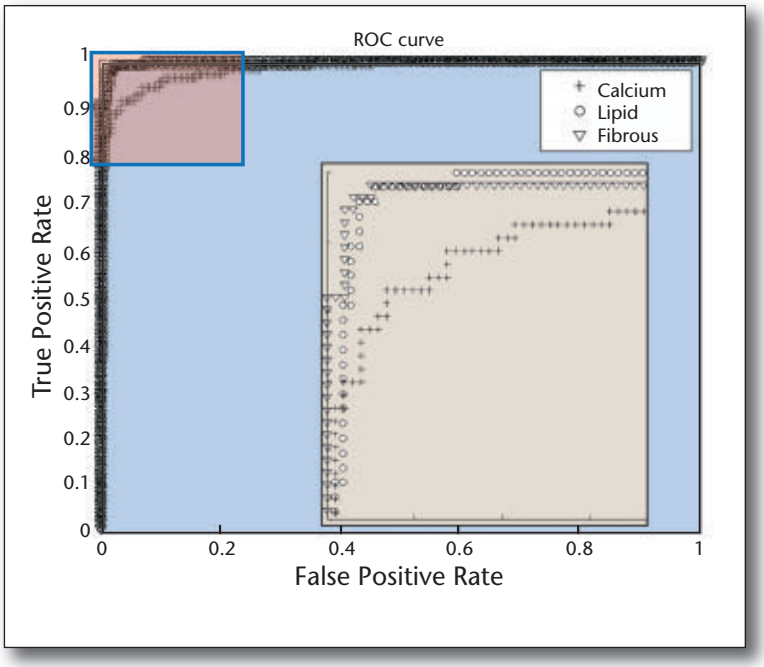


Figure 6. ROC Curve for All Three Plaque Types.

Area under the curve (AUC) values are 0.9837, 0.9947, and 0.9959 for calcium, lipid, and fibrous, respectively.

	Accuracy	Median Acc.
Overall	90.70 ± 8.28%	
Calcium	92.14 ± 10.74%	100%
Lipid	96.40 ± 8.87%	100%
Fibrous	100% ± 0.0%	100%

Table 2. Accuracy Results for Leave-One-Pullback-Out Experiment.

	Our Approach	Baseline
Overall	81.15%	69.4%
Calcium	97.62%	66.88%
Lipid	87.65%	67.07%
Fibrous	97.39%	77.95%
Other	77.96%	30.46%

Table 3. Accuracy Results for Cryo-images.

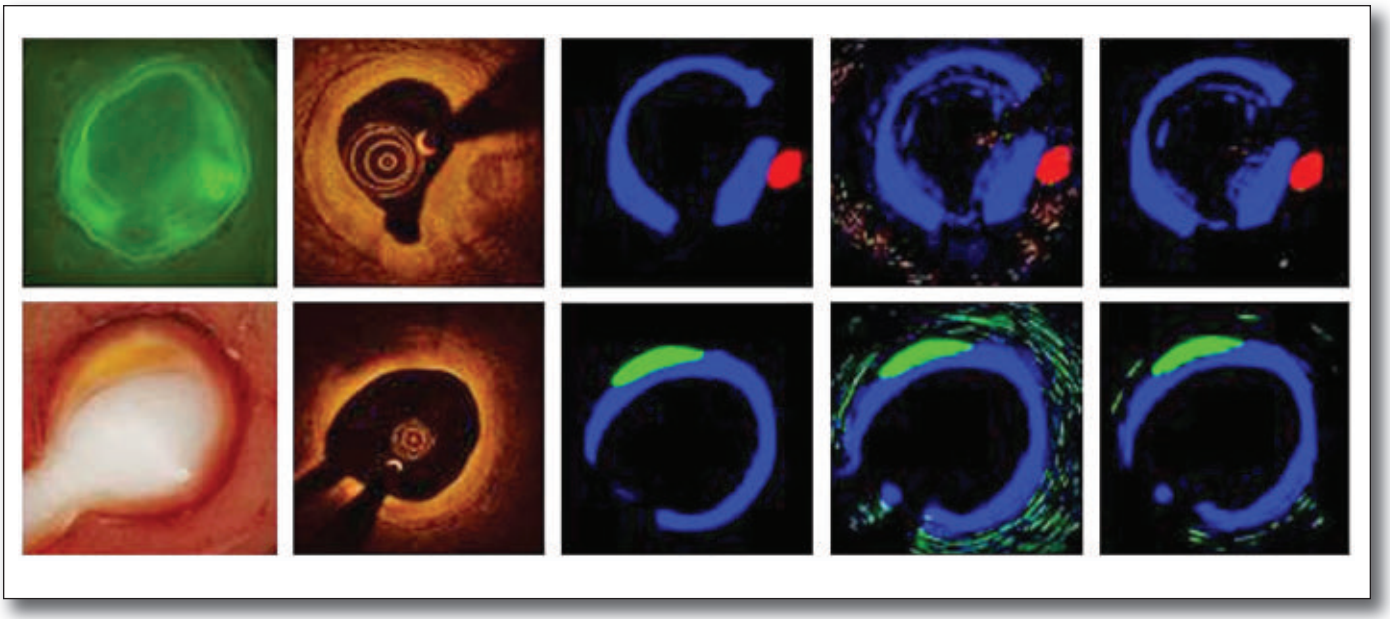


Figure 7. Example of Validation Analysis.

Top row images (from left to right): cryo-image fluorescence, IVOCT, expert annotation of IVOCT guided by registered cryo-image, results of automated classification, and automated classification after noise cleaning. In the bottom row, image data from a different vessel segment are shown with the exception that the fluorescence image is replaced by the color cryo-image. Calcium, fibrous, and lipid are labeled red, blue, and green respectively. Note the good correspondence between the third and fifth columns, indicating good classifications.

a powerful tool for live-time treatment planning of coronary artery interventions by adding functionality such as integration with a real-time 3D visualization module that will be able to quantify (volume, area covered, and others) the presence of calcified regions. An example of such visualization is shown in figure 9, which is implemented by stacking the output of multiple 2D images.

This can help in decision making regarding stent implantation and preimplantation treatment, or plaque remodeling (for example, directional atherectomy). We also plan to add an explanatory module to help explain the automated classification process to the interventional cardiologists and to accept feedback in an active learning environment. Finally, we will develop an easily accessible web-based tool for offline analysis of IVOCT images.

We expect that such a tool will be used by entities requiring fast analysis that can provide data useful for drug assessment, experimental therapeutics, and experimental medical devices.

Acknowledgments

This project was supported by Ohio Third Frontier, and by the National Heart, Lung, and Blood Institute through grants NIH R21HL108263 and 1R01HL114406-01, and by the National Center for Research Resources and the National Center for Advancing Translational Sciences through grant UL1RR024989. These grants are collaboration between Case Western Reserve University and University Hospitals of Cleveland.

References

Chamié, D.; Bezerra H. G.; Attizzani. G. F.; Yamamoto, H.; Kanaya, T.; Stefano, G. T.; Fujino, Y.; Mehanna, E; Wang, W.; Abdul-Aziz, A.; Dias, M.; Simon, D. I.; Costa, M. A.; 2013. Incidence, Predictors, Morphological Characteristics, and Clinical Outcomes of Stent Edge Dissections Detected by Optical Coherence Tomography. *JACC Cardiovascular Interventions* 6(8): 800–813. doi.org/10.1016/j.jcin.2013.03.019

Costa, M. A.; Angiolillo. D. J.; Tannenbaum, M.; Driesman, M.; Chu, A.; Patterson, J.; Kuehl, W.; Battaglia, J.; Dabbons, S.; Shamoon, F.; Flieshman, B.; Niederman, A.; Bass, T. A. 2008. Impact of Stent Deployment Procedural Factors on Long-Term Effectiveness and Safety of Sirolimus-Eluting Stents (Final Results of the Multicenter Prospective STLLR Trial). *American Journal of Cardiology* 10(12): 1704–1711. dx.doi.org/10.1016/j.amjcard.2008.02.053

Cristianini, N., and Shawe-Taylor, J. 2000. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, UK: Cambridge University Press.

Falk, E. 1983. Plaque Rupture with Severe Pre-Existing Stenosis Precipitating Coronary-Thrombosis — Characteristics of Coronary Atherosclerotic Plaques Underlying Fatal Occlusive Thrombi. *British Heart Journal* 50(2): 127–134.

Fujimoto, H.; Nakamura, M.; and Yokoi, H. 2012. Impact of Calcification on the Long-Term Outcomes of Sirolimus-Eluting Stent Implantation: Subanalysis of the Cypher Post-Marketing Surveillance Registry. *Circulation Journal* 76(1): 57–64.

Gargesha, M.; Shalev, R.; Prabhu, D.; Tanaka, K.; Rollins, A. M.; Costa, M.; Bezerra, H. G.; Wilson, D. L. 2015. Parameter Estimation of Atherosclerotic Tissue Optical Properties from 3D Intravascular OCT. *SPIE Journal of Medical Imaging 2(1)*: 14.

Gonzalez, R. C.; Woods, R. E.; and Eddins, S. L. 2009. *Digital Image Processing Using MATLAB*. Knoxville, TN: Gatesmark Publishing.

Jang, I. K.; Bouma, B. E.; Kang, D. H.; Park, S. J.; Park, S. W.; Seung, K. B.; Choi, K. B.; Shishkov, M.; Schlendorf, K.; Pomerantsev, P.; Houser, S. L.; Aretz, H. T.; Tearney, G. J.; 2002. Visualization of Coronary Atherosclerotic Plaques in Patients Using Optical

Figure 8. Graphic User Interface of Plaque Analysis Tool.

The left side provides various editing tools. On the right the tool provides views to be used by the interventional cardiologist.

Coherence Tomography: Comparison with Intravascular Ultrasound. *Journal of the American College of Cardiology* 39(4): 604–609.

Lu, H.; Gargesha, M.; Wang, Z.; Chamie, D.; Attizzani, G. F.; Kanaya, T.; Ray, S.; Costa, M. A.; Rollins, A. M.; Bezerra, H. G.; and Wilson, D. L. 2012. Automatic Stent Detection in Intravascular OCT Images Using Bagged Decision Trees. *Biomedical Optics Express* 3(11): 2809–2824.

Mendis, S.; Puska, P.; and Norrving, B. 2011. *Global Atlas on Cardiovascular Disease Prevention and Control*. Geneva, Switzerland: World Health Organization.

Mintz, G. S.; Popma, J. J.; Pichard, A. D.; Kent, K. M.; Satler, L. F.; Chuang, Y. C.; Ditrano, C. J.; Leon, M. B. 1995. Patterns of Calcification in Coronary Artery Disease. A Statistical Analysis of Intravascular Ultrasound and Coronary Angiography in 1155 Lesions. *Circulation* 91(7): 1959–1965.

Nguyen, M. S.; Salvado, O.; Roy, D.; Steyer, G.; Stone, M. E.; Hoffman, R. D.; WIlson, D. L. 2008. Ex Vivo Characterization of Human Atherosclerotic Iliac Plaque Components Using Cryo-Imaging. *Journal of Microscopy* 232(3): 432–441.

Nishida, K.; Kimura, T.; Kawai, K.; Miyano, I.; Nakaoka, Y.; Yamanoto, S.; Kaname, N.; Seki, S.; Kubokawa, S.; Fukatani, M.; Hamashige, N.; Morimoto, T.; Mitsudo, K. 2013. Comparison of Outcomes Using the Sirolimus-Eluting Stent in Calcified Versus Non-Calcified Native Coronary Lesions in Patients On- Versus Not On-Chronic Hemodialysis (From the J-Cypher Registry) *American Journal of Cardiology* 112(5): 647–655.

Prabhu, D.; Mehanna, E.; Gargesha, M.; Brandt, E.; Wen, D.; van Ditzhuijzen, N. S.; Chamie, D.; Yamanoto, H.; Fujino, Y.; Alian,

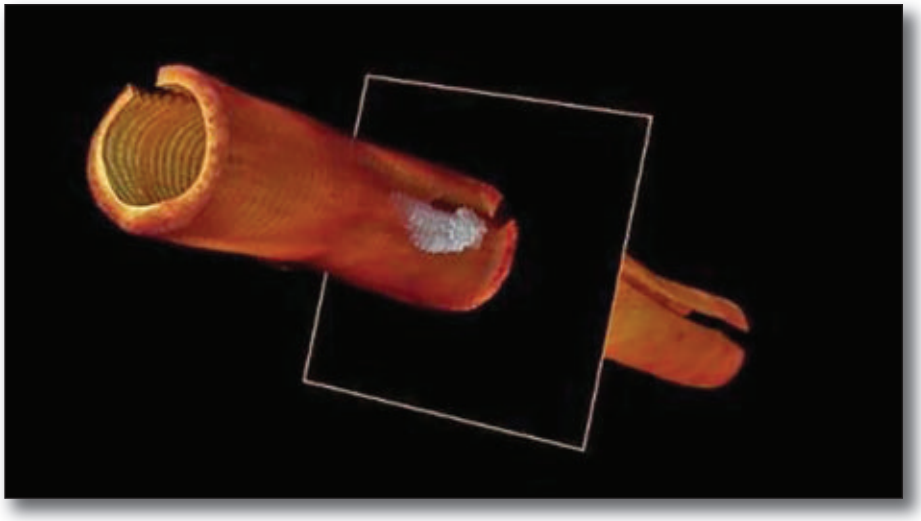


Figure 9. 3D Visualization of Calcified Region Within, Otherwise, Healthy Blood Vessel.

The white region indicates the presence of calcium (created using Amira). *Best viewed in color.*

A.; Patel, J.; Costa, M.; Bezerra, H. G.; Wilson, D. L. 2016. Three-Dimensional Registration of Intravascular Optical Coherence Tomography and Cryo-Image Volumes for Microscopic-Resolution Validation. *Journal of Medical Imaging* (Bellingham) 3(2): 026004.

Roy, D.; Steyer, G. J.; Gargesha, M.; Stone, M. E.; Wilson, D. L. 2009. 3D Cryo-Imaging: A Very High-Resolution View of the Whole Mouse. Anatomical Record: Advances in Integrative Anatomy and Evolutionary Biology. *The Anatomical Record*. 292(3): 342–351. doi.org/10.1002/ar.20849

Salvado, O.; Roy, D.; Heinzel, M.; McKinley, E.; Wilson, D. 2006. 3D Cryo-Section/Imaging of Blood Vessel Lesions for Validation of MRI Data. *Proceedings of SPIE International Society of Optical Engineering* 6142(614214): 377–386. doi.org/10.1117/12.649093

Tearney, G. J.; Jang, I. K.; and Bouma, B. E. 2006. Optical Coherence Tomography for Imaging the Vulnerable Plaque. *Journal of Biomedical Optics* 11(2): 021002-021002.

Tearney G. J.; Regar, E.; Akasaka, T.; Adriaenssens, T.; Barlis, P.; Bezerra, H. G.; Bouma, B.; Bruining, N.; Cho, J. M.; Chowdhary, S.; Costa, M. A.; de Silva, R.; Dijkstra J.; Di Mario, C.; Dudek, D.; Falk, E.; Feldman, M. D.; Fitzgerald, P.; Garcia-Garcia, H. M.; Gonzalo, N.; Granada, J. F.; Guagliumi, G.; Holm, N. R.; Honda, Y.; Ikeno, F.; Kawasaki, M.; Kochman, J.; Koltowski, L.; Kubo, T.; Kume, T.; Kyono, H.; Lam, C. C.; Lamouche, G.; Lee, D. P.; Leon, M. B.; Maehara, A.; Manfrini, O.; Mintz, G. S.; Mizuno, K.; Morel, M. A.; Nadkarni, S.; Okura, H.; Otake, H.; Pietrasik, A.; Prati, F.; Räber, L.; Radu, M. D.; Rieber, J.; Riga, M.; Rollins, A.; Rosenberg, M.; Sirbu, V.; Serruys, P. W.; Shimada, K.; Shinke, T.; Shite, J.; Siegel, E.; Sonoda, S.; Suter, M.; Takarada, S.; Tanaka, A.; Terashima, M.; Thim, T.; Uemura, S.; Ughi, G. J.; van Beusekom, H. M.; van der Steen, A. F.; van Es, G. A.; van Soest, G.; Virmani, R.; Waxman, S.; Weissman, N. J.; Weisz, G; International Working Group for Intravascular Optical Coherence Tomography. 2012. Consensus Standards for Acquisition, Measurement, and Reporting of Intravascular Optical Coherence Tomography Studies. *Journal of the American College of Cardiology* 59(12): 1058–1072.

Ughi, G. J.; Adriaenssens, T.; Sinnaeve, P.; Desmet, W.; D'hooge, J. 2013. Automated Tissue Characterization of in Vivo Atherosclerotic Plaques by Intravascular Optical Coherence Tomography Images. *Biomedical Optics Express* 4(7): 1014–1030.

Van Soest, G.; Regar, E.; Koljenovi, S.; van Leender, J. H.; Gonzalo, N.; van Noorden, S.; Okamura, T.; Bouma, B. E.; Tearney, G. J.; Oosterhuis, J. W.; Serruys, P. W.; van der Steen, A. F. W. 2010. Atherosclerotic Tissue Characterization in Vivo by Optical Coherence Tomography Attenuation Imaging. *Journal of Biomedical Optics* 15(1): 011105-011105.

Virmani, R.; Kolodgie, F. D.; Burke, A. P.; Farb, A.; Schwartz, S. M. 2000. Lessons from Sudden Coronary Death: A Comprehensive Morphological Classification Scheme for Atherosclerotic Lesions. *Arteriosclerosis, Thrombosis, and Vascular Biology* 20(5): 1262–1275.

Wagstaff, K. 2012. Machine Learning That Matters. Unpublished Manuscript. arXiv preprint arXiv: 1206.4656. Ithaca, NY: Cornell University Library.

Wang, L., and Wu, H. I. 2007. *Biomedical Optics: Principles and Imaging*. Hoboken, N.J.: Wiley-Interscience.

Wang, Z. 2012. Volumetric Quantification of Fibrous Caps Using Intravascular Optical Coherence Tomography. *Biomedical Optics Express* 3(6): 1413.

Yabushita, H.; Bouma, B. E.; Houser, S. L.; Aretz, H. T.; Jang, I. K.; Schlendorf, K. H.; Shishkov, M.; Kang, D. H.; Halpern, E. F.; Tearney, G. J. 2002. Characterization of Human Atherosclerosis by Optical Coherence Tomography. *Circulation* 106(13): 1640–1645.

Zhang, X. M.; McKay, C. R.; and Sonka, M. 1998. Tissue Characterization In Intravascular Ultrasound Images. *IEEE Transactions on Medical Imaging* 17(6): 889–899.

Ronny Shalev, Ph.D. has spent much of the past 21 years in executive positions including vice president of marketing and sales at Orbotech (NASDAQ: ORBK) where he managed sales and marketing teams, and director of worldwide program management at Marvell Semiconductor (NASDAQ: MRVL) where he managed the development of more than two hundred products. In the beginning of his career, he worked as a software developer at IBM. He founded a startup company and acquired a significant amount of experience as an entrepreneur. Shalev attended Case Western Reserve University (Cleveland, OH) and holds a M.Sc. in electrical engineering and applied physics specializing in robotics and control, and a B.Sc. in electrical engineering and applied physics. He recently returned to CWRU and earned his Ph.D. specializing in medical imaging, machine learning, and visualization.

Andrew M. Rollins, Ph.D., is a professor of biomedical engineering and medicine at Case Western Reserve University. His research interests are the development and application of advanced biomedical optical technologies, especially optical coherence tomography (OCT), and including optical stimulation and imaging of electrophysiology. Current projects include the study of developmental cardiology and endoscopic imaging of cardiovascular disease and cancer.

David L. Wilson, Ph.D., is the Robert Herbold Professor of Biomedical Engineering and Radiology, Case Western Reserve University. His research has focused on biomedical image analysis in clinical and preclinical models as well as quantitative evaluation of image quality. In the last several years, he has focused on intravascular OCT imaging and cardiovascular applications. Wilson has been principal investigator on numerous federal and state grant awards, and has more than 130 peer-reviewed publications. He serves on editorial boards, NIH study sections, and conference organization committees. At CWRU, he has served as principal investigator on major development grants, the doctoral student NIH T32 Imaging training grant, and advisor to numerous graduate and undergraduate students. Wilson is a founder of BioInVision, Inc., which is commercializing cryo-imaging. Wilson received his Ph.D. in electrical engineering from Rice University, Houston, TX.

Soumya Ray is an associate professor in the Department of Electrical Engineering at Case Western Reserve University. He is interested in machine-learning theory and applications, reinforcement learning and planning, and natural language processing. He received his undergraduate degree from the Indian Institute of Technology, Kharagpur, followed by MS and Ph.D. in computer science from the University of Wisconsin, Madison.

Hiram Bezerra is an assistant professor of medicine at Case Western Reserve University School of Medicine, the director of the Cath Lab, and the medical director of the Cardiovascular Imaging Core Lab at University Hospitals – Case Medical Center. He received his MD degree from the School of Sciences of the Santa Casa de Misericordia, and his Ph.D. from the University of São Paulo, Brazil.

Daisuke Nakamura, MD, is affiliated with the Harrington Heart and Vascular Institute, University Hospitals Case Medical Center, Case Western Reserve University in Cleveland, OH. He has stated that he has no financial interests related to this article.

Setsu Nishino, MD, Ph.D., is affiliated with the Harrington Heart and Vascular Institute, University Hospitals Case Medical Center, Case Western Reserve University in Cleveland, OH. He has stated that he has no financial interests related to this article.

Using Global Constraints to Automate Regression Testing

Arnaud Gotlieb, Dusica Marijan

■ *Communicating or autonomous systems rely on high-quality software-based components. that must be thoroughly verified before they are released and deployed in operational settings. Regression testing is a crucial verification process that compares any new release of a software-based component against its previous versions, by executing available test cases. However, limited testing time makes selection of test cases in regression testing challenging, and some selection criteria must be respected. Validation engineers usually address this problem, coined as test suite reduction (TSR), through manual analysis or by using approximation techniques. In this paper, we address the TSR problem with sound artificial intelligence techniques such as constraint programming (CP) and global constraints. By using distinct cost-value-aggregating criteria, we propose several constraint-optimization models to find a subset of test cases that cover all the test requirements and optimize the overall cost of selected test cases. Our contribution includes reuse of existing preprocessing rules to simplify the problem before solving it and the design of structure-aware heuristics that take into account the notion of the costs associated with test cases. The work presented in this paper has been motivated by an industrial application in the communication domain. Our overall goal is to develop a constraint-based approach of test suite reduction that can be deployed to test a complete product line of conferencing systems in continuous delivery mode. By implementing this approach in a software prototype tool and experimentally evaluating it on both randomly generated and industrial instances, we hope to foster a quick adoption of the technology.*

More and more, communicating or autonomous systems rely on software-based components that have high standards in terms of reliability, robustness, and quality. For instance, professional conferencing systems or industrial robotics systems are released in markets where high quality is considered a competitive advantage. For these systems, an increased effort in software validation and verification is required to produce high-quality components that can be deployed in operational settings.

Software validation and verification include several distinct phases such as functional testing, performance testing, and regression testing. Regression testing verifies that a new release of a software component still performs as expected after new features are implemented. By executing the software component with existing test cases that were used to test previous releases, regression testing checks for the absence of regression faults, that is, faults that may have been reintroduced into the application during development of new features. In order to keep the time to market of new releases short, a judicious selection of test scripts to execute has to be performed.

Dealing with multiple criteria when performing regression testing is important. For example, selecting a test suite that minimizes total execution time while preserving its coverage of user requirements is highly desirable for testing of software components. Yet the budget allocated to testing is limited, and optimizing the selection of test cases is a time-consuming activity. In practice, validation engineers solve the test suite reduction (TSR) problem through manual analysis or by approximation techniques. However, automated means to solve TSR instances efficiently are required when software components are developed in continuous delivery mode (Stolberg 2009). In fact, continuous integration involves frequent execution of regression test scripts to detect faults as early as possible, which means that automated selection of regression tests is indispensable in this context.

Overview

Formally speaking, given a set of requirements and a test suite that covers these requirements, the test suite reduction problem aims at finding a smallest subset of test cases in the test suite such that any requirement is covered at least once. By considering a cost value associated with each test case, a natural extension of this problem is to minimize the overall cost of the test suite, not just its size. Unfortunately, solving TSR is intractable in general (Harrold, Gupta, and Soffa 1993), and compromises have to be found either by adopting heuristics-based approximation algorithms or by using time-aware exact approaches.

Existing Results

The topic of test suite reduction has received considerable attention in the last two decades. Roughly speaking, we can distinguish greedy techniques (Tallam and Gupta 2005, Jeffrey and Gupta 2005), search-based testing techniques (Ferrer et al. 2015; Wang, Ali, and Gotlieb 2015), and exact approaches (Hsu and Orso 2009; Chen, Zhang, and Xu 2008; Campos et al. 2012; Li et al. 2014; Gotlieb and Marijan 2014).

Greedy techniques for test suite reduction usually select first the test cases that cover the most requirements and iterate until all requirements are covered. In the 1990s, Harrold, Gupta, and Soffa (1993) proposed a technique that approximates the computation of minimum-cardinality hitting sets. This work was further refined with different variable orderings (Offutt, Pan, and Voas 1995). More recently, Tallam and Gupta (2005) introduced the delayed-greedy technique, which exploits implications among test cases and requirements to refine further the reduced test suite. The technique starts by removing test cases that cover the requirements already covered by other test cases. Then it removes test requirements that are not in the minimized requirements set, and finally it determines a minimized test suite from the remaining test cases by using a greedy approach. Jeffrey and Gupta (2005) extended this approach by retaining test cases that improve a fault-detection capability of the test suite. Comparing to the paper by Harrold, Gupta, and Soffa (1993), the approach produces bigger solutions, but with higher fault-detection effectiveness.

One shortcoming of greedy techniques is that they only approximate global optima without providing any guarantee of optimality. Search-based testing techniques have also been tailored for test suite reduction. Wang, Ali, and Gotlieb (2015) explore classical metaheuristics such as hill climbing, simulated annealing, or weight-based genetic algorithms for (multiobjective) test suite reduction. By comparing 10 distinct algorithms for different criteria, they observed that random-weighted multiobjective optimization is the most efficient approach. However, by assigning weights at random, this approach is unfortunately not able to place priority over the various objectives. Ferrer et al. (2015) examine other algorithms based on metaheuristics. All these techniques can scale up to problems that have a large number of test cases and requirements, but they cannot explore the overall search space and thus they cannot guarantee global optimality.

On the contrary, exact approaches, which are based either on Boolean satisfiability or integer linear programming (ILP), can reach true global minima. The best-known approach for exact test suite minimization is implemented in MINTS (Hsu and Orso 2009). MINTS has been used to perform test suite reduction for various criteria including energy consumption on mobile devices (Li et al. 2014). Similar exact techniques have also been designed to handle fault localization (Campos et al. 2012). Generally speaking, the theoretical limitation of exact approaches is the possible early combinatorial explosion to determine the global optimum, which exposes these techniques to serious limitations even for small problems. A hybrid method based on ILP and search, called DILP, is proposed by Chen, Zhang, and Xu (2008), where a lower bound for the minimum is computed and a search for finding a smaller test suite close to this bound is performed. Recently, another ILP-based approach is proposed by Hao and colleagues (Hao et al. 2012) to set up upper limits on the loss of fault-detection capability in the test suite. Mouthuy, Deville, and Dooms (2007) proposed a constraint called SC for the set-covering problem. They created a propagator for SC by using a lower bound based on an ILP relaxation. Finally, Gotlieb and Marijan (2014) introduced an approach for test suite reduction based on the computation of maximum flows in a network flow. This initial idea has partly triggered the work reported in the present article.

Contributions

This article proposes a new approach of test suite reduction based on constraint programming (CP) and global constraints. Global constraints encode relations over a nonfixed number of variables with dedicated and efficient filtering algorithms. Our approach uses three special global constraints developed in CP, namely NVALUE, GLOBALCARDINALITY, and SCALAR_PRODUCT. NVALUE constrains the number of distinct values that can be taken by a set of variables (Pachet and Roy 1999), while GLOBALCARDINALITY generalizes this relation by considering explicit cardinality values for these variables (Régin 1996). SCALAR_PRODUCT simply encodes the scalar product between two vectors of variables as a relation. By combining these global constraints with advanced preprocessing rules and sophisticated structure-aware search heuristics, the proposed approach creates a constraint-optimization model that competes with the best known exact approach for test suite reduction, namely MINTS (Hsu and Orso 2009). As said above, associating a cost value to each test case is a natural extension of TSR. Indeed, such a cost value can represent or aggregate distinct notions such as execution time, code coverage, energy consumption (Li et al. 2014), or fault-detection capabilities (Campos et al. 2012). Using these cost values, TSR reduces to the problem of selecting a subset of test cases such that all the requirements are covered and the overall cost of the test suite is minimized. The proposed approach is also capable of optimizing an overall cost function depending of these cost values, while preserving the full coverage of requirements. We implemented our approach in a tool called Flower/C and performed a set of experiments with both randomly generated TSR instances and industrial instances. The experimental results show that Flower can be deployed into an industrial context and its route for exploitation is discussed. Next section formally defines TSR and gives some background on CP and global constraints. The following section shows three CP optimization models involving distinct combinations of global constraints. It also introduces preprocessing rules for TSR that can simplify the instances beforehand, and a dedicated search heuristics. The following section presents an experimental evaluation of the proposed models as well as a comparison with other approaches. Finally, the last sections draw perspectives for the industrial exploitation of the proposed approach and conclude the article.

Background

This section formalizes the test suite reduction problem and briefly reviews the notion of global constraints.

Test Suite Reduction

Test suite reduction aims to select a subset of test cases out of a test suite, which minimizes its overall cost, while retaining its coverage of requirements. Roughly speaking, a TSR instance is defined by an initial test suite T composed of m test cases $\{t_1, \ldots, t_m\}$, each test case being associated with a cost value noted $c(t_i)$, a set of n requirements $R = \{r_1, \ldots, r_n\}$, and a function called $cov(r_i)$ that maps each requirement r_i to the subset of test cases that cover it. We suppose that each requirement is covered by at least one test case and each test case covers at least one requirement. An example with five test cases and five requirements is given in table 1, where the value given in the table denotes the cost of each test case. Solving TSR aims at finding a subset of test cases such that every requirement is covered at least once, and the overall cost is minimized.

cost	r_1	r_2	r_3	r_4	r_5
t_a	2	2	-	-	-
t_b	1	-	1	-	-
t_c	-	3	3	-	3
t_d	-	-	-	2	2
t_e	-	-	-	1	-

Table 1. A TSR Problem Instance.

A labeled bipartite graph can be used to encode any TSR instance, with edges denoting the relation *cov* and labels denoting the costs over the test cases, as shown in figure 1. The overall cost of a test suite can be computed as the sum of each individual cost of its test cases, but other functions can be considered as well (for example, the max of costs). Note that the cost associated to any test case does not differ with respect to the covered requirement. The framework can be extended with a distinct cost for each requirement, but this brings more complexity without much benefit for validation engineers. Note also that the optimal solution shown in figure 1 is not unique. For example, $\{t_a, t_b, t_d\}$ covers all the requirements and has also $TotalCosts = c(t_a) + c(t_b) + c(t_d) = 5$. When the cost associated to each test case are all the same, then TSR reduces to the problem of finding a subset of minimal size.

Constraint Programming and Global Constraints

Constraint programming is a powerful declarative paradigm where logic and control are driven by constraint solving. Any constraint enforces a symbolic relation over a set of unknown variables, which take their values in a domain (Rossi, van Beek, and Walsh 2006). When the domain is finite, it can be mapped to a finite subset of integers without any loss. A con-

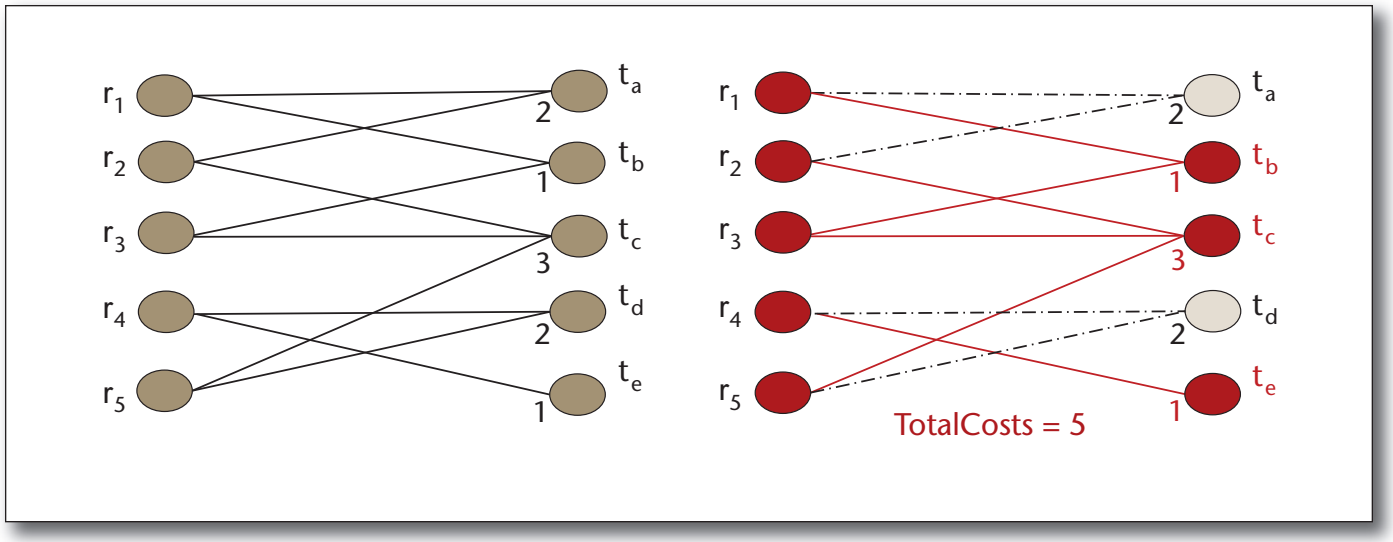


Figure 1. TSR as a Bipartite Graph with an Optimal Solution.

(a) Bipartite Graph. (b) Optimal Solution.

straint program over finite domain variables is a finite set of constraints, which come with filtering algorithms. These algorithms prune the domains of the constraint variables from some of their inconsistent values. For instance, if X takes an unknown value in the finite domain {2, 3, 5} and Y takes a value in {3, 4, 5, 6} then the filtering algorithm associated with $X = Y$ can prune the domains of both X and Y to {3, 5}. In this context an assignment is just a mapping from any variable, noted with uppercase letters in the article, to a value from its respective domain, noted with lowercase letters. A constraint program is satisfiable when there exists at least one assignment that satisfies all the constraints. It is unsatisfiable otherwise. Among the satisfiable assignments, some can minimize a cost function and thus, CP can be used to solve optimization problems as well.

In CP, two types of constraints can be distinguished, namely the relations that hold over a known number of variables (typically 1, 2, or 3) and relations that hold over a nonfixed number of variables. Constraints from this latter category are called global constraints, especially when they implement dedicated and efficient filtering algorithms.

A first example of global constraints is given by the following constraint:

Definition 1. (NVALUE [Pachet and Roy 1999]).
Let N be a domain variable and V be a vector of domain variables, NVALUE(N, V) holds iff the number of distinct values in V is equal to N.

For instance, NVALUE(N, [3, 1, 3]) entails $N = 2$ and is solved, NVALUE(3, [X_1, X_2]) is unsatisfiable, and NVALUE(1, [X_1, X_2, X_3]) entails $X_1 = X_2 = X_3$.

Another example of global constraint, which generalizes NVALUE is now given:

Definition 2. (GLOBALCARDINALITY [Régin 1996]).
Let $T = (T_1, \ldots, T_n)$ be a vector of domain variables, let $d = (d_1, \ldots, d_m)$ be a vector of distinct integers, and let $C = (C_1, \ldots, C_m)$ be a vector of domain variables, GLOBALCARDINALITY(T, d, C) holds iff for each i ∈ 1..m the number of occurrences of d_i in T is C_i. The C_i variables are the occurrence variables of the constraint.

For instance, GLOBALCARDINALITY(((T_1, T_2, 5), (5, 7), (C_1, C_2)) prunes the domains of T_1 and T_2 to {5, 7}, the domain of C_1 to {1, 2, 3} and the domain of C_2 to {0, 1, 2}. A polynomial filtering algorithm for this constraint was given by Regin (1996).

CP Models of the TSR Problem

In this section, we present distinct constraint-optimization models based on NVALUE, GLOBALCARDINALITY for the TSR problem.

A Naive Model (NVALUE)

In CP, TSR can easily be encoded with the following scheme: each requirement to be covered can be associated with a domain variable R having as finite domain, which is composed of the test cases that cover the requirement. More precisely, R belongs to {t_1, …, t_n}, where each t_i corresponds to an integer associated with a test case that covers R. So, for example, the instance reported in table 1 can be encoded as follows:

$R_1 \in \{1, 2\}$, $R_2 \in \{1, 3\}$, $R_3 \in \{2, 3\}$, $R_4 \in \{4, 5\}$, $R_5 \in \{4\}$

where t_a is associated with 1, t_b is associated with 2, and so on.

Figure 2 shows a first constraint-optimization program for an instance of test suite reduction.

This model aims to minimize the number of different values that can be taken by R_1, …, R_n, that is, the number of distinct test cases that cover all the

Minimize Ns.t.NVALUE(N, (R_1, . . . ,R_n))
for i = 1 to n s.t. $R_i \neq R_j$ for any j do $\sum_i c(t_{R_i})$ = *Total Costs.*

Figure 2. A First Constraint-Optimization Model for TSR (Naive).

Maximize N s.t.

GLOBALCARDINALITY((R_1, . . . ,R_n); (t_1, . . . , t_m); (O_1, . . . ,O_m))∧
GLOBALCARDINALITY((O_1, . . . ,O_m); (0); (N))
for i = 1 to n s.t. $R_i \neq R_j$ for any j do $\sum_i c(t_{R_i})$ = *Total Costs.*

Figure 3. A Constraint-Optimization Model for TSR Based on GLOBALCARDINALITY.

requirements. Using NVALUE enables the minimization process to reduce the number of test cases, while the second part of the model computes the sum of costs. This model is naive for two reasons: firstly, it does not guarantee finding the minimum of costs even though it finds the minimum number of test cases (issue 1), and secondly, it allows us only to search on a tree composed of the requirement variables (issue 2). In fact, the only variables of this model are the R_i, which means that branching on the selection of test cases is unfortunately not possible. For example, selecting first the test cases that cover the most requirements while searching for a minimum is not possible. In order to tame this problem (issue 2), another model based on GLOBALCARDINALITY can be proposed.

A Model with GLOBALCARDINALITY(*GCC*[2])

Let O_i be a domain variable representing the number of times test case t_i is selected to cover R_1, …, R_n. The model shown in figure 3 addresses TSR by using two GLOBALCARDINALITY constraints.

The first GLOBALCARDINALITY enforces the coverage relation between test cases and requirements by constraining the occurrence variables O_i, while the second GLOBALCARDINALITY counts the number of 0 (zeros) in the list of occurrence variables. This allows the model to constrain the selection of test cases by maximizing the number of unselected test cases. Thus, branching on the number of occurrences of each test case becomes possible with this model. Still, this model does not address issue 1 mentioned above, as it does not guarantee to reach the minimum of the overall cost of test cases. Another model can be proposed to deal with both issue 1 and issue 2.

An Optimized Model *(Mixt)*

In this third model, (R_1, …, R_n.), (O_1, …, O_m) are decision variables, only known through their domain. The Boolean variables B_1, …, B_m are local variables introduced to establish the link with costs. By using the global constraint SCALAR_PRODUCT((B_1, …, B_m), (c_1, …, c_m), *TotalCosts*), which enforces the relation

$$TotalCosts = \sum_{1 \leq i \leq m} B_i * c_i$$

this model actually minimizes *TotalCosts,* the sum of the costs of selected test cases. In fact, the nonnull B_i variables correspond to the selected test cases. The constraint GLOBALCARDINALITY allows us to constrain the variables O_i, which are associated with the number of selected test cases. This model can be solved by searching the space composed of the possible choices for (R_1, …, R_n), (O_1, …, O_m). Interestingly, it allows us to branch either on the choice of requirements or on the choice of test cases. Hence, it addresses both issues 1 and 2. An optimal solution of this model is an optimal solution of TSR and vice versa, as proved by the following sketch of proof.

(⇒) An optimal solution of TSR corresponds to the assignment of (R_1, …, R_n) with test cases that mini-

Minimize *TotalCosts* s.t.
GLOBALCARDINALITY((R$_1$, . . . ,R$_n$); (1, . . . ,m); (O_1, . . . ,O_m)),
for i = 1 to m do $B_i = (O_i > 0)$,
SCALAR_PRODUCT((B_1, . . . ,B_m); (c_1, . . . , c_m), *TotalCosts*).

Figure 4. A Constraint-Optimization Model for Solving TSR.

mize the sum of costs. Let us call $\{t_p, \ldots, t_q\}$ this solution and minimum this sum. This is also an optimal solution of our model. In fact, the variables $\{O_p, \ldots, O_q\}$ are strictly positive because their associated test case is selected in the solution through GLOBALCARDINALITY, which means that only the corresponding $\{B_p, \ldots, B_q\}$ are equal to 1 and thus SCALAR_PRODUCT((B_1, …, B_m), (c_1, …, c_m), *TotalCosts*) is equal to *minimum*.

(⇐) An optimal solution m of our constraint-optimization model is also an optimal solution of TSR. In the model, *TotalCosts* is assigned to the sum of costs of selected test cases and there exists no other assignment of B_i, which gives a smaller value than m. Then, it means that m is actually the minimum cost of the TSR instance, and the test cases selected by the B_i are the solution of this problem.

Even if the model given in figure 4 is generic, it involves searching a space of exponential size $O(D_n)$ where D denotes the size of the greatest domain of any requirement variable and n is the number of test cases. This does not come as a surprise as TSR has been shown to be NP-hard (Hsu and Orso 2009).

Solving TSR can be improved by considering a number of optimizations, including preprocessing rules and specialized search heuristics.

Preprocessing

Preprocessing can be used to reduce the size of the problem beforehand, by using the following rules:

Rule 1.
For two test cases t_1, t_2, if all the requirements covered by t_1 are included in the subset of requirements covered by t_2, then t_1 can be safely ignored during search, as it is always be preferable to select t_2 instead of t_1.

Rule 2.
Conversely, for two requirements r_1, r_2, if all test cases covering r_1 are included in the subset of test cases covering r_2, then r_2 can be safely removed from the set of requirements to be covered. Indeed, any test case covering r_1 will automatically cover r_2 as well.

Rule 3.
If there is a requirement that is covered by only a single test case t then t must be included in the solution set. Figure 5 illustrates these preprocessing rules.

A Dedicated Heuristic

Search heuristics include strategies for selecting a variable to be enumerated first and a value to be selected first. Both strategies can be tuned by the available constraints and variables of the model. The first idea is to use the classical first-fail principle, which selects first the variable representing the requirement that is covered by the least number of test cases. As all the requirements have to be covered, it means that these test cases are most likely to be selected. However, this strategy ignores the selection of the test case having the least cost or the test cases covering the most requirements. Regarding value selection, it is thus better to define a special heuristic for our problem.

Unlike the static variable selection strategy used in greedy algorithms, such as, for example, the selection of variables based on the number of covered requirements, our TSR-dedicated strategy is dynamic and the ordering is revised at each step of the selection process. It selects first the variable O_i associated with the test case with the smallest cost. Then, among the remaining test cases that cover any requirement not yet covered, it selects the variable O_j with the smallest cost and iterates until all the requirements are covered. In case of a choice that does not lead to a global minimum, the process backtracks and selects a distinct test case, not necessarily associated with the smallest cost. Regarding the value-selection strategy, each time a value selection is made, our TSR-dedicated heuristics select first the test cases that cover the most requirements. Property 1 formalizes this idea.

Property 1.
Let each test case t_i be represented by an occurrence variable O_i taking its values in $0..max_i$ where max_i is dynamically updated with the current partial assignment. Then, for each solution X of the TSR problem with cost $f(X)$ where $Oi = n_i$ such that $0 < n_i < max_i$ (strict inequalities), there is at least one other solution Y with cost $f(Y) \leq f(X)$ where either $O_i = 0$ or $O_i = max_i$.

Note that our proposed TSR-dedicated heuristic is incomplete, meaning that some parts of the search tree can remain unexplored. Indeed, symmetrical solutions can be ignored as explained in figure 6, but, refer-

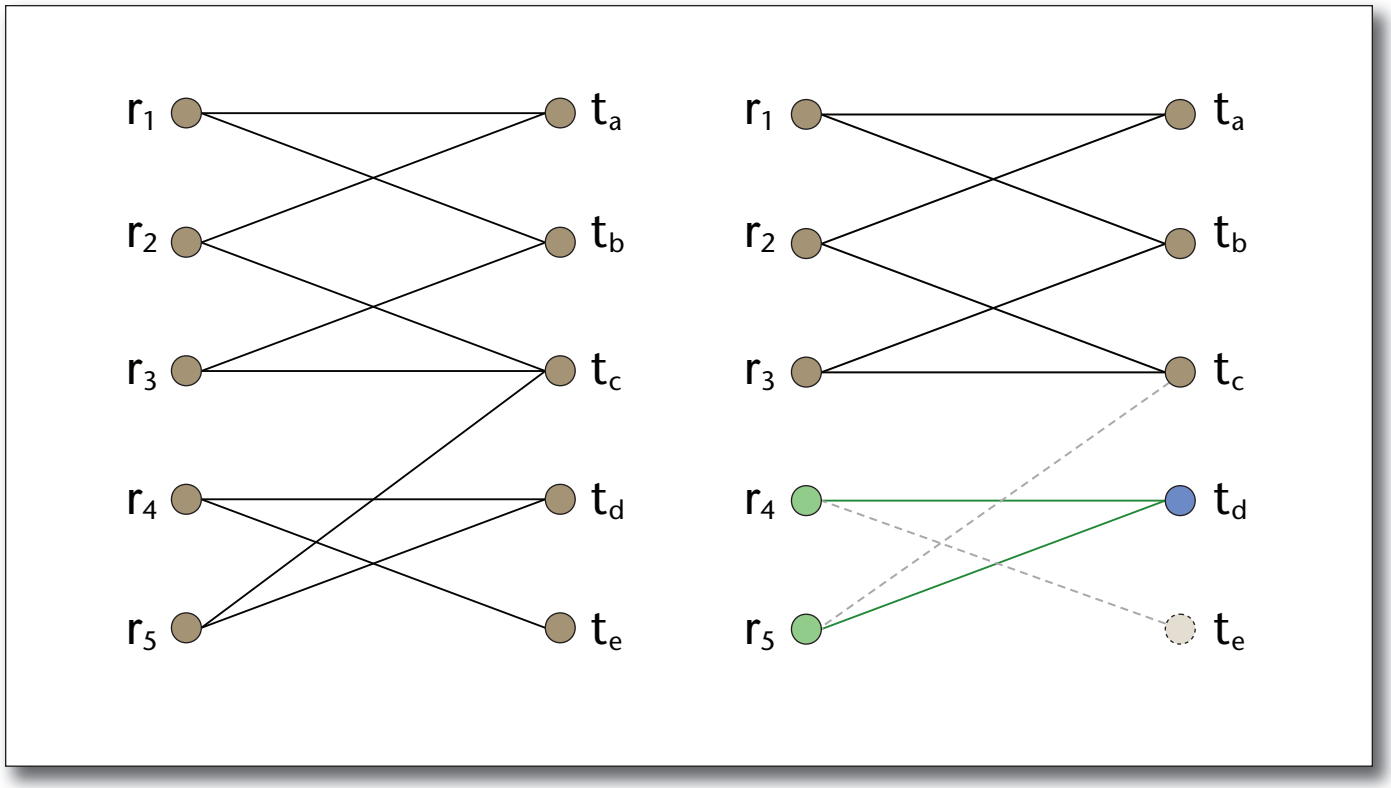


Figure 5. Preprocessing.

The edge (r_4, t_c) can be safely removed by rule 1, since r_4 is also covered by t_d, which covers another requirement. Rule 2 allows one to remove edges (r_5, t_c) and (r_5, t_d) as any test case covering r_4 also covers r_5. Finally, t_d is included in the solution set by rule 3, since it remains the only covering r_4 and r_5.

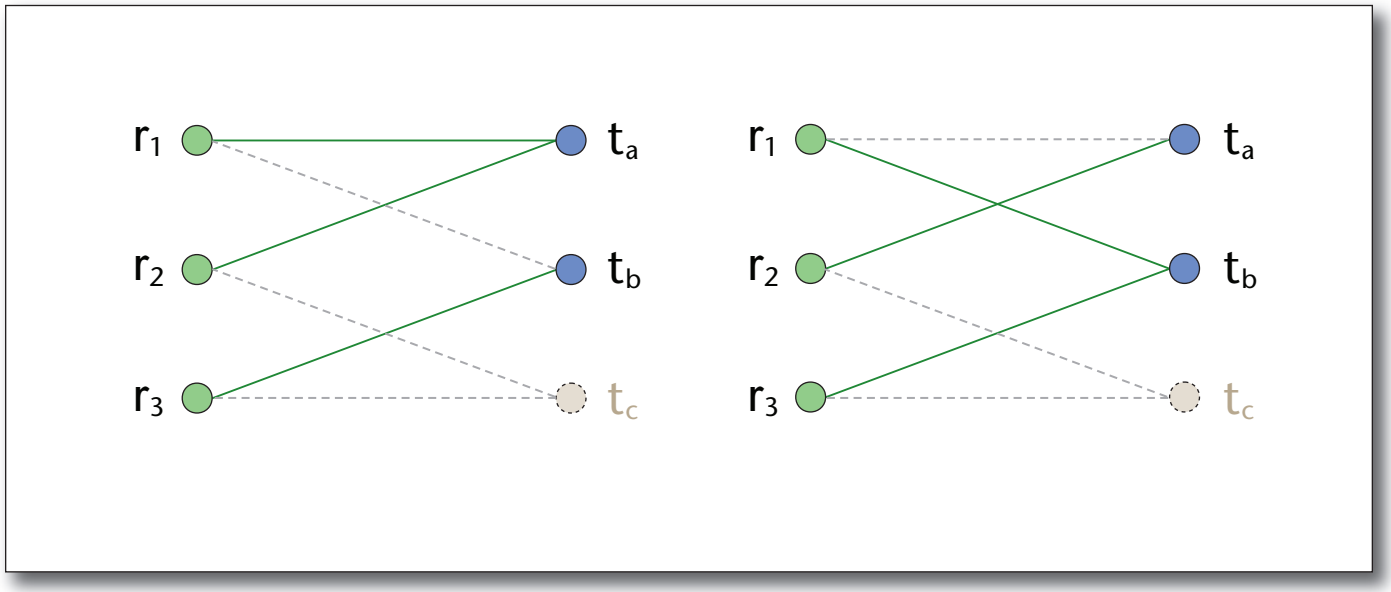
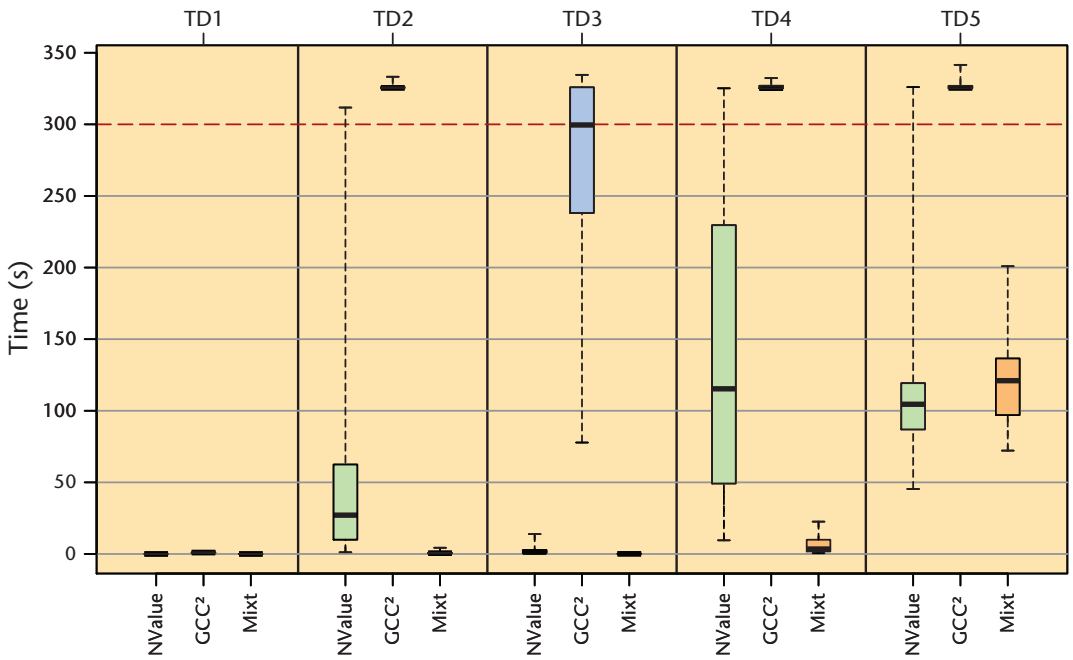


Figure 6. Two Symmetrical Solutions for CP, a Single Solution of TSR.

In both graphs, the same optimal test suite is obtained, $T' = \{t_a, t_b\}$. However, it is associated with distinct solutions for CP because the R_i are assigned to distinct values: on the left, R_1 is assigned to t_a while on the right R_1 is assigned to t_b. With our dedicated heuristic, an arbitrary selection is made, for example,, the occurrence variable O_a representing t_a is assigned to 2 as shown on the left. In case of necessary backtrack, it would be assigned to 0, but never to 1, as shown on the right.



	TD1	TD2	TD3	TD4	TD5
Requirements	20	90	60	60	30
Test cases	70	100	100	200	500
Density	8	20	20	20	8

Figure 7. Comparison of CPU Time for the CP Models.

(time-out = 300 seconds).

ring to Property 1, our TSR-dedicated heuristic guarantees that at least one optimal solution is found.

Experimental Evaluation

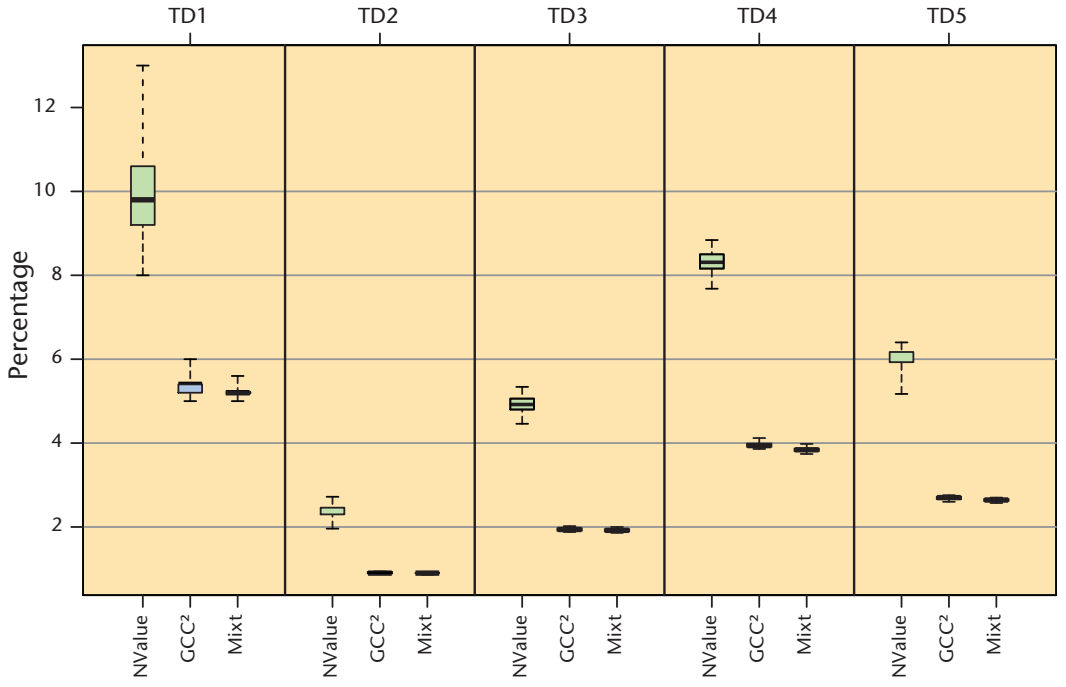
We implemented the constraint-optimization models and search heuristic described above in a tool called Flower/C, by using SICStus Prolog and its clpfd library. This library implements a finite domains constraint solver. Flower/C reads a file that contains the data about test cases, the covered requirements, and the costs associated to test cases and processes these data by constructing a corresponding bipartite graph and tuning the constraint-optimization models for solving the TSR instance. Solving the model involves preprocessing and search among feasible solutions with the proposed TSR-dedicated search heuristics. These steps are encoded in SICStus Prolog.

Both random and industrial instances of TSR were considered for the experimental evaluation. For random problems, we created a generator of TSR instances, which takes several parameters as inputs, such as the number of requirements, the number of test cases along with their associated costs, and the density of the relation *cov,* which is captured with *d* representing the maximum arity of any links in *cov.* The generator draws a number *a* at random between 1 and *d* and creates *a* edges in the bipartite graph, which represents *cov.* For industrial instances, we used data (test cases, coverage, costs) from regression testing of communication software provided by industry.

All our experiments were run on a standard i7-2929XM CPU machine at 2.5 GHz with 16 GB RAM.

Comparison of the Various CP Models

Figure 7 compares the CPU time required for finding optima with the three distinct CP models. In order to keep the comparison fair, we ignored costs in this first experiment so that optimality was only considered



	TD1	TD2	TD3	TD4	TD5
Requirements	250	500	1000	1000	1000
Test cases	500	5000	5000	5000	7000
Density	20	20	20	8	8

Figure 8. Comparison of Reduction Rate.

(as percentage of remaining test cases, time-out = 30 seconds).

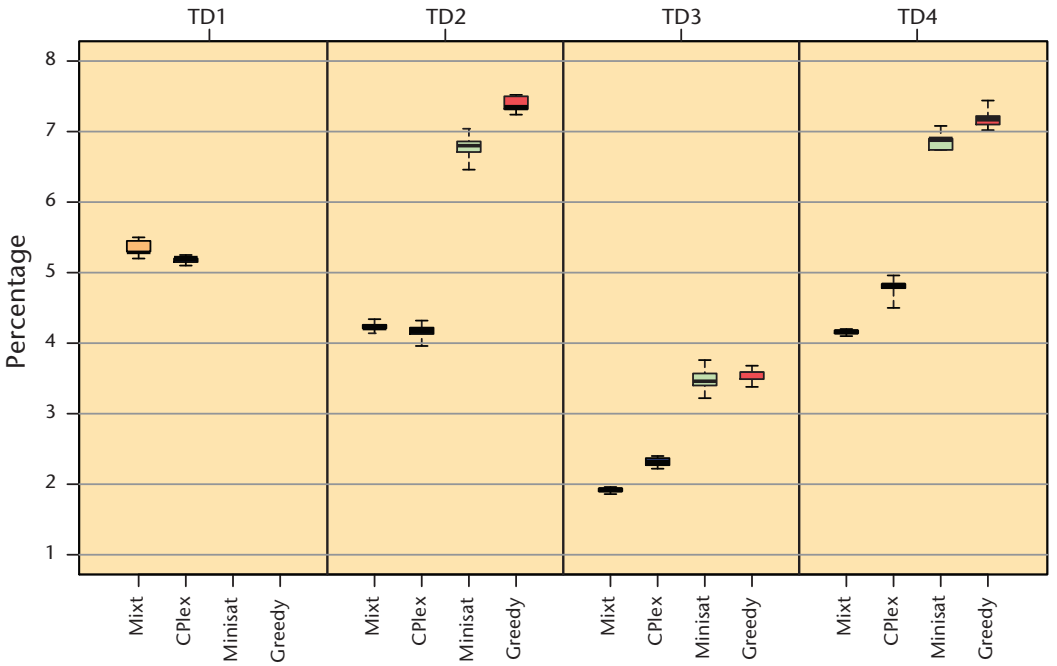
on the number of selected test cases. In each data set, 20 random samples were generated. For all but TD1, the *GCC*^{2} model times out (after 300 seconds). For the NVALUE model, we observe that the variation is very high in most cases (TD2, TD4, TD5). Sometimes, this models also times out. On the contrary, the Mixt model does not present much variation, which means that the TSR-dedicated heuristic is robust and useful in most cases. In figure 8, we compute the percentage of test cases remaining in the solution set after 30 seconds. A good reduction rate in a limited amount of time is crucial for any industrial adoption, as test suite reduction has to be performed within a continuous integration process, where the reduction is computed each time a new software release is committed.

We observe in this experiment that NVALUE is outperformed by both *GCC*^{2} and Mixt, which both reach the same reduction rate. This is due to the selection of the branching heuristic, which is different for the NVALUE model, where only the requirement variables are available for branching.

Comparison with Other Approaches

In the first experiment, we compared our implementation, Flower/C, with three other approaches, namely MINTS/MiniSAT+, MINTS/CPLEX, and Greedy on randomly generated instances. MINTS is a generic tool that handles the test suite reduction problem as an integer linear program (Hsu and Orso 2009). For each requirement to be covered, a linear inequality over Boolean variables is generated that enforces the coverage of the requirement. The Boolean variables ensure the selection of test cases. MINTS can be interfaced with distinct black-box constraint solvers, including MiniSAT+ and CPLEX. We also implemented a simple greedy approach for solving the TSR problem, which is based on a static ordering of the test cases covering the most requirements.

Figure 9 shows the results of comparison of the



	TD1	TD2	TD3	TD4
Requirements	1000	1000	1000	2000
Test cases	2000	5000	5000	5000
Density	20	7	20	20

Figure 9. Results of Comparison of the Four Approaches in Terms of Reduction Rate.

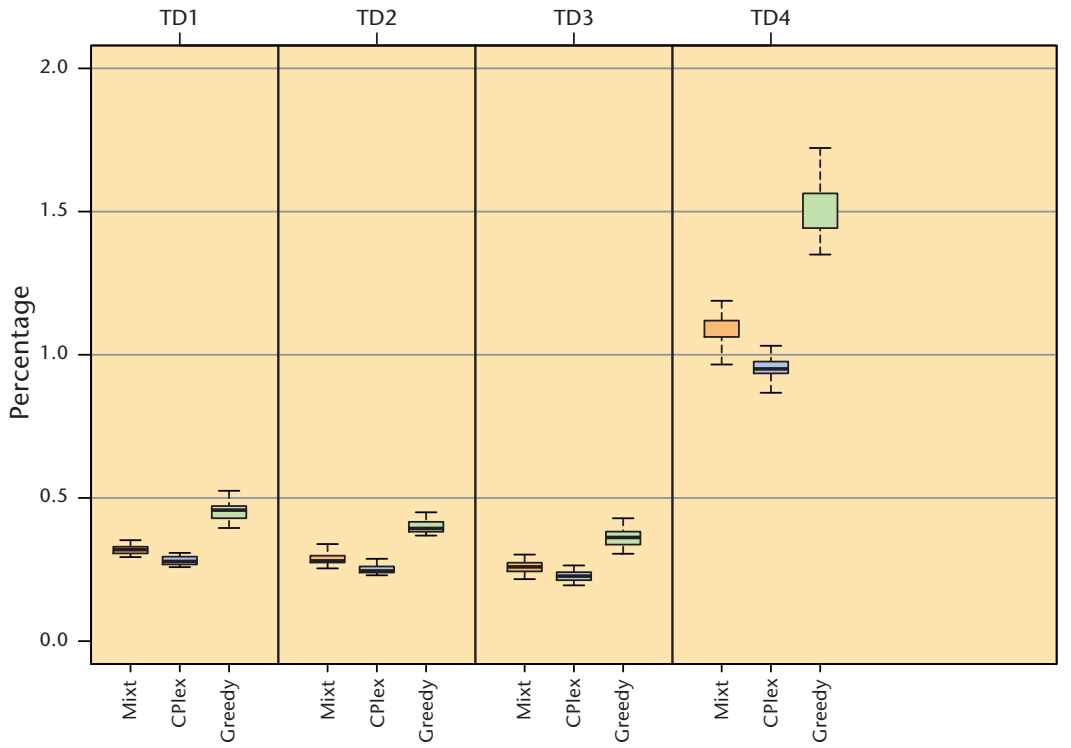
Comparison of reduction rate of Flower/C (Mixt), MINTS/MiniSAT+, MINTS/CPLEX, and Greedy on random instances with uniform costs (time-out = 60 seconds).

four approaches in terms of reduction rate, obtained in 60 seconds running time. In this experiment, the same cost values are used for all test cases. We observe that for the four groups of random instances (ranging from 1000 to 2000 requirements with two distinct maximal density values, 7 and 20), Flower/C achieves equal or better results than all the three other approaches in terms of reduction rate in a limited amount of time. Regrading the two last groups (TD3 and TD4), Flower/C performs strictly better than all the three other approaches, reaching exceptional reduction rates. It is worth noticing that for each group 100 random instances were generated, which means that the results are quite stable with respect to random variations. It is also quite clear that CPLEX performs much better for these problems than MiniSAT+. This does not come as a surprise as TSR has a simple formulation in terms of integer linear program (CPLEX), while MiniSAT+ requires translation into SAT clauses.

In the second experience, reported in figure 10, we gave different cost values to each instance by making the random generator select at random a value between 1 and a maximum value for each test case. In this experiment, no result is reported for MINTS/MiniSAT+ because the objective function as the sum of cost values cannot easily be encoded into Boolean SAT clauses. Therefore, only the results with MINTS/CPLEX, Flower/C, and Greedy are reported. Figure 10 shows that the results are in favor of MINTS/CPLEX on the four groups of random instances, which means that more effort is needed to find better CP models and search heuristics when costs are present.

Evaluation on Industrial Instances

We conducted the third experiment on industrial



	TD1	TD2	TD3	TD4
Requirements	1000	1000	1000	2000
Test cases	5000	5000	5000	5000
Density	20	20	20	20
Cost	50	100	1000	100

Figure 10. Results of Comparison of Reduction Rate on the Four Groups of Random Instances with Nonuniform Cost Values

instances coming from an industrial partner involved in the development of communication systems. The data were extracted from the continuous integration process during one cycle and converted to the specific format processed by Flower/C. The results are shown in table 2. The CPU time required to solve industrial instances of TSR shows that the Mixt model performs the best. Interestingly, the reduction rate shown in the fifth row (obtained with Mixt) is quite high for all the five industrial instances (ranging from 61.80 percent to 26.67 percent). This shows the importance of solving TSR in practice for our industrial partner. Finally, the last row of the table shows the number of removed requirements during preprocessing.

Evaluating Preprocessing Rules

We performed other experiments to evaluate precisely the effectiveness of preprocessing rules for both randomly generated TSR problems and industrial instances, as compared to the preprocessing used in MINTS/CPLEX. In figure 11, we evaluated the importance of MINTS/CPLEX's own preprocessing[1] in reaching an optimal solution by observing the size of the solution sets at different time points, and compared it with our own preprocessing. We found some data sets where Flower/C's preprocessing rules were more efficient than MINTS/CPLEX's preprocessing as shown in figure 11. However, there are also other cases where the opposite was observed. In fact, Flower/C preprocessing rules cannot be well compared with

Requirements	59	53	50	37	37	156
Test cases	107	90	93	100	100	377
CPU Time Nvalue(s)	0.00	0.10	0.01	0.01	0.01	0.03
CPU Time GCC2(s)	300.00	102.00	91.80	59.16	6.09	300.00
CPU Time Mixt(s)	0.00	0.01	0.00	0.00	0.00	0.01
Reduction rate (%)	28.97	26.67	29.03	40.00	37.00	61.80
Removed requirements (%)	32.00	30.19	30.00	32.43	45.95	44.87

Table 2. Evaluation of Flower/C on Industrial Instances.

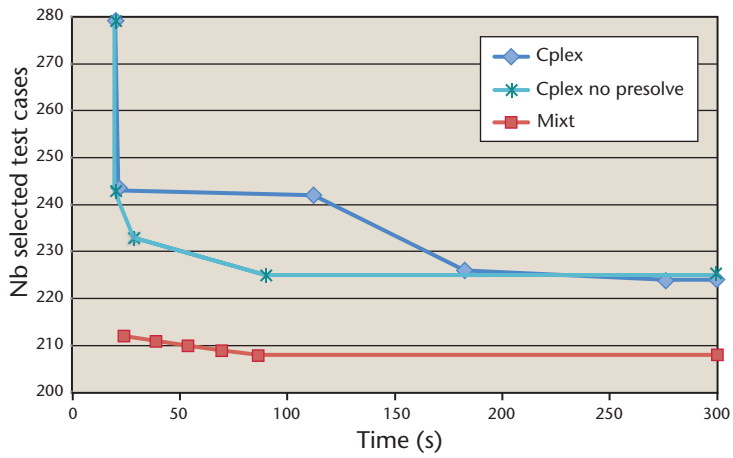


Figure 11. Evaluation of CPLEX Preprocessing versus Mixt.

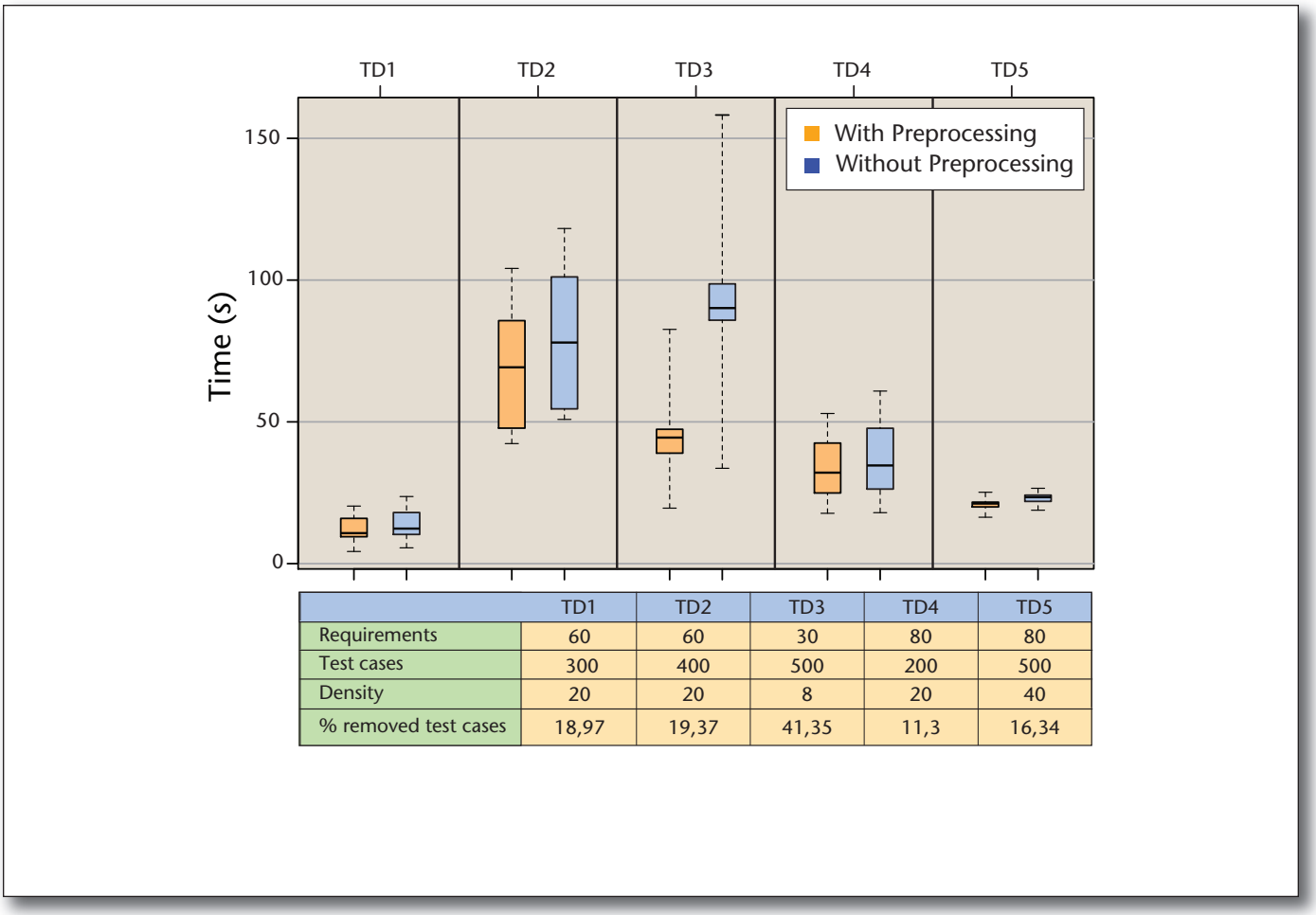
MINTS/CPLEX's preprocessing as both tools work on very different data structures. Finally, we looked at the gain in terms of CPU time while activating and deactivating Flower/C's preprocessing as shown in figure 12. The gain is not really spectacular even if the percentage of removed test cases is quite good.

Comparison of Several Search Heuristics

Figure 13 shows the CPU time for three variable-selection heuristics (that is, max, min, ff) used together with the CP Mixt model, while the value-selection heuristic remains unchanged. The heuristic max selects the variable with the greatest upper bound, min selects the variable with smallest lower bound, while ff selects the variable with the smallest domain. In this experiment, max achieves better result by selecting the occurrence variable that has the greatest arity, that is, the one associated with a test case that covers the most requirements. We selected it to be employed with our CP Mixt model.

Figure 14 compares different value-selection heuristics with max, including our own heuristics called value(enum), step, and bisect. The heuristic step branches on all the values of the domain of occurrence variables in increasing order, bisect performs domain-splitting using the middle point of the domain of each variable, while our heuristic only branches on *Max* and 0 for domain {0, 1, …, *Max*}.

As expected, figure 14 shows much better results for our heuristic. However, it is worth keeping in



	TD1	TD2	TD3	TD4	TD5
Requirements	60	60	30	80	80
Test cases	300	400	500	200	500
Density	20	20	8	20	40
% removed test cases	18,97	19,37	41,35	11,3	16,34

Figure 12. Comparison of CPU Time Versus Preprocessing.

mind that our strategy is incomplete. Even though it may not explore parts of the search space that contain optimal solutions, it preserves at least one optimal solution. When sufficient time is allocated to the search, it always has the opportunity to reach an optimal value faster than complete heuristics.

Path Toward Deployment

The work presented in this article has been motivated by the industrial problem of software regression testing in the communication domain. Software is characterized by a high degree of configurability, providing flexibility for end users to adapt systems to their specific needs. However, configurability involves higher complexity of software testing, and typically larger test suites. At the same time, software is developed following a continuous integration practice, which is characterized by a short test feedback loop. Extensive test suites, limited test time, and high requirements for software quality together set the challenge of implementing an efficient test suite

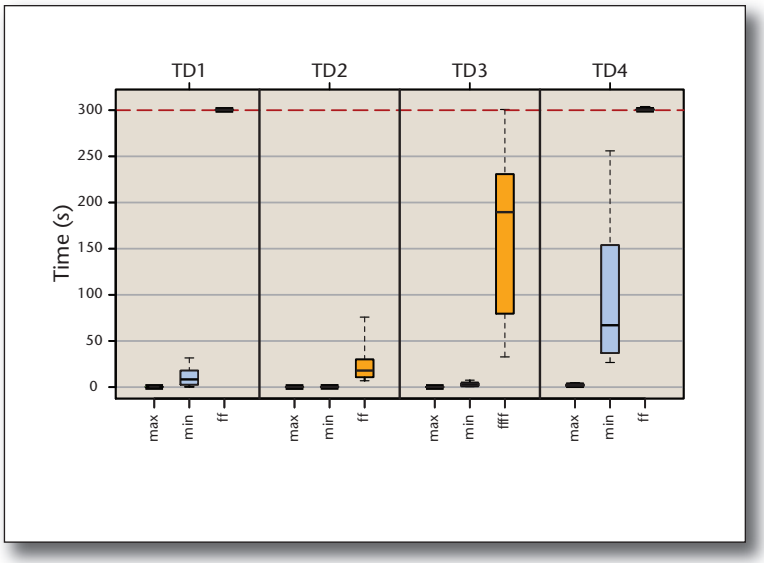


Figure 13. CPU Time of the Variable-Selection Heuristics.

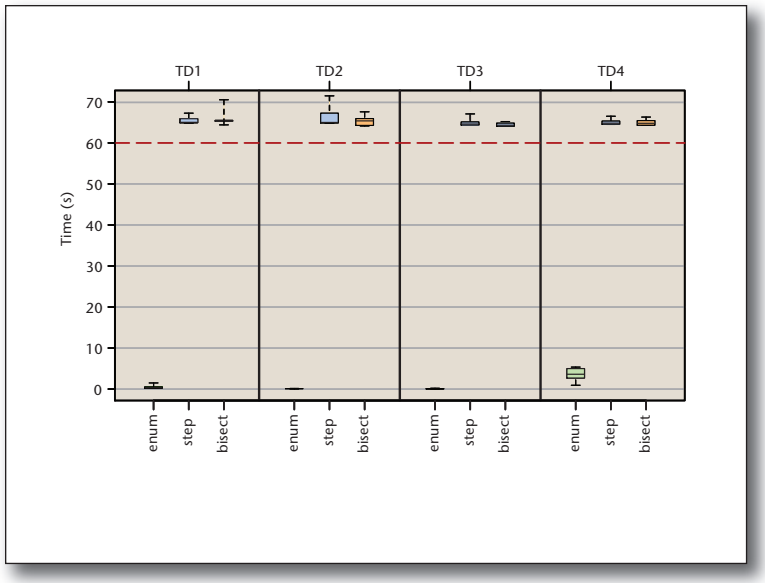


Figure 14. CPU Time of the Value-Selection Heuristics.

reduction that is able to reduce costs and improve the effectiveness of regression testing in practice.

Our approach has been designed in interaction with test engineers. The process involved observing current test selection practice done manually by engineers, interviews with engineers to understand the objective behind test selection, and capturing typical metrics such as the frequency and size of regression test runs, regression test selection criteria, and test failure rates. We evaluated the performance of the approach on several instances of industrial test suites coming from the described domain. The results shown that the approach is applicable, and that it can improve the speed and quality of regression test selection in continuous integration in practice. However, more work is needed to enable seamless integration with an industrial testing framework.

We see the deployment of our approach as a staged process. As part of first-phase deployment, we developed a prototype tool, and we provided training for engineers on the key concepts of CP used in our approach. We deem this step necessary for adopting the approach by industry, as we observed a limited familiarity with CP in this particular setting. We envisage full deployment as an iterative process, where we will be enhancing the tool functionality and usability based on industry feedback. The enhancements will relate more flexible test optimization, including test prioritization, to support achieving various testing objectives. At the final stage, we expect the tool to be deployed organizationwide, supporting cost-effective test automation much needed in complex continuous integration environments.

Conclusion

This article presents the application of CP techniques using global constraints to improve the cost efficiency of software regression testing. Three CP models using the global constraints NVALUE and GLOBALCARDINALITY are proposed to encode test suite reduction (TSR) in such a way to ensure the coverage of all user requirements while additionally minimizing the overall cost of a test suite. According to our knowledge, this is the first time that these global constraints are applied to the reduction of test suites in software testing. We find that some preprocessing rules can drastically reduce the size of initial problem instances and that our proposed TSR-dedicated strategy can outperform other more classical labeling heuristics such as those based on the first-fail principle.

Note that the proposed labeling heuristic is not complete, which means that it does not explore the overall search space. This may explain why it has a stronger competitive advantage over other heuristics. At the same time, this incompleteness in the search does not compromise reaching a true global optimum for the constraint-optimization model, since only symmetrical solutions are removed. The three CP models are compared on random instances with the state-of-the-art academic tool MINTS interfaced with MiniSAT+ and CPLEX. Our results show that CP is efficient and competitive with MINTS in terms of percentage of test suite reduction.

Furthermore, we evaluate our approach on industrial regression testing on communication software systems. Initial results show that the approach is useful for improving the speed and quality of regression test selection in continuous integration. However, there are challenges in fully applying complex CP techniques to testing in practice. Although the proposed search heuristics are quite efficient to prune the search space beforehand, we do not know yet if other heuristics could be more beneficial. Exploring this question is part of our planned further work. We also want to ease the adoption of CP-based solutions in industry by the design of tailored software tools encapsulating the complexity of constraint solving and providing scalable integrations with software testing tool chains.

Acknowledgments

This work is supported by the Research Council of Norway through the Certus SFI grant. Initial experiments were performed by Alexandre Pétillon. He was supported by Mats Carlsson from the Swedish Institute in Computer Science and got access to real data provided by Marius Liaaen from Cisco Systems, Norway. We would like to thank them all for their enthusiasm and support of this work. We are grateful to Jean-Charles Régin for fruitful discussions on the topic of the article.

Note

1. CPLEX processing can be deactivated on demand.

References

Campos, J.; Riboira, A.; Perez, A.; and Abreu, R. 2012. Gzoltar: An Eclipse Plug-In for Testing and Debugging. In *IEEE/ACM International Conference on Automated Software Engineering (ASE'12),* 378–381. New York: Association for Computing Machinery.

Chen, Z.; Zhang, *X.*; and Xu, B. 2008. A Degraded ILP Approach for Test Suite Reduction. Paper presented at the Twentieth International Conference on Software Engineering and Knowledge Engineering (SEKE'08), San Francisco, CA, USA, July 1–3.

Ferrer, J.; Kruse, P.; Chicano, F.; and Alba, E. 2015. Search Based Algorithms for Test Sequence Generation in Functional Testing. *Information and Software Technology* 58(0): 419–432.

Gotlieb, A., and Marijan, D. 2014. Flower: Optimal Test Suite Reduction as a Network Maximum Flow. In *International Symposium on Software Testing and Analysis,* ISSTA'14. New York: Association for Computing Machinery.

Hao, D.; Zhang, L.; Wu, *X.*; Mei, H.; and Rothermel, G. 2012. On-Demand Test Suite Reduction. In *Proceedings of the 34th International Conference on Software Engineering* (ICSE'12), 738–748. Los Alamitos, CA: IEEE Computer Society.

Harrold, M. J.; Gupta, R.; and Soffa, M. L. 1993. A Methodology for Controlling the Size of a Test Suite. *ACM Transactions on Software Engineering and Methodology* (TOSEM) 2(3): 270–285.

Hsu, H.-*Y.*, and Orso, A. 2009. MINTS: A General Framework and Tool for Supporting Test-Suite Minimization. In *Proceedings of the 31st International Conference on Software Engineering* (ICSE'09), 419–429. Los Alamitos, CA: IEEE Computer Society.

Jeffrey, D., and Gupta, N. 2005. Test Suite Reduction with Selective Redundancy. In *Proceedings of the 21st IEEE International Conference on Software Maintenance,* 549–558. Los Alamitos, CA: IEEE Computer Society.

Li, D.; Jin, Y.; Sahin, C.; Clause, J.; and Halfond, W. G. J. 2014. Integrated Energy-Directed Test Suite Optimization. In *International Symposium on Software Testing and Analysis,* ISSTA'14. New York: Association for Computing Machinery.

Mouthuy, S.; Deville, *Y.*; and Dooms, G. 2007. Global Constraint for the Set Covering Problem. Paper presented at Journées Francophones de Programmation par Contraintes,Université d'Orléans, Orléans, France, 3–5 June.

Offutt, A. J.; Pan, J.; and Voas, J. M. 1995. Procedures for Reducing the Size of Coverage-Based Test Sets. Paper presented at the Twelfth International Conference on Testing Computer Software, Washington, DC, June.

Pachet, F., and Roy, P. 1999. Automatic Generation of Music Programs. In *Principles and Practice of Constraint Programming,* volume 1713 of Lecture Notes in Computer Science. Berlin: Springer.

Régin, J.-C. 1996. Generalized Arc Consistency for Global Cardinality Constraint. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference*, 209–215. Menlo Park, CA: AAAI Press.

Rossi, F.; van Beek, P.; and Walsh, T. 2006. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Amsterdam: Elsevier Science.

Stolberg, S. 2009. Enabling Agile Testing Through Continuous Integration. In *Proceedings of the 2009 Agile Conference,* 369–374. Los Alamitos, CA: IEEE Computer Society.

Tallam, S., and Gupta, N. 2005. A Concept Analysis Inspired Greedy Algorithm for Test Suite Minimization. In *Proceedings of the 2005 ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering, PASTE'05,* 35–42. New York: Association for Computing Machinery.

Wang, S.; Ali, S.; and Gotlieb, A. 2015. Cost-Effective Test Suite Minimization in Product Lines Using Search Techniques. *Journal of Systems and Software* 103 (May): 370–391.

Arnaud Gotlieb Ph.D., is a senior research scientist at Simula Research Laboratory, where he leads the Certus Software Validation and Verification Center. He obtained his Ph.D. degree in computer science from the University of Nice-Sophia Antipolis in 2000. He worked for seven years in industry at Thales and then joined Inria, France, as a research scientist before moving to Simula, Norway. He coauthored more than 80 academic publications, led several research projects in software testing, cochaired several program committees including the SEIP track of ICSE'14 and the testing and verification track of CP'16.

Dusica Marijan Ph.D. is a research scientist in software engineering at Simula Research Laboratory, working on practical strategies for more cost-effective software testing. Her research interests include test automation and optimization, with a goal to help practitioners perform higher quality testing at lower costs and with fewer resources. Prior to joining Simula, she worked as a senior software engineer in the mobile and multimedia software industry.

Shakey: From Conception to History

Benjamin Kuipers, Edward A. Feigenbaum, Peter E. Hart, Nils J. Nilsson

■ *Shakey the Robot, conceived 50 years ago, was a seminal contribution to AI. Shakey perceived its world, planned how to achieve a goal, and acted to carry out that plan. This was revolutionary. At the 29th AAAI Conference on Artificial Intelligence, attendees gathered to celebrate Shakey and to gain insights into how the AI revolution moves ahead. The celebration included a panel that was chaired by Benjamin Kuipers and featured AI pioneers Ed Feigenbaum, Peter Hart, and Nils Nilsson. This article includes written versions of the contributions of those panelists. —ed.*

An Introduction

Benjamin Kuipers

At AAAI-15 (25–29 January 2015) in Austin, Texas, we met to celebrate the impact of the Shakey project, which took place from 1966 to 1972 at the Stanford Research Institute (now SRI International) in Menlo Park.

We researchers in artificial intelligence during this time in history have the privilege of working on some of the most fundamental and exciting scientific and engineering problems of all time: What is a mind? How can a physical object have a mind?

Some of the work going on today will appear in future textbooks, even centuries from now. We gain insights into our own struggles in the field today by learning about the historical struggles of great scientists of the past about whom we read in today's textbooks. The textbooks tempt us to think that they moved surely and confidently from questions to answers. In reality, they were frequently as confused then as we are now, by the mysterious phenomena they were trying to understand. When we read their history, we know the

  ISSN 0738-4602

answers they were seeking, and we can learn from the blind alleys they spent time in, and the insights that led them to the right paths.

Artificial intelligence marks its birth at the 1956 Dartmouth Conference. There have been many important milestones along the way. The important milestone we will celebrate today is the Shakey project, which created a physical robot that could perceive its environment and the objects within it. Shakey could make a plan to achieve a goal state. And it could carry out that plan with physical actions in the continuous world. The Shakey project laid a foundation for decades of subsequent research. We are here to celebrate and understand that project.

The centerpiece of the Shakey celebration was a panel presentation at AAAI-15, designed to give the audience an understanding and appreciation of the process of the research in the Shakey project, and of the long-term impact of that work on the larger field of AI. The goal was to have three speakers address (1) the state of the art in AI before the Shakey project (Ed Feigenbaum); (2) the progress of the Shakey project itself (Peter Hart); and (3) the impact of the Shakey project on the future of AI (Nils Nilsson).

Celebrating Shakey and Its Builders

Edward A. Feigenbaum

The history of science is a source of knowledge of the complex search for solutions to difficult problems. Not only is this history endlessly intriguing and awe-inspiring; but also it should be of particular interest to AI scientists because this kind of complex problem solving and discovery is at the heart of many of our theories of mental activity.

Life is lived in the moment. Everything else is memory and stories. The word *history* itself contains the word *story*. This talk is constructed as several stories of the Shakey project situated in its time, and among other landmark AI projects.

I have been lucky enough to have lived and worked through the entire 60 years of AI, from early 1956, months before the famous "founding" Dartmouth Conference, to today's AAAI-2015. My stories are drawn from those 60 years of memories, helped, but only a little, by the best memory assistant ever, the web.

My role today is to set the historical context in which the Shakey project was born, lived a remarkable but short life, and was terminated. Shakey research set the stage for decades of important experimental work in AI and robotics, and in other AI applications that will be mentioned later by Nils Nilsson.

I phoned several well-known robotics scientists to ask about the grandchildren of Shakey. All of them said the robots they developed were grandchildren of Shakey.

As shown in an original Shakey video, we remember Shakey as slowly and laboriously computing models of its environment; planning; moving and navigating its way around obstacles toward a goal on the far side of one large room.

Fast forward to some recent news about grandchildren of Shakey, from Manuela Veloso at Carnege Mellon University (CMU):

> I am very pleased to tell you that today, on November 18, 2014, the CoBot robots (3 of them) have jointly autonomously navigated for 1,000 km in our multi-floor SCS buildings at Carnegie Mellon University!

A great-grandchild of Shakey, Stanford's self-driving car Stanley, the car that drove itself across the Mojave Desert, is in the Smithsonian National Air and Space Museum in Washington, DC. Other cars like Stanley, built at Google, have driven more than 700,000 miles, navigating the San Francisco Bay Area and other roads, according to the *San Jose Mercury News* of November 12, 2014. (Consider this: Shakey's traversal, integrated over the whole life of the experiment, probably never made it to one kilometer).

Shakey's grandchildren on Mars are still having a productive long life — 11 years into a planned 90-day visit, semiautonomously assisting planetary scientists.

I would now like to tell you personal stories that together made the importance of Shakey research vivid to me.

First Story

In 1993, a major Japanese corporation asked me to do an evaluation of the quality of a robotics project that its research lab been working on for several years. After signing a nondisclosure agreement, I was shown a robot that was "humanoid," but very big (scary, actually). Tethered to a power source, its motion was fluid, a marvel of modern electromechanical engineering.

Though heavy, it could walk reliably without falling, and it could even climb a flight of stairs. But this creature had no Mind. It did no symbolic processing, no problem solving. It did not have goal-directed behavior.

There was more than enough space inside for a PC-sized computer and there was plenty of power. What this project lacked were scientists and engineers trained in AI, or even trained in software systems. There were no young Nils Nilssons, no young Peter Harts, no young Bert Raphaels or Richard Fikes — and of course no visionary like Charles Rosen to integrate AI with electromechanical engineering.

And this was 1993, twenty years after the end of the Shakey project! It can be perilous to ignore scientific history.

Second Story

The Computer History Museum in Mountain View, California, is the world's premier museum for the history of computers and information technology and is

recognized for its interpretation of that history. In January 2011, the museum opened its permanent exhibition, called Revolution. Here is a quote from the museum's press release:

> Ten years in the making, Revolution is the product of the Museum's professional staff collaborating with designers, content producers and more than 200 experts, pioneers and historians around the world.

Revolution showcases 20 different areas of computers, computer science, semiconductors, and communications, from early history to futuristic visions. Among those 20 areas is one called AI and Robotics.

For each area, the museum staff has chosen one historical artifact to be the icon exhibit for the area. For AI and Robotics, the icon is Shakey the Robot, beautifully exhibited.

The museum could have chosen any one of a dozen or more landmark AI artifacts. It could have chosen AI's first heuristic problem-solving program (the Logic Theorist of Newell, Shaw, and Simon); or a speech-understanding program from Reddy; or one of the early expert systems from our Stanford group; or Deep Blue, the AI system that beat the world's chess champion. It could have … but in the end the museum chose Shakey.

So let's put the first story and the second story together to make a:

Third Story

SRI's Shakey work was a decade or two ahead of its time in demonstrating the power of integrating AI with robotics. Remarkably, even today, when robotics is being taught to high school students, and computing and sensors cost almost nothing, most robots in labs and companies do not have the AI capabilities that Shakey had in the 1970s.

Historians of the field have given Shakey deserved recognition, but the field of AI had not. It took a while for an AAAI national program committee to recognize this and make room for this celebration. I want to thank the AAAI-15 program committee, and hope that this will be a model for bringing forth other important parts of AI's history.

Fourth Story

The Shakey Project was done from 1966 to 1972. What was AI and computer technology like before and during that period?

There is a generation of younger researchers that have no idea how few were the powerful ideas of the first decade of AI (1956 to 1966) to build upon for new AI systems. Nor can that younger generation envision the lack of power of the computers that we had upon which to build these systems.

But there was no lack of enthusiasm, and excitement; no lack of interaction, because almost everyone in the field knew almost everyone; and we all read each other's papers, tech reports, and books. That's what it's like, when a field is small and emerging.

The AI science had a workable set of ideas about how to use heuristic search to solve problems. But proving things about heuristic search had to wait until later (the Shakey group's A*). Some powerful successful experiments had been done: the Logic Theorist; Gelernter's Geometry Theorem Proving program; Slagle's calculus problem solving programs are examples. These were all on the "cognitive" side of AI work. On this side, much discussion and energy was focused on generality in problem solving: Newell and Simon with means-ends analysis; McCarthy and other "logicists" with theorem proving.

On the "perceptual" side of AI work, a similar story can be told about research on vision. There were several basic workable techniques involving line finding, curve finding, and putting elements together into logical descriptions of objects. Generality of the techniques was also an issue, as it still is today.

What did we have with which to do this work? Our programming languages were great! List processing was invented at CMU and then made more powerful and beautiful in LISP at the Massachusetts Institute of Technology (MIT). But there was almost no interaction between people and computers. Time-shared interaction did not become available to most researchers in this first decade.

Try to imagine this about computer processing power and memory: I did my thesis work on an IBM 650 computer in the late 1950s: maximum 2500 operations per second; memory was 20,000 digits (what we would now call bytes). Not only your program, but your language interpreter had to fit into this memory. There was no virtual memory.

In 1959, the IBM's large multimillion-dollar transistorized computer was introduced. It ran at 100K FLOPS, and had about 150K bytes of main memory. The largest DEC computer that would have been available in 1966 for the Shakey group to buy was the PDP-6, which operated at 250,000 additions per second with a memory of about 150K bytes.

Compare these numbers with, say, today's Apple MacPro at four gigaops/sec with memory of 16 gigabytes; or even today's smartphones at about 1 gigaop/sec but with memories going up to 128 gigabytes.

Fifth Story

All projects end, even the great ones. The DARPA funding pendulum for support of AI swung away from robotics and toward both knowledge-based systems and the national speech understanding project. As funding shifted, SRI continued to do world-class work in both of these other themes of the 1970s.

Final Story

The Shakey project, as cutting edge work in computer science, inspired young people to do great things. In an email to Eric Horvitz, former president of AAAI,

let me quote from one of these people, a junior in high school at the time. He and a high-school friend traveled to visit the Shakey project in 1971, unannounced, but were welcomed by the Shakey team.

> I was inspired by the Shakey video from SRI. I actually went down and visited when I was a junior in high school and they showed me the lab.
>
> Shakey was pretty cool — vision, modeling, planning. It decided to move things around so it could go up a ramp.
>
> Paul — do you remember how we got this video?

The "Paul" is … Paul Allen; and the author of the quote is Bill Gates.

Making Shakey

Peter E. Hart

The proposal that launched the Shakey project was submitted by the Artificial Intelligence Center of Stanford Research Institute (now SRI International) in January, 1965. SRI proposed to develop "intelligence automata" for "reconnaissance applications." But the research motivation — and this was the inspiration of Charles A. Rosen, the driving force behind the proposal — was to develop an experimental test bed for integrating all the subfields of artificial intelligence as then understood. SRI wanted to integrate in one system representation and reasoning, planning, machine learning, computer vision, natural language understanding, even speech understanding, for the first time.

Readers interested in technical details of Shakey's development will find an excellent summary[1] in an SRI report. A 25-minute video, made by the Shakey team at the time, is available.[2]

The design of the "automaton," as it was initially called (perhaps out of a justifiable concern that "robot" sounded like science fiction, which it was before Shakey), was governed by two ground rules: First, in order to keep it mechanically as simple as possible, no arm was installed. And second, to avoid issues of miniaturization, the design evolved as an electronics rack on wheels with a sensor assembly mounted on top.

The project team was well aware of Shakey's limited mechanical and sensory capabilities, and designed a correspondingly simple experimental environment consisting of half a dozen rooms populated with large, geometric blocks. The blocks were painted so that edges were visible to the low-resolution TV camera, while still being sufficiently reflective for our homemade laser rangefinder to work. We also used dark baseboards, again for visibility, and exploited them to update the position error that accumulated in the dead reckoning process that relied on Shakey's stepping motors.

Our first computer was an SDS 940, an early commercial time-shared mainframe (whose main memory was smaller than the L2 cache of most laptops). In 1970 we upgraded to a more powerful DEC PDP-10. Shakey talked to the PDP-10 through a communications processor, and the system was one of the handful of nodes that constituted the birth of the ARPANET. Around this time we embarked on a complete rewrite of much of Shakey's software, while making only minor upgrades to the robot hardware. In the next section we describe this version 2 of Shakey.

Figure 1. Charles A. Rosen and the "Automaton."

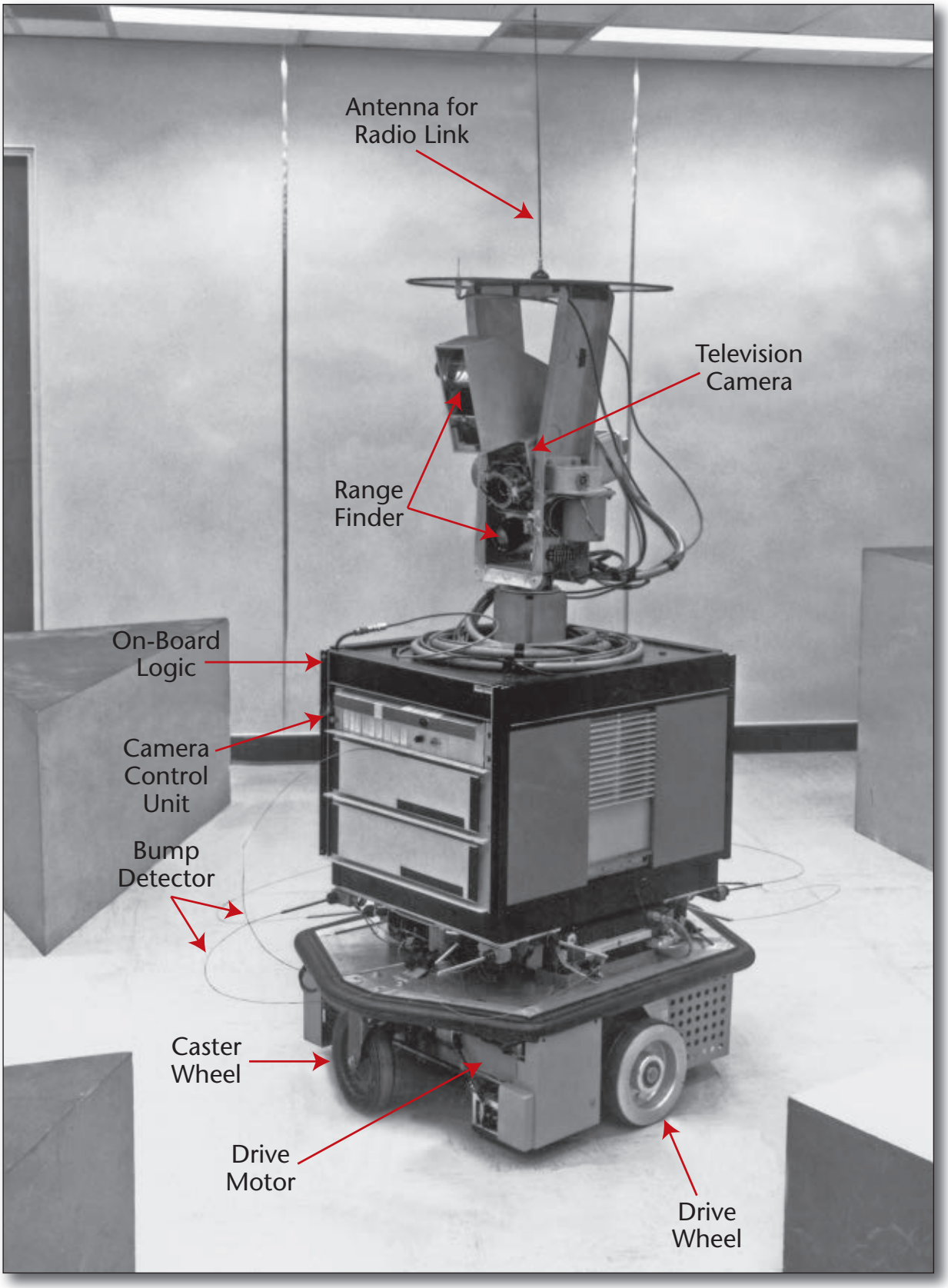


Figure 2. Shakey with Components Labeled.

Shakey's Control Software

There were two big ideas behind version 2. The first idea was to represent Shakey's world by statements in the first-order predicate calculus, augmenting a form of grid model that was a key component of the first version (figure 4).

The second idea was to structure Shakey's control software as a series of layers, the first time this design was used to control a robot (figure 5). In the following we briefly describe each layer, beginning with the low-level actions.

Low-Level Actions

Low-level actions like ROLL and PAN talked directly to Shakey's hardware (figure 6). Also in this layer are actions like PANTO, which rotates the "head" to a specified orientation.

Intermediate-Level Actions: Markov tables

Above this level are intermediate-level actions like GOTHRUDOOR. These actions are put in their own layer because all of them are represented as Markov tables (figure 7).

One interprets a Markov table by scanning down the left column until the first true condition is reached, executing the corresponding action, and then looping back to the top. Accordingly, Markov tables have an inherent perseverance: they keep trying to do something useful. (This account is slightly simplified, but the looping behavior is fundamental and, as we'll see, an important feature of these tables.)

If these intermediate-level actions were the end of the software story, Shakey would be very limited in what it could achieve. It would only be able to achieve goals that require just a single preprogrammed action. To do more, Shakey has to be able to compose a sequence of actions into a plan. That's the job of STRIPS, the Stanford Research Institute Problem Solver, which constitutes the next higher software level.

STRIPS, the Stanford Research Institute Problem Solver

STRIPS came about by combining two big ideas of the day. The first was the planning strategy called means-ends analysis, as exemplified by the General Problem Solver program of Newell and Simon.

The second big idea was theorem proving in the predicate calculus and its application to question answering systems, as exemplified by the work of Cordell Green. Richard Fikes and Nils Nilsson combined these ideas to create STRIPS (Fikes and Nilsson 1971), which applied means-ends analysis to predicate calculus representations (figure 8).

PLANEX, the Plan Execution Executive

Shortly after designing STRIPS, the SRI team found a way to generalize a STRIPS plan by replacing constants in the plan with variables. They also invented a data structure called a triangle table that represents the internal dependencies of a generalized plan. These constructs formed the basis of PLANEX, the Plan Execution Executive that is the top layer of Shakey's control software (Fikes, Hart, and Nilsson 1972).

Figure 3. Triple Exposure of Shakey Moving Among Boxes.

Using this software machinery, PLANEX could monitor the real-world execution of a plan. It could detect if something had gone wrong, and could replan from that point, reusing portions of the existing plan wherever possible. It could even be "opportunistic": If by chance Shakey was closer to achieving its goal than anticipated, it could capitalize on its good fortune.

This error detection and recovery ability was a critically important part of Shakey's control software. A chasm separates planning for a physical robot, that has to execute plans in the real world where things often go wrong, and an "abstract" planner that merely needs to print out a symbolic plan once it is computed. The Plan Execution Executive, together with those persevering Markov tables, was the solution to the problem of achieving robust, real-world plan execution.

Computer Vision

The initial project plan did not call for intensive research in computer vision. Rather, the plan was to integrate existing computer vision techniques into the experimental test bed. But, as it turned out, very little technology was available, so a focused effort in computer vision was started.

One important result of this work was the invention of what could be called the modern form of the Hough transform for finding lines in images (Duda and Hart 1972). This result came about by combining two concepts that on the surface appear unrelated.

The first idea is contained in a patent by Paul Hough, in which he described a transform from points in an image plane to straight lines in a trans-

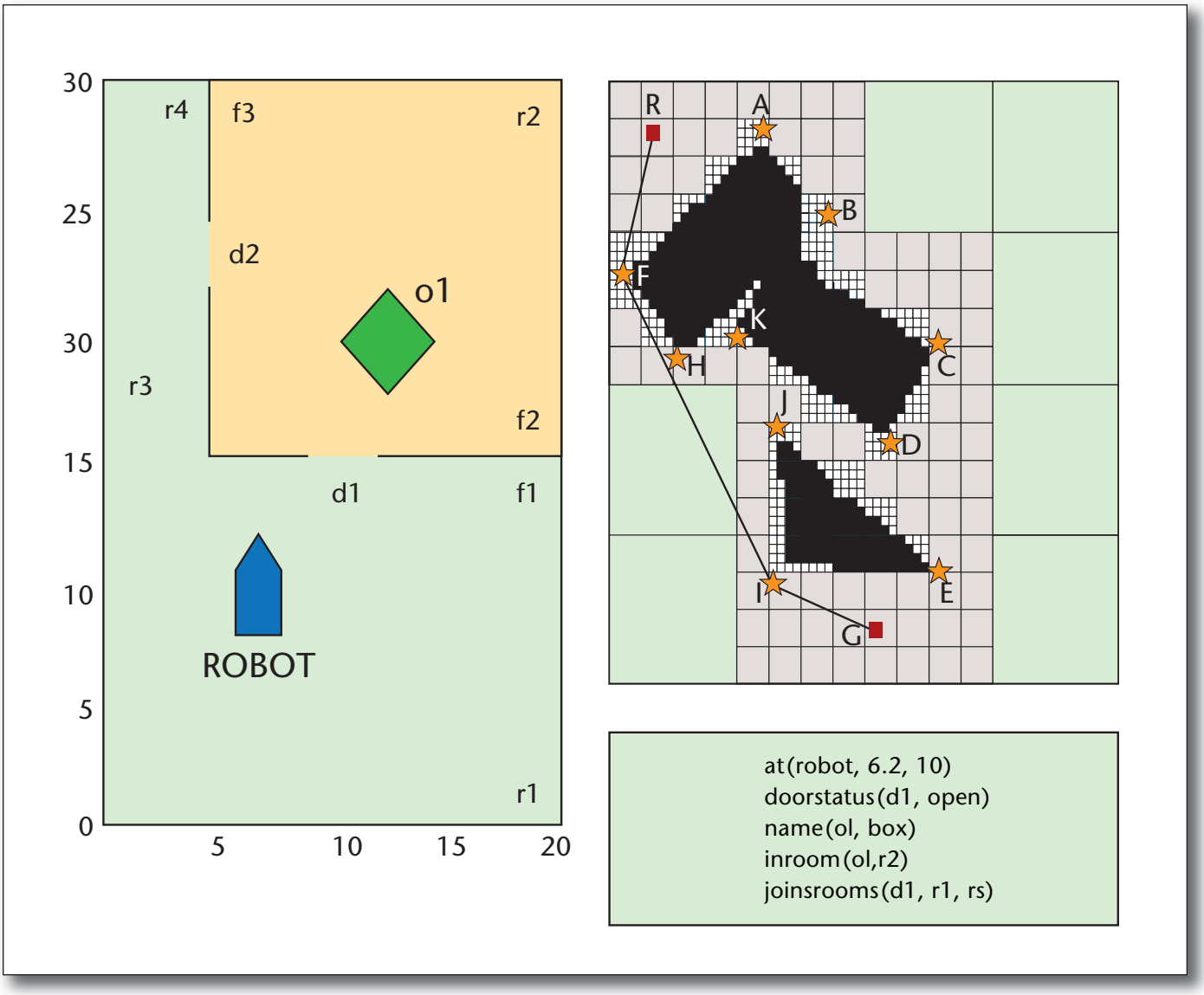


Figure 4. Predicate Calculus Model Fragment with Plan View of World and Grid Model

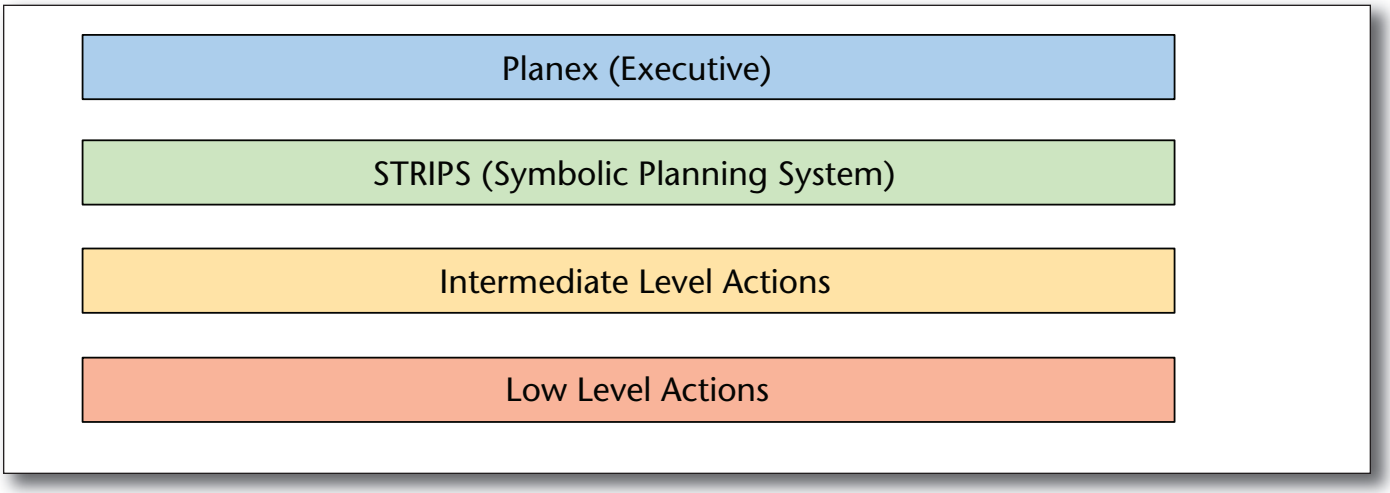


Figure 5. Layered Software Architecture.

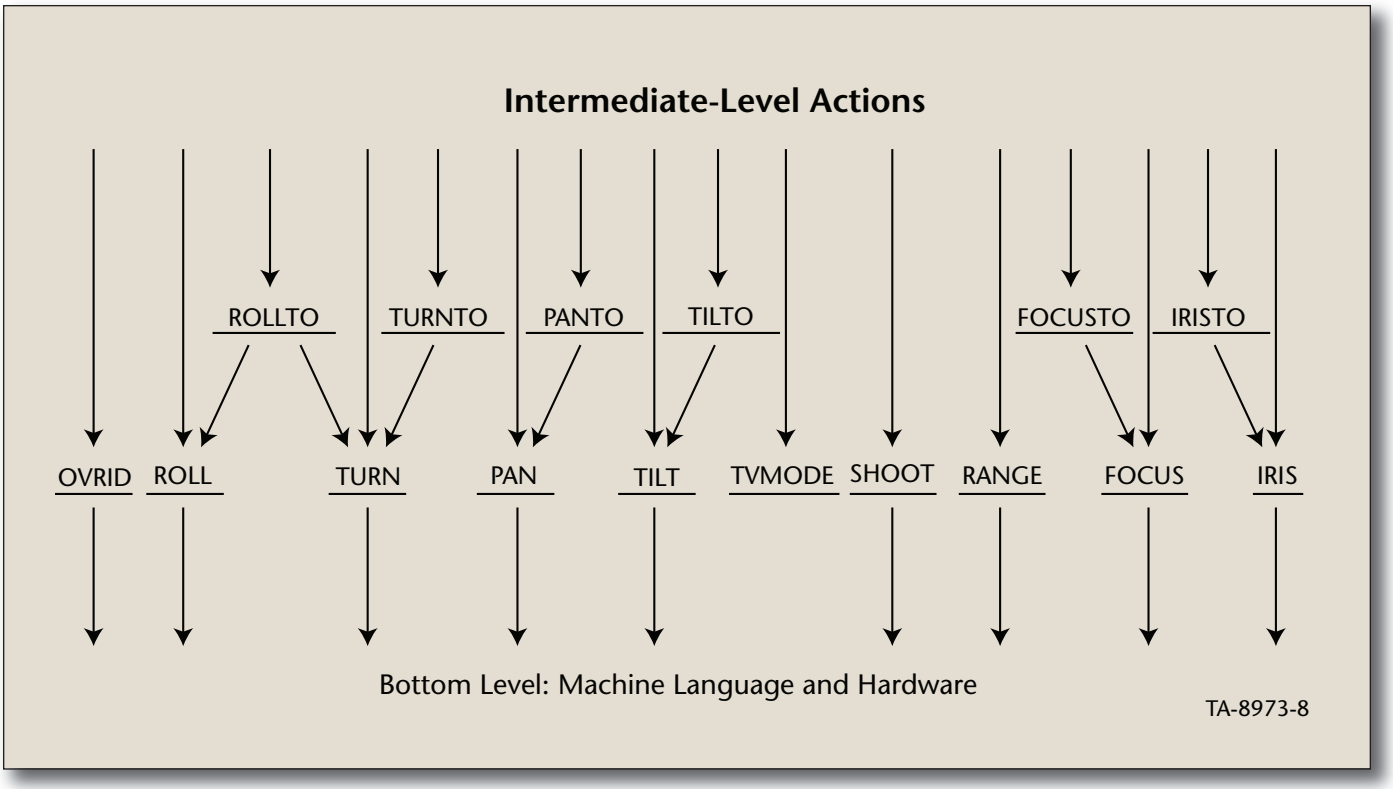


Figure 6. Low-Level Actions.

Condition	Action
infrontof(door) /\eq(s,OPEN)	bumblethru(room1,door,room2)
near(door) /\eq(s,OPEN)	align(room1,door,room2)
near(door) /\eq(s,UNKNOWN)	doorpic(door)
eq(s,CLOSED)	return [fail]
T	navto(nearpt(room1,door))

Figure 7. Markov Table for GOTHRUDOOR.

form space. Intersecting lines in the latter correspond to collinear points in the form. But a problem of infinite slopes arises that makes this transform computationally unwieldy (figure 9).

The second idea comes from an obscure branch of 19th century mathematics called integral geometry. Mathematicians had theoretical reasons for using an angle-radius parameterization of a line, rather than the more familiar slope intercept used by Hough. Peter Hart noticed that by replacing Hough's linear transform with a sinusoidal one, not only is the problem of infinite slopes avoided, but the new transform is invariant to choice of coordinate accesses. Hart and Richard Duda also extended this method to detect analytic curves in images, and this transform has been used ever since.

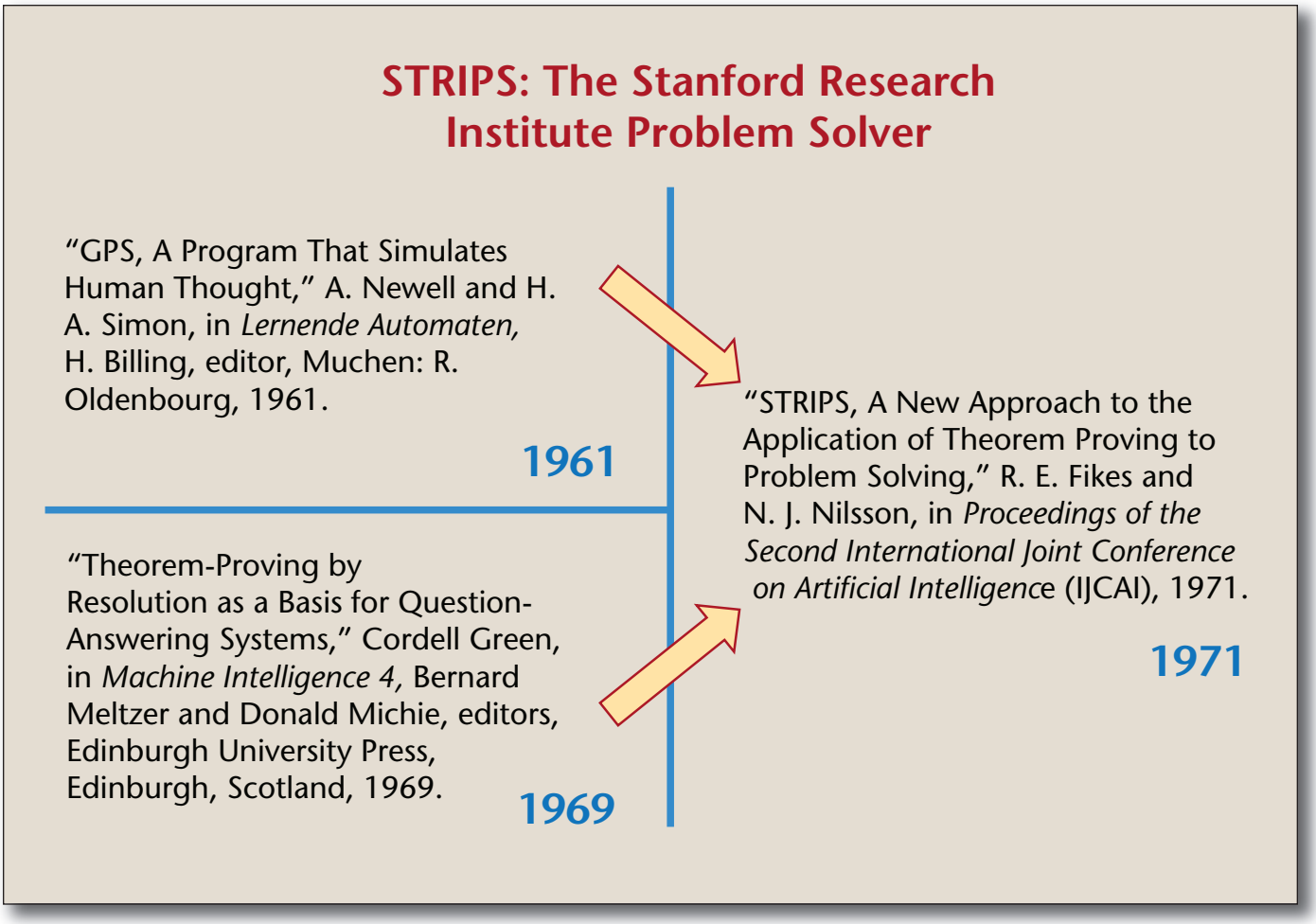


Figure 8. STRIPS.

Navigation and the A* Algorithm

Shakey had to find its way around, so several shortest-path algorithms were developed. One, called A* by its creators, Peter Hart, Nils Nilsson, and Bertram Raphael, had two very desirable properties. It can be rigorously proved that (a) it always finds the shortest path, and (b) that it does so while considering the smallest possible number of alternatives. In non-mathematical shorthand, we can say that it always works and it's computationally efficient.

One would think that such a strong result would be eagerly accepted by any reputable publication, but it's perhaps a sign of those times that just the opposite was the case. The A* manuscript was rejected by the most prestigious journals of the day. Looking at those old reviews, we can speculate that review editors sent the manuscript to mathematicians, because of all those intimidating-looking theorems. But mathematicians were unimpressed because the proofs were limited to graphs with only a finite number of nodes. It seemed to the authors at the time that mathematicians saw no difference between a graph with ten nodes and one with ten trillion nodes, but to computer scientists that difference matters.

The paper (Hart, Nilsson, and Raphael 1968) was finally accepted by the *IEEE Transactions on Systems Science and Cybernetics,* where it eventually got noticed and continues to be referenced more than 45 years after it was published.

The World Back Then

The foregoing gives a glimpse of some (though by no means all!) of the work done by the Shakey project team. To place this work in a broader societal context we can take a brief look at the intellectual and cultural climate of the time.

In 1970, *Life,*[3] a popular magazine of the day, ran a big story about Shakey (figure 10). The author, journalist Brad Darrach, seemed to hyperventilate a bit, with a subtitle, "the fascinating and fearsome reality of a machine with a mind of its own." But while some like Darrach worried that machines might take over the world, there were deep skeptics. Hubert Dreyfus was one, who argued on deep philosophical grounds that AI is in principle impossible. And some-

where between Brad Darrach and Hubert Dreyfus were labor union leaders who worried that robots might some day take manufacturing jobs.

Charlie Rosen was undaunted by the critics, noting that there will always be "naysayers," as he called them, whenever something new is done. The best response is to push onward.

Shakey's Visitors

The Shakey team was generous with its time and welcomed virtually any visitor who was interested in the work. We can get another perspective on the world back then by viewing it through their eyes. Here are some examples:

A school group visited, and a teacher asked what our "real jobs" were. "This robot is your hobby, isn't it?"

A general visited and asked "Can you mount a 36-inch blade on that?"

Arthur C. Clark visited just after the movie *2001* appeared, but was more interested in talking about the *New York Times* review of the movie than about the future of robots.

A young high school student drove all the way from Seattle to Menlo Park, California, to see Shakey. Decades later Bill Gates recalled being impressed.

A US government auditor visited and asked whether SRI had indeed taken delivery of billions of "packets of bits." This question was followed by others regarding the state of those packets, including whether there was any tarnish or corrosion on any of those bits.

The End of the Shakey Project

The Shakey project ended in 1972, not for lack of exciting ideas to pursue, but because the funding climate had changed and the research program became unsupportable. What had been achieved, as viewed from the perspective of 1972?

While there are likely as many views as there were project team members, it seems safe to make a few broad generalizations:

There was an appreciation that many of the individual results — STRIPS, PLANEX, A*, and the new form of the Hough transform are good examples — were solid technical contributions.

Overall, Shakey was a significant achievement, being both the first mobile, intelligent robot, and also being the first system that integrated AI software with physical hardware.

But Shakey's overall capabilities, both mechanical and software, didn't reach the level of the initial aspirations. This would hardly be surprising, given those lofty early goals. Indeed, it would take decades before some were reached, while others remain as research challenges.

Today's perspective is very different from the view in 1972. Shakey has had impacts on both current research and on the everyday lives of all of us that could not have been recognized or anticipated at the time. Those impacts are the subject of the remaining sections of this article.

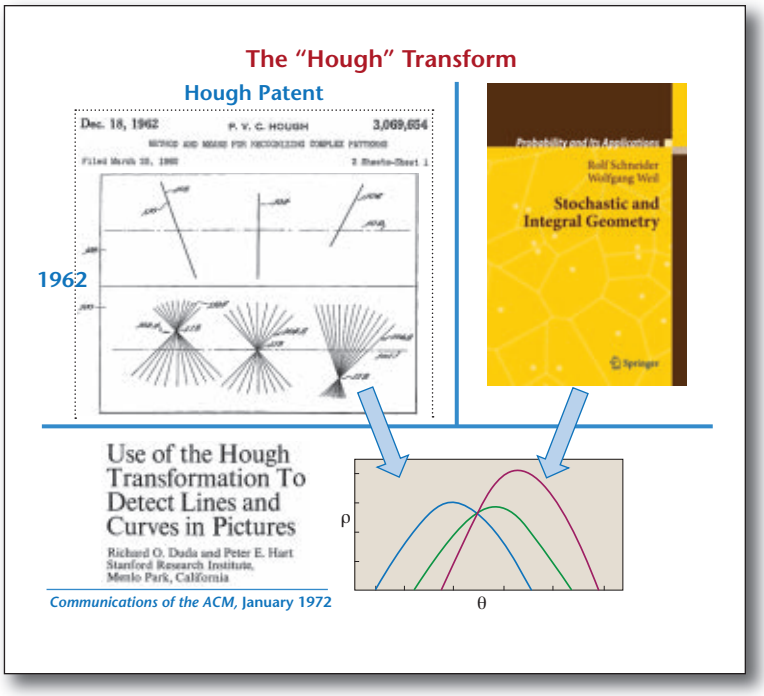


Figure 9. The Hough Transform.

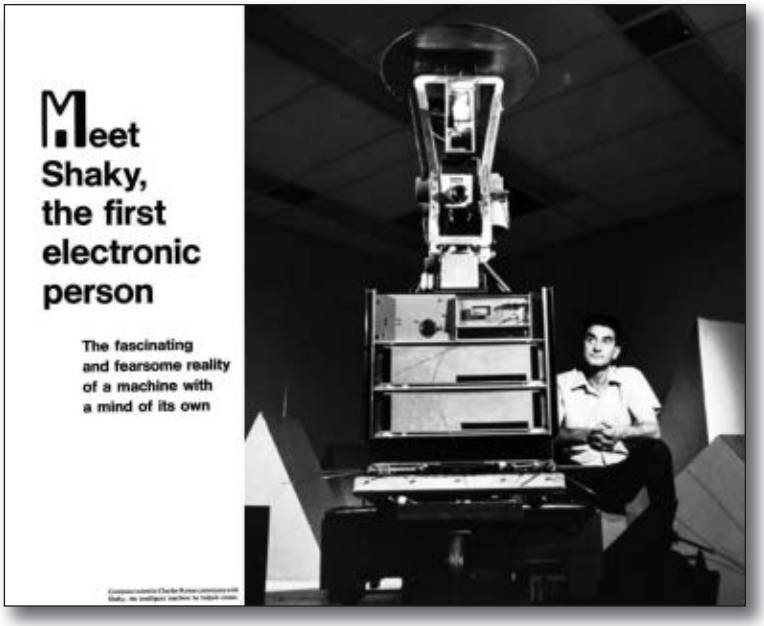


Figure 10. Shakey Story in Life Magazine.
Photo © by Ralph Crane, *The LIFE Picture Collection*.

Shakey's Legacy

Nils J. Nilsson

"Shakey the Robot" was the first system that integrated artificial intelligence programs (most of which were newly developed during the project) with physical hardware. In this part of the panel discussion,

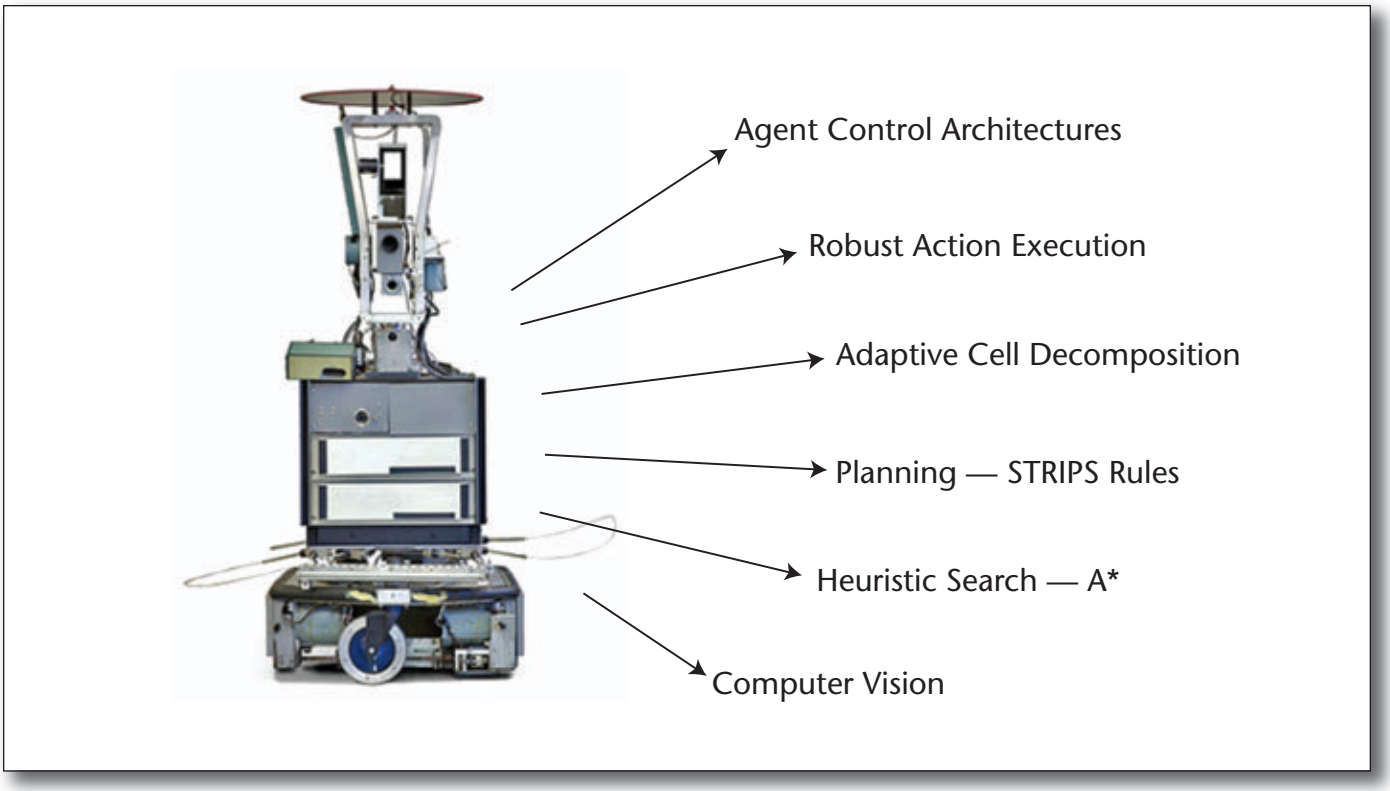


Figure 11. Shakey's Achievements.

Nils Nilsson used the chart (figure 11) to list some major achievements of the project. In greatly elaborated and extended form, descendants of some of this software are still in use today.

Agent Control Architectures

Although hierarchies and layers had previously been used in software systems, Shakey was the first robot to be controlled by a layered architecture. As illustrated earlier, there were four main layers, namely, the PLANEX (executive), STRIPS (symbolic planning system), the intermediate-level actions, and the low-level actions. Layered control architectures have been used in several subsequent robot systems, among them the DS1's "Remote Agent" (RAX), which controlled a space craft (Bernard et al. 1999), and the Monterey Bay Research Institute's (MBARI) autonomous underwater vehicle (McGann et al. 2008). Of course the control architectures of these modern systems, although layered as Shakey's were, are much more complex.

Robust Action Execution

As described earlier, Shakey used two main techniques to guarantee robust action execution. One was Markov tables, which scanned a list of conditions to find the first one that was satisfied by the current situation and then invoked the corresponding intermediate level action. The second was a structure we called a "triangle table," which stored preconditions and actions assembled by the STRIPS planning system. These techniques evoked actions that were both reactive to the current situation and opportunistic in unforeseen situations.

One follow-on to those techniques used by Shakey is the concept of teleo-reactive (T-R) programs developed by Nilsson and his students during the 1990s (Nilsson 1994) (figure 12).

That action associated with the first currently satisfied condition in the list (or tree) is the one that is executed. But execution continues only so long as that condition remains the first one currently satisfied. As soon as it is no longer satisfied, the list is scanned again to find the one now first satisfied, and so on. In the T-R formalism, the actions could themselves be T-R programs. Dozens of papers have been written about T-R programs, one book (Clark and Robinson 2016) is soon to appear, and many robots have been controlled by them.[4]

Another control technique uses structures called "hierarchical state machines." Hierarchical state machines are similar to T-R programs except that actions are represented by nodes and conditions by links. Several robots use them, including the PR2 robots developed by Willow Garage and the SaviOne robot developed by Savioke. There is a Python library, called SMACH,[5] that can be used to build hierarchical state machines.

Adaptive Cell Decomposition

Shakey used a grid model, such as the one shown in figure 13,[6] to map the obstacles in its environment. If a cell is not completely empty or completely full, it is divided into smaller cells and so on until one of these conditions is met for all cells. We believe Shakey's was the first use of an adaptive grid model. Adaptive cell decomposition is still used in robot navigation and in computer-aided design and manufacturing.

STRIPS Rules

STRIPS was the system Shakey used for generating plans to accomplish goals. Figure 14 shows a STRIPS rule for modeling the action of moving a toy block from C to B. The *preconditions* must be satisfied before the action can be applied, and the terms on the *delete list* can no longer be guaranteed to be satisfied after the action is applied, so they are deleted from Shakey's post-action model of the world. The terms on the *add list* are added to the post-action model.

STRIPS rules (or their derivatives) are used in most modern planners. (The STRIPS paper gets over 5000 citations on Google Scholar, and "STRIPS-style planning" gets over 3410 results on Google.) It's the rules that are used, not the STRIPS program itself. STRIPS rules were a practical solution to the "frame problem" — inherent in the use of the "situation calculus," proposed by McCarthy and Hayes (1969) for generating plans.

Hierarchical task networks (HTNs) are much used and powerful planning systems that use STRIPS rules.[7] These systems assemble plan steps into networks of actions, some of which can be executed in parallel and others that must be executed serially. We show an example in figure 15.

SIPE-2,[8] O-Plan (Currie and Tate 1991), and SHOP2[9] are examples of implemented HTNs. Among other important applications of HTNs, SIPE-2 has been used for production planning at an Australian Brewery.

Some video games make use of STRIPS rules and HTNs for planning the actions of nonolayer characters (NPCs).[10]

Heuristic Search and A*

A* is a heuristic search algorithm developed during the Shakey project for efficiently searching a graph of navigation waypoints. It uses an evaluation function to rank the nodes reached during search and continues the search below the best-ranked node. The evaluation function for a node, *n,* is the sum of the cost of the links traversed already on the way to *n* plus an estimate of the cost from *n* to a goal node.

There are lots and lots of descendants and variants of A*. Here is a list of just some of them: D*, Field D*, Theta*, Real-Time A*, Iterative Deepening A*, Life-Long Planning A*, Simplified Memory Bounded A*, and Generalized Adaptive A*. Richard Korf at UCLA

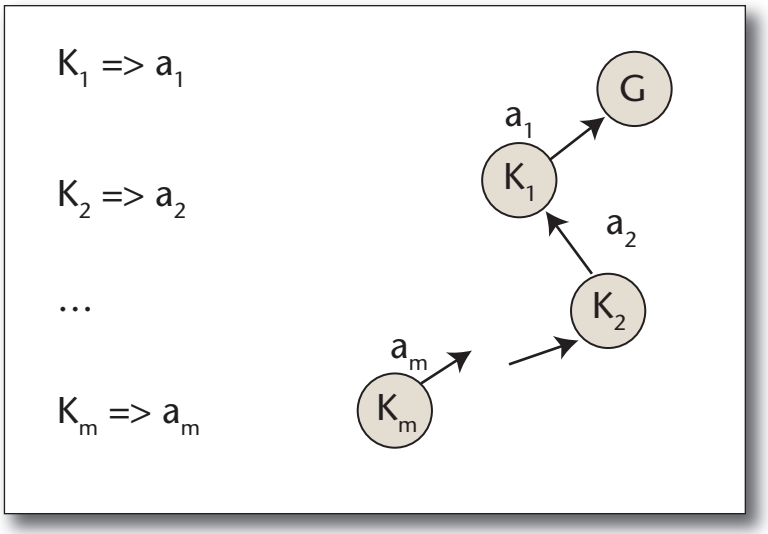


Figure 12. Teleo-Reactive Programs.

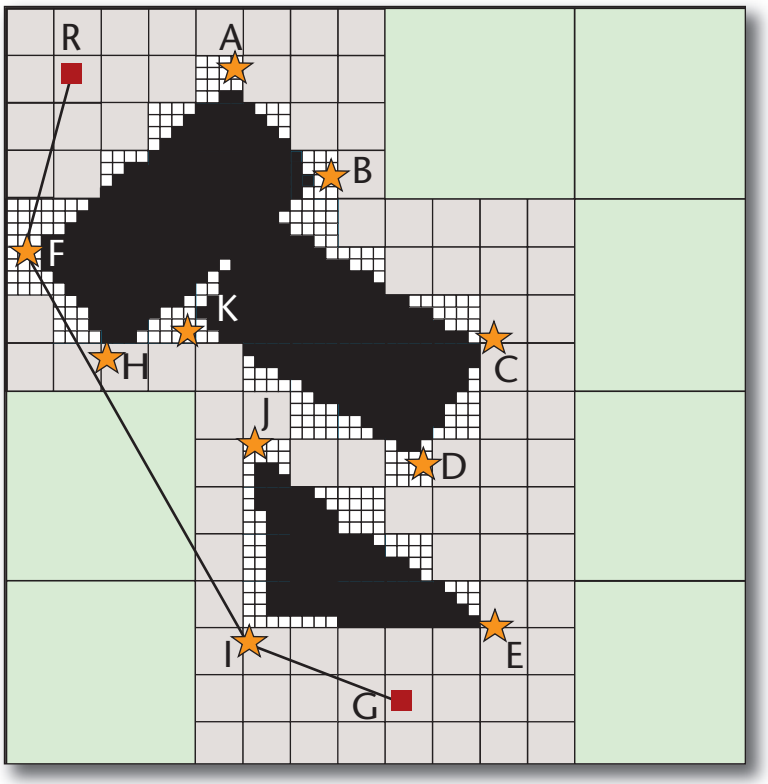


Figure 13. Shakey's Adaptive Grid Model.

and researchers at Carnegie-Mellon University have played major roles in the development of many of these.

The Mars rover, *Curiosity*, uses Field D*, a derivative of A* written by CMU's Tony Stentz and his student, Dave Ferguson (now with Google). It is capable of planning paths around obstacles in unknown, par-

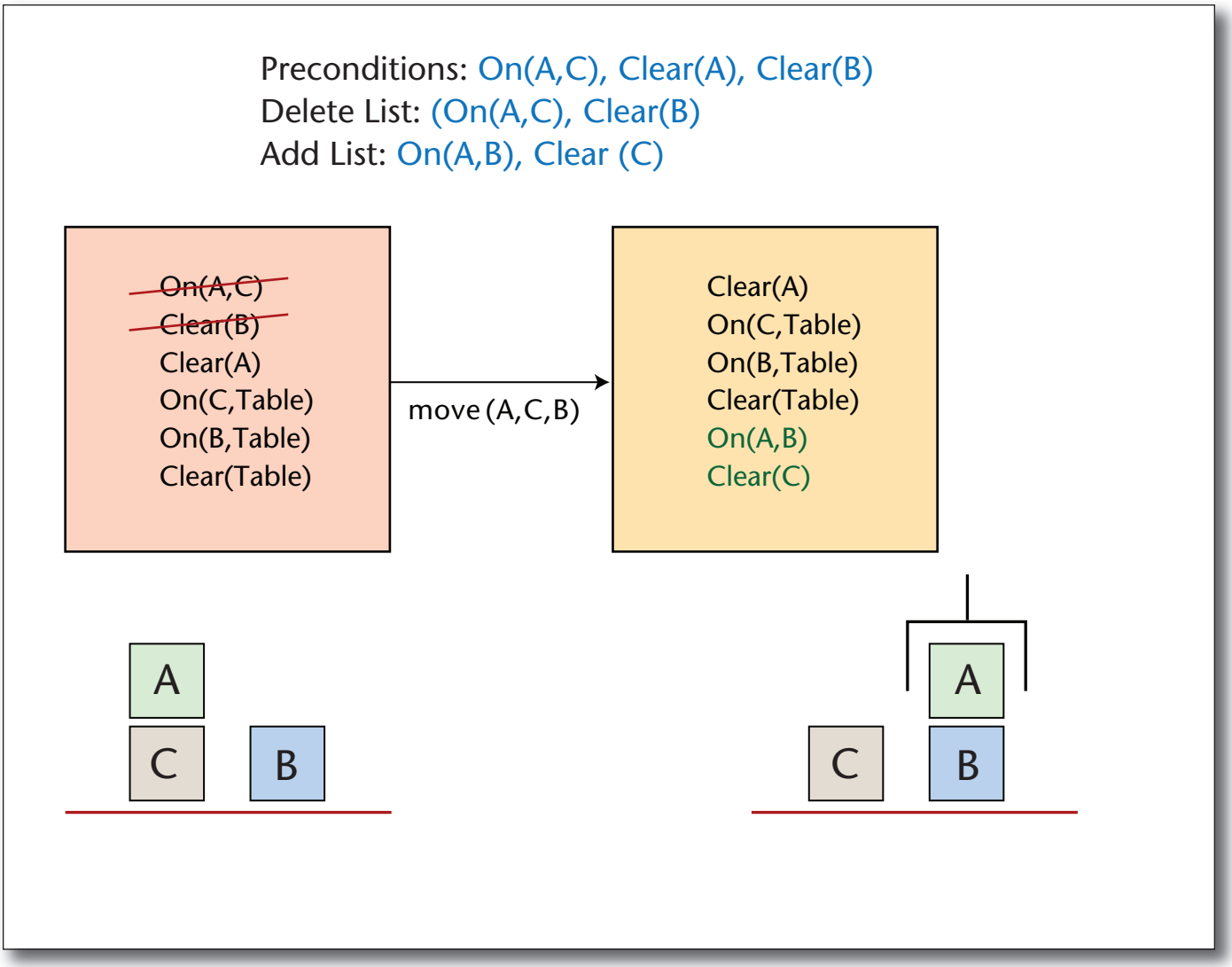


Figure 14. Application of a STRIPS Rule.

tially known, and changing environments in an efficient, optimal, and complete manner.

Most route-finding algorithms in maps use variants and elaborations of A*. Elaborations include the use of hierarchies, saved routes (which don't need to be recomputed), and much more.

In other uses of A*, linguists Dan Klein and Chris Manning write, "The use of A* search can dramatically reduce the time required to find a best parse ..." (Klein and Manning 2003). And Steven Woodcock, a computer games consultant, wrote that "A* is far and away the most used ... and most useful ... algorithm for [nonplayer-character] path finding in games today .... developers have noted that they make more use of A* than any other tool for pathfinding."[11] (Actually, we are gratified that the major applications of A* are on problems a bit more serious than video games!)

Computer Vision

As mentioned earlier, we had hoped that we could use then-existing computer vision routines to process images from Shakey's camera. But the state of computer vision was quite primitive at that time, so we did have to develop some routines of our own. One, which influenced subsequent vision systems, was a system for segmenting images into "like-appearing regions" (Brice and Fennema 1970). An example of the regions found for some of the objects in Shakey's environment is illustrated in figure 16. Segmentation is still a major technique used in computer vision today.[12]

Another result of work on computer vision during the Shakey project was the development of the "modern form" of the Hough transform for finding lines and curves in images. As mentioned earlier, Richard Duda and Peter Hart modified the original

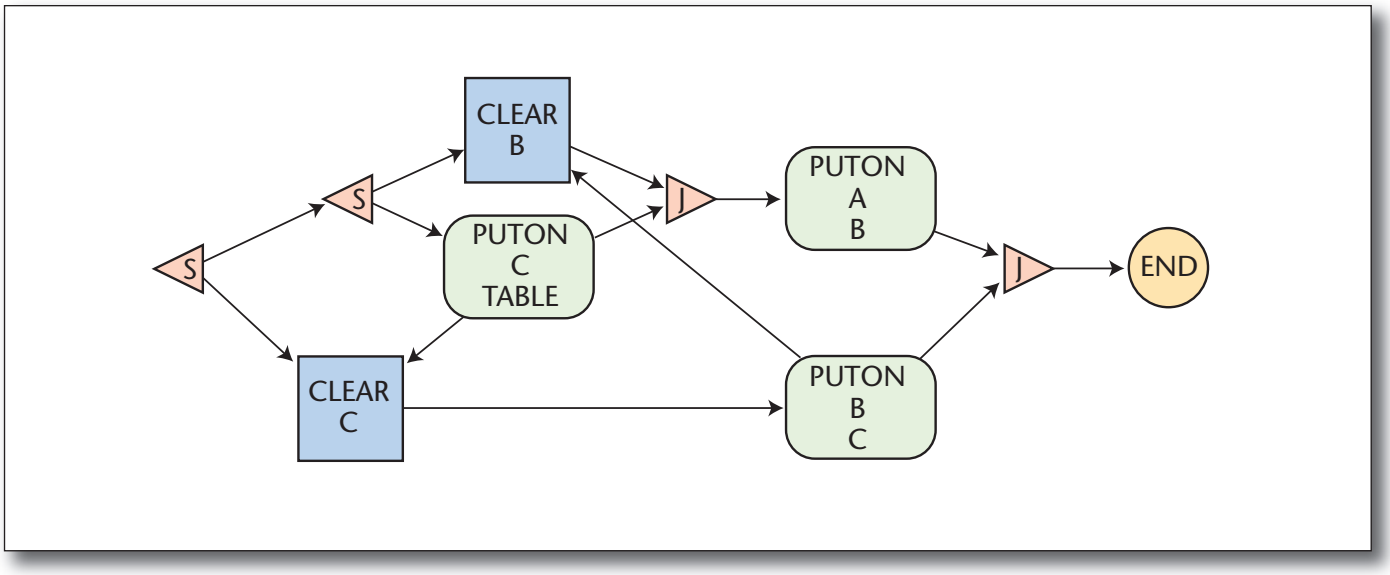


Figure 15. A Hierarchical Task Network.

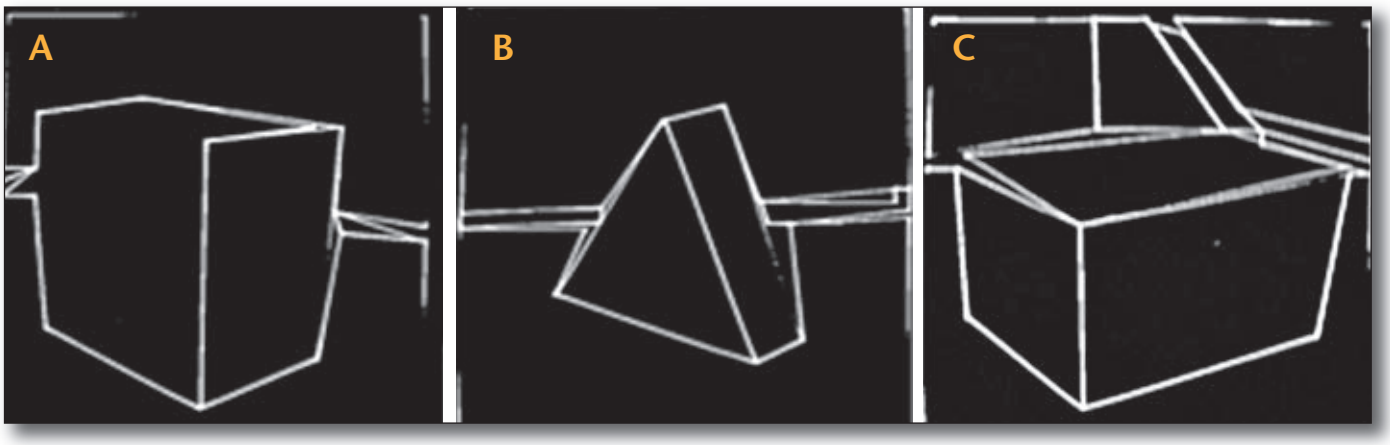


Figure 16. Region Finding as Used by Shakey's Vision System.

Reprinted with permission from Claude Brice and Claude Fennema, Scene Analysis Using Regions. *Artificial Intelligence* 1970 (3-4).

version of the Hough transform to include circles and analytic curves and to use a rho-theta parameterization (Duda and Hart 1972). Their paper gets 4500 hits on Google Scholar.)

The modern form of the Hough transform is used in automobiles to detect lane markings to warn the driver about drifting out of his or her lane.[13]

Conclusions

One reason for the success of the Shakey project and for its extensive legacy is that we were the first group to think that developing a robot that could perceive its environment and make and execute plans was a feasible idea. At the time, there was little existing software for us to use, so we had to invent what we needed. It turned out that the new inventions were ones that had broad applicability once people heard about them.

Another reason for our success is that we had a very talented team of AI researchers and software developers, along with people who could make the connections between software and hardware (figure 17). Some team members had a reunion at SRI International in November 2014.

There are still many problems in AI where talented researchers could be first. An idea mentioned by Nilsson during the panel is to develop an "action hierarchy" analogous to the deep learning hierarchies that are being used for vision and speech recognition.[14] Some of these are said to be rough models of the perceptual part of the neocortex. But the cortex also coordinates and plans actions, as illustrated in the diagram

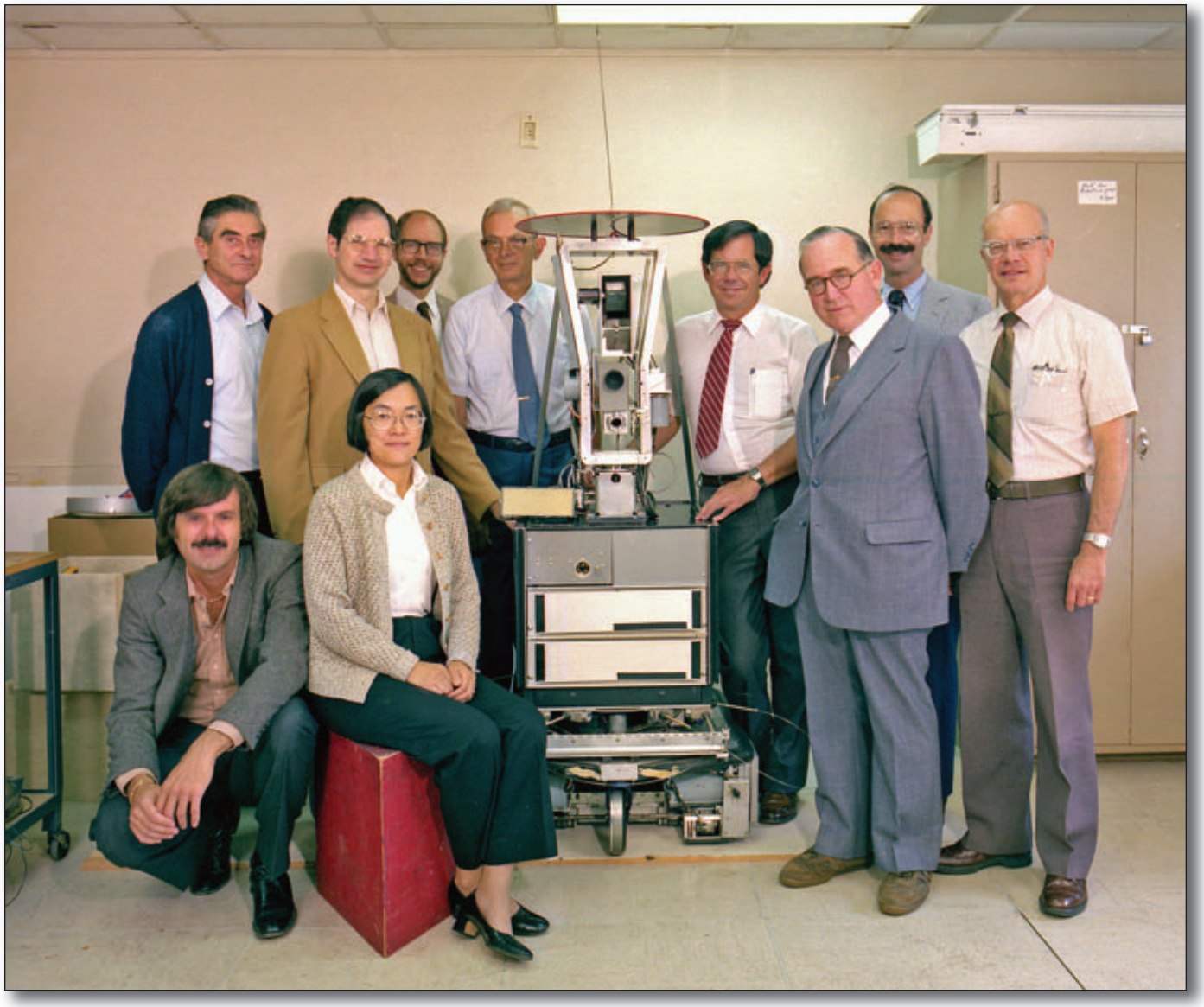


Figure 17. Some of the People Who Worked on Shakey

Front Row, left to right: Richard Fikes, Helen Chan Wolf. *Rear row, left to right:* Charles A. Rosen, Bertram Raphael, Richard O. Duda, Milt Adams, Jerry Gleason, Alfred E. (Ted) Brain, Peter E. Hart, and Jim Baer.

(figure 18).[15] How about developing a "deep action" system, with cross connections to the perceptual hierarchy and using (perhaps) hierarchical reinforcement learning to learn the actions?[16] One could then try to use both hierarchies to control a robot.

Notes

1. www.ai.sri.com/pubs/files/629.pdf.

2. ai.stanford.edu/~nilsson/Shakey.mp4.

3. LIFE, November 20, 1970.

4. For more information, see the T-R website teleoreactiveprograms.net.

5. See wiki.ros.org/smach.

6. The figure is from Nils J. Nilsson, "A Mobile Automaton: An Application of Artificial Intelligence Techniques," *Proceedings of the International Joint Conference on Artificial Intelligence*, 7–9 May, 1969. Washington, DC. Los Altos, CA: William Kaufmann Inc.

7. See en.wikipedia.org/wiki/hierarchical_task_network for more information.

8. See www.ai.sri.com/~sipe.

9. See www.cs.umd.edu/projects/shop.

10. See aigamedev.com/open/review/planning-in-games.

11. Email from Steven Woodcock sent to Nilsson on 6/14/2003.

12. See, for example, en.wikipedia.org/wiki/Image_segmentation.

13. For a video of the Hough Transform in action, see www.youtube.com/watch?v=DPApsnpPjuU/.

14. See, for example, www.cs.toronto.edu/~hinton/.

15. Diagram from willcov.com/bio-consciousness/sidebars/Perception—Action%20Cycle.htm.

16. The following paper seems relevant to this problem: Nicholas K. Jong and Peter

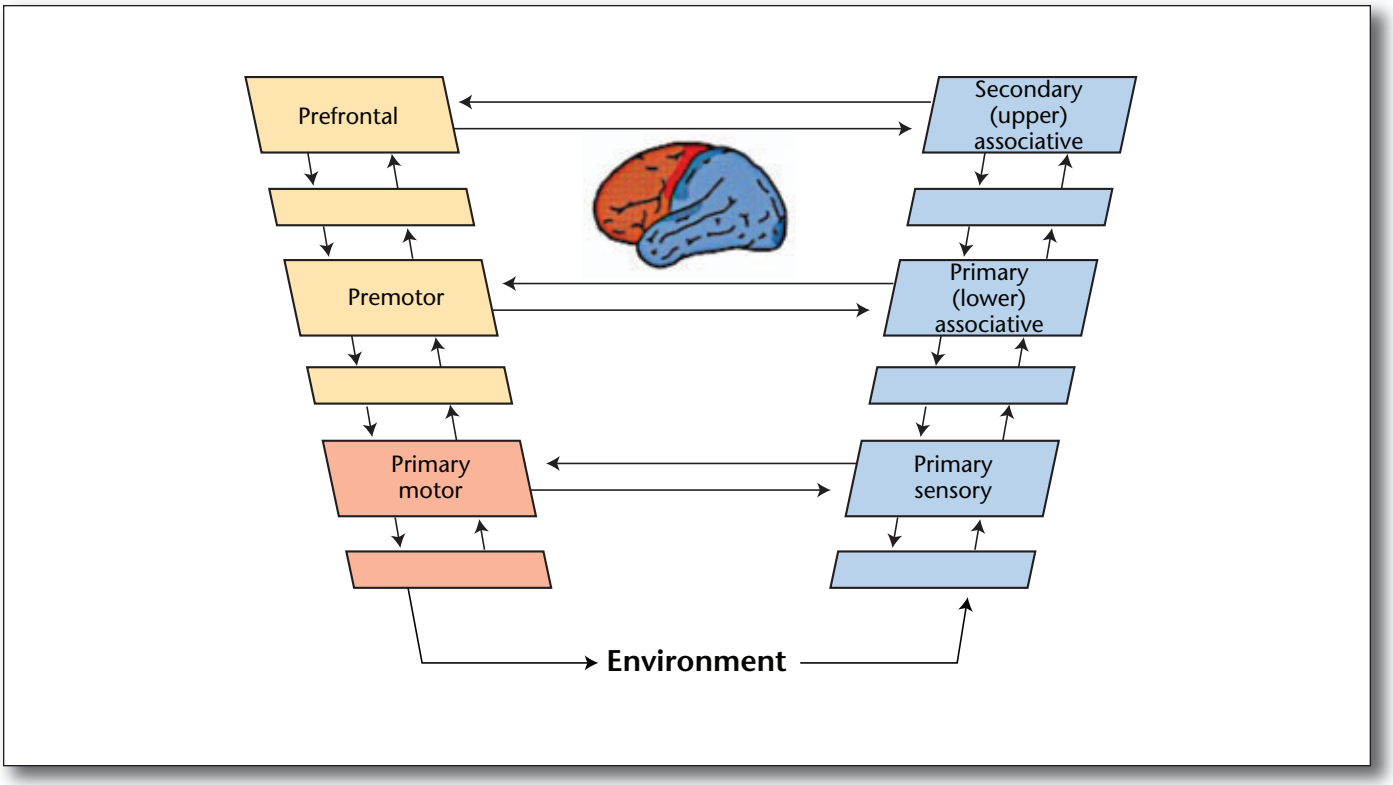


Figure 18. Cortex Motor and Perceptual Hierarchies.
Adapted from Joaquín Fuster, *The Prefrontal Cortex*, 360 (New York: Raven Press.)

Stone, Hierarchical Model-Based Reinforcement Learning: Rmax + MAXQ, in *Proceedings of the Twenty-Fifth International Conference on Machine Learning*, July 2008.

References

Bernard, D.; Dorais, G.; Gamble, E.; Kanefsky, B.; Kurien, J.; Man, G. K.; Millar, W.; Muscettola, N.; Nayak, P.; Rajan, K.; Rouquette, N.; Smith, B.; Taylor, W.; Tung, Y. W. 1999. Spacecraft Autonomy Flight Experience: The DS1 Remote Agent Experiment. In *Proceedings of the American Institute of Aeronautics and Astronautics*, 28–30. Reston, VA: American Institute of Aeronautics and Astronautics.

Brice, C. R., and Fennema, C. L. 1970. Scene Analysis Using Regions. *Artificial Intelligence* 1(3): 205–226.

Clark, K., and Robinson, P. 2016. *Programming Robotic Agents: A Multi-Tasking Teleo-Reactive Approach.* Berlin: Springer.

Currie, K., and Tate, A. 1991. O-Plan: The Open Planning Architecture. *Artificial Intelligence* 52(1): 49–86.

Duda, R. O., and Hart, P. E. 1972. Use of the Hough Transform to Detect Lines and Curves in Pictures. *Communications of the ACM* 12(1): 11–15.

Fikes, R. E., and Nilsson, N. J. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence* 2(3–4): 189–208.

Fikes, R. E.; Hart, P. E.; and Nilsson, N. J. 1972. Learning and Executing Generalized Robot Plans. *Artificial Intelligence* 3(4): 251–288.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths in Graphs, *IEEE Transactions on Systems Science and Cybernetics* SSC-4(2): 100–107.

Klein, D., and Manning, C. D. 2003. "A* Parsing: Fast Exact Viterbi Parse Selection. In *Proceedings of the Human Language Technology Conference and the North American Association for Computational Linguistics (HLT-NAACL),* 119–126. Stroudsberg, PA: Association for Computational Linguistics.

McCarthy, J., and Hayes, P. 1969. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In *Machine Intelligence 4,* ed. B. Meltzer and D. Michie. Edinburgh, Scotland: Edinburgh University Press.

McGann, C.; Py, F.; Rajan, K.; Thomas, H.; Henthorn, R.; McEwen, R. 2008. A Deliberative Architecture for AUV Control. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation.* Piscataway, N.J.: Institute for Electrical and Electronics Engineers.

Nilsson, N. J. 1994. Teleo-Reactive Programs for Agent Control. *Journal of Artificial Intelligence Research* 1: 139–158.

Benjamin Kuipers is a professor of computer science and engineering at the University of Michigan. He is a Fellow of AAAI.

Edward A. Feigenbaum is a professor of computer science emeritus at Stanford University, a joint winner of the 1994 ACM Turing Award, a Fellow of AAAI, and an AAAI former president.

Peter E. Hart was chair and president of Ricoh Innovations and an alumnus of SRI International. He is a Fellow of AAAI..

Nils Nilsson is the Kumagai Professor of Engineering (Emeritus) in the Department of Computer Science at Stanford University, a Fellow of AAAI, and an AAAI former president.

The Fifth International Competition on Knowledge Engineering for Planning and Scheduling: Summary and Trends

Lukáš Chrpa, Thomas L. McCluskey, Mauro Vallati, Tiago Vaquero

■ *We review the 2016 International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS), the fifth in a series of competitions started in 2005. The ICKEPS series focuses on promoting the importance of knowledge engineering methods and tools for automated planning and scheduling systems.*

The International Competition on Knowledge Engineering for Planning and Scheduling has been running since 2005 as a biennial event promoting the development and importance of the use of knowledge engineering methods and techniques within this area. The aim of the competition series is to foster developments in the knowledge-based and domain modeling aspects of automated planning, to accelerate knowledge engineering research, and to encourage the creation and sharing of prototype tools and software platforms that promise more rapid, accessible, and effective ways to construct reliable and efficient automated planning systems.

ICKEPS 2016 aimed specifically (1) to provide an interesting opportunity for researchers and students to experience the challenges of knowledge engineering; (2) to motivate the

  ISSN 0738-4602

planning community to create and improve tools and techniques for supporting the main design phases of a planning domain model; and (3) to provide new interesting and challenging models that can be used for testing the performance of state-of-the-art planning engines. In order to achieve the mentioned aims, ICKEPS 2016 focused on on-site modeling of challenging scenarios, performed by small teams.

This article summarizes the ICKEPS held in 2016. More information about the competition, including complete scenario descriptions, can be found on the ICKEPS 2016 website.[1]

Format and Participants

ICKEPS 2016 format included two main stages: On-site modeling and demonstration.

During the on-site modeling stage, each team received a set of scenarios description and had to exploit the available time for generating the corresponding models. Four scenarios were provided. Two of them — Star Trek, Rescue of Levaq, and Roundabout — required temporal constraints, while the other two — RPG and Match-Three, Harry! — only required classical reasoning. Participants were free to select the scenarios to tackle and had no restrictions on the number and type of tools that can be used. The only constraints were on the available time (six hours were given) and on the maximum size of teams (at most four members).

The day after the on-site modeling, each team had 10 minutes to present and demonstrate the aspects of the knowledge engineering process they exploited for encoding the scenarios. Specifically, teams were expected to discuss the division of work among team members, the tools used, key decisions taken during the encoding, and the issues they faced.

Teams were then ranked by a board of judges, which included Minh Do (NASA, USA), Simone Fratini (ESA, Germany), Ron Petrick (Heriot-Watt University, UK), Patricia Riddle (University of Auckland, New Zealand), and David Smith (NASA, USA). The evaluation process will be described in the corresponding section below. Noteworthy, judges were presented during the demonstrations session and had the opportunity to ask questions and discuss relevant aspects of the knowledge engineering process the teams followed.

The competition had two tracks: the PDDL track, where teams had to generate PDDL models using PDDL features up to those introduced in version 3.1, and the Open track, where teams could encode models in any other language. However, for the open track, participants were also required to provide a planner able to deal with the selected language. Sixteen people, divided into six teams, took part in the competition. One team entered the Open track, while the remaining five decided to participate in the PDDL track.

Participants came from institutions in Australia, Brazil, Canada, USA, Japan, and the United Kingdom. The level of expertise of participants covered various academic ranks, that is, Ph.D. students, lecturers, research fellows, and professors. One team was composed only of industry experts.

Evaluation

The board of judges evaluated each team by considering two main aspects: the exploited knowledge engineering process and the quality of the generated models.

The knowledge engineering process was assessed once for each team, regardless of the number of scenarios the team was able to encode. Three main criteria were taken into account: teamwork, method, and tools. Teamwork focused on the degree of cooperation and effective collaboration among team members. In terms of the method, effectiveness and systematicity of the knowledge engineering process were assessed. Finally, the innovation and originality of exploited tools, and their actual usefulness (that is, the support their use provided to the process) were evaluated.

To assess the quality of the generated models, the organizers provided the judges with the models the teams had submitted along with quantitative and qualitative information about these models. Qualitative information included evaluations about correctness, (that is, whether all the requirements were correctly handled); readability (how easy it was to read and understand the model); generality (if the domain model could be reused on different problem instances); and originality, where the use of innovative ways for modeling element or interactions was evaluated. Quantitative information included statistics on the number of types, number of predicates, number of operators, total number of lines, and the average (maximum) number of parameters, effects, and preconditions per operator. Moreover, in the PDDL track, the run time and quality of solutions generated by 10 well-known planners (5 classical and 5 temporal) were provided to judges. For teams participating in the Open track, the corresponding performance of the planner(s) submitted by the participants were provided to judges.

In accordance with the aims of the competition, emphasis was given to good practice in knowledge engineering, with particular regard to the degree of cooperation between the members of each team. For this reason the judges used a 0–100 scale, where up to 45 points could be awarded for the knowledge engineering process, and the remaining 55 points could be assigned according to the number and quality of generated models, as follows: Star Trek, the Rescue of Levaq (up to 20 points); Roundabout (up to 15 points); Match-Three, Harry! (up to 10 points); and RPG (up to 10 points).

Interesting in Hosting ICWSM-19?

AAAI, in cooperation with the ICWSM Steering Committee, is currently seeking proposals for a host city for the Thirteenth International AAAI Conference on Web and Social Media (ICWSM-19). The conference is typically held Monday – Thursday during the timeframe of mid-May through mid-June. Final selection of a site will be made by August 2017. For more information about proposal requirements, please write to icwsm19@aaai.org.

Note: ICWSM-18 will be held at Stanford University in Palo Alto, California USA.

Results

The board of judges acknowledged the efforts of all the competitors. Honorable mentions were then awarded in two categories:

The Innovative Methodology Award was presented to Emre Savas and Michael Cashmore. This team generated a complete domain transition graph for the RPG scenario by hand, analyzed the graph to remove bad states and transitions, and then created a compact and elegant model for the domain.

The Dilithium Crystal Award was presented to Sara Bernardini, Maria Fox, and Chiara Piacentini. This team was the only one to produce a working model that correctly captures most of the requirements of the Star Trek Scenario, which was the most difficult domain in the competition.

The Overall Winner Award was presented to the team composed of Nir Lipovetzky and Christian Muise. This team demonstrated a great ability to develop high-quality models quickly in multiple scenarios, while utilizing, and at the same time enhancing, model development tools for PDDL.

Given the positive feedback from competitors and judges, we believe that ICKEPS 2016 was a success. It is therefore envisaged that future ICKEPS will exploit a similar format.

Reflections

We observed that the generated models showed significant differences, even on easier scenarios, where, for instance, the number of operators ranged from two to seven, with remarkable impact on readability and generality. The impact on different planning approaches has to be assessed, in order to advance the state-of-the-art of knowledge engineering.

Two items were of concern at ICKEPS. First, most teams did not use any tools (except text editors), and thus relied only on their expertise. Second, existing tools do not effectively support cooperation: to cope with the growing complexity of planning applications, planning experts have to cooperate and coordinate the knowledge engineering process. In addition, the number of participants of ICKEPS is still not very large, especially when compared with the latest edition of the International Planning Competition (Vallati et al. 2015). This suggests that the planning community underestimates the importance of knowledge engineering, despite of its enormous impact on applicability of domain-independent planning in real-world scenarios.

Reference

Vallati, M.; Chrpa, L.; Grzes, M.; McCluskey, T. L.; Roberts, M.; and Sanner, S. 2015. The 2014 International Planning Competition: Progress and Trends. *AI Magazine* 36(3): 90–98.

Lukáš Chrpa is a research fellow at University of Huddersfield. His main research interests are in the area of AI planning, machine learning, and knowledge engineering. He was a co-organizer of the 2014 International Planning Competition (IPC).

Thomas L. McCluskey is a coleader of the PARK research group of the University of Huddersfield, UK. His research interests include automated plan generation, domain modeling, information extraction, knowledge engineering, machine learning, data mining, requirements validation, and formal specification. He cofounded the Knowledge Engineering for Planning and Scheduling competitions. He also co-organized IPC-2014.

Mauro Vallati is a senior lecturer at University of Huddersfield, UK. His main research interest is in AI planning. He was a co-organizer of the 2014 International Planning Competition.

Tiago Vaquero is a postdoctoral fellow at MIT CSAIL and Caltech. His research interests include autonomous systems, automated planning and scheduling, knowledge engineering, robotic space exploration, artificial intelligence, and robotics in general. He was a co-organizer of the 2012 edition of the International Competition on Knowledge Engineering for Planning and Scheduling.

The Evolution of Scheduling Applications and Tools

Mark Boddy

■ *The available tools and support for building planning and scheduling systems and applications have been steadily improving for decades. At the same time, the scope, scale, and complexity of the problems to be addressed have been increasing. In this column, I discuss several different scheduling applications developed over the past 25 years and then describe the tools and techniques used in addressing these problems, showing how improved tools simplified (and in some cases enabled) the solution of problems of increasing difficulty.*

According to folk definitions of planning and scheduling commonly used in the AI community, *planning* is deciding what to do, while *scheduling* is deciding when and how to do it. Neither of these terms are fundamental categories. *Scheduling applications* are simply those that can be addressed using scheduling tools and techniques. Using a few examples drawn from personal experience spanning more than two decades, in this column I provide one perspective on how those tools and techniques have evolved, as well as the resulting effect on the scope and scale of applications that can be addressed.

Between 1993 and 1995, a group of us at Honeywell implemented a constraint-based scheduler for the Airplane Information Management System (AIMS) that was developed for the Boeing 777. The initial AIMS scheduling problem encompassed 29,000 discrete activities, subject to 97,000 complex metric constraints specified by AIMS applications developers. Generating feasible schedules was an essential requirement for operating the 777, potentially threatening a Boeing investment of almost 10 billion dollars. The scale and complexity of this problem were unprecedented, and there were very few applicable tools or standards. Input requirements were provided as text, with a semantics negotiated and maintained through frequent discussion. As this was one of the earliest schedulers based on the simple temporal problem (STP), we implemented methods for incremental updates and bounds computation, as well as integrating the STP model with a large set of discrete decision variables. The solver used a locally implemented adaptation of Ginsberg's recently published Dynamic Backtracker, a systematic search algorithm combining stack reordering (not just conflict-directed backjumping) with what would later be called clause learning, as well as several customized constraint propagators (Boddy and Goldman 1994). Notable in this development effort was the extended process of negotiation with the AIMS developers as they sought to preserve functionality, repeatedly providing sets of requirements that we demonstrated to be unsatisfi-

able, using tools implemented specifically for that purpose.

Subsequently, tooling support for building planning and scheduling systems became more prevalent. For example, ilog Solver and Scheduler provided constraint modeling and solving capabilities, including specialized implementations of global constraints such as all-different. These tools used an extension of the Prolog goal stack, rendering them of limited utility where explicit control of the search process was required. Advances in understanding the relationship between propositional reasoning and integer programming widened the set of solvers available, while improvements in tools like CPLEX and a range of constraint-satisfaction problem (CSP) solvers including but not limited to iLog tools such as OPL further reduced the amount of implementation needed for a new application. Scheduling-specific ontologies and specialized constraints, custom control over constraint propagation, and support for backjumping during search made these systems increasingly useful. By 2010, these improvements in integration and scale enabled us (now at Adventium) to implement a system modeling processing and communication demands in large networks (10,000 to 1,000,000 nodes). Newly developed tools and standards played a key part: the domain and problem instances were represented in the Architecture Analysis and Design Language (AADL), from which we extracted a set of constraints in the high-level constraint language MiniZinc, translated from there into a linear program, which solved in single-digit minutes, using stock hardware (Michalowski, Boddy, and Carpenter 2010).

Then in 2013, very nearly 20 years after the AIMS scheduler, we constructed a prototype scheduler for a modern avionics system for a large commercial jetliner. The problem was larger, more complex, and significantly more diverse than that addressed by the AIMS scheduler. Instead of a single integrated network with external sensor interfaces, there were multiple networks with gateways between them. Instead of one communication protocol (and thus one scheduling model) there were several. Instead of functions preassigned to processors, that assignment was part of the problem. Multiple, gatewayed networks with different communication protocols were a particular issue, requiring enforcement of timing guarantees across multiple asynchronous boundaries. Fortunately, the available tools were much improved as well, to the point where the application could be assembled largely by integrating existing tools. Instead of hand-rolled domain models, we used AADL. Instead of manual integration of discrete and continuous parts of the problem as had previously been required, we used satisfiability modulo theories (SMTs). Instead of hand-implemented search control, we used an off-the-shelf solver. The largest remaining implementation task was to translate from the AADL model to a formulation suitable for the SMT solver.[1] This tool remains in a prototype form, but the models and integration methods have been used for other applications.

Most recently, we have integrated a scheduling engine into design tools for integrated, heterogeneous embedded software systems. These tools have been evaluated by system developers, and are now being provided as Government Furnished Equipment for the preliminary design phase of a major aircraft development program.[2] In our view, this kind of integration is where the main challenge of the future lies. Public infrastructure is increasingly distributed and integrated, while embedded systems grow increasingly complex and interdependent. Individual components of these large systems may have very different dynamics and may be provided by vendors desiring to protect proprietary information. Addressing these problems requires integrating diverse tools, sharing limited information, with a rigorous semantic mapping among them.

Over the past 20 years, there has been significant progress in the tools available to support all aspects of defining and implementing constraint-based scheduling applications. This process is ongoing; examples of current research technologies with promise for real applications include various refinements of Monte-Carlo tree search, and the synthesis of specific algorithms for problem instances, as for example in ongoing work by Doug Smith at Kestrel. At the same time, the increasing complexity and integration of computing systems continues to provide more, larger, and more complex problems to solve. This class of applications should provide a fruitful source of new modeling and solution challenges for years to come.

Notes

1. See nari.arc.nasa.gov/sites/default/files/Boddy SPICA PhaseI FinalReport r2.pdf

2. www.adventiumlabs.com/our-work/products-services/model-based-engineering-mbe-tools

References

Boddy, M., and Goldman, R. 1994. Empirical results on scheduling and dynamic backtracking. Paper presented at the Third International Symposium on Artificial Intelligence, Robotics, and Automation for Space, October 18–20, Pasadena, California USA.

Michalowski, M.; Boddy, M.; and Carpenter, T. 2010. Coordinated Management of Large-Scale Networks Using Constraint Satisfaction. Paper presented at the 2010 AAAI Workshop on Intelligent Security (Security and Artificial Intelligence), 12 July, Atlanta Georgia, USA.

Mark Boddy is co-owner and chief scientist at Adventium Labs, a small, for-profit research lab located in Minneapolis, MN. His areas of research include planning and scheduling, automated reasoning, and constraint-based reasoning. Over the past 30 years, most of his work has been on adapting and extending techniques from a broad variety of research areas for application to a range of problems, primarily in planning and scheduling domains. This work has drawn from and in some cases contributed to research in constraint satisfaction, heuristic search, mathematical optimization, classical planning and extensions thereto, temporal reasoning, multiagent cooperative negotiation, resource-bounded reasoning, and a number of other areas. He coauthored the papers that coined the widely used terms *anytime algorithm, performance profile,* and *conformant planning.*

RuleML (Web Rule Symposium) 2016 Report

Paul Foder, Guido Governatori,
José Júlio Alfers, Leopoldo Bertossi

■ *This article reports on the 10th International Web Rule Symposium, which was held at Stony Brook University in Stony Brook, New York, from July 6–9, 2016.*

The 10th International Web Rule Symposium (RuleML 2016)[1] was held at Stony Brook University, Stony Brook, New York, from July 6–9, 2016. A total number of 68 papers were submitted from which 18 full papers, 2 short papers, 3 industry papers, 7 challenge papers, and 3 Doctoral Consortium papers were selected. Moreover, 2 keynote and 2 tutorial papers were invited. Most regular papers were presented in one of these tracks: Smart Contracts, Blockchain, and Rules, Constraint Handling Rules, Event Driven Architectures and Active Database Systems, Legal Rules and Reasoning, Rule- and Ontology-Based Data Access and Transformation, Rule Induction, and Learning.

Following up on previous years, RuleML also hosted the 6th RuleML Doctoral Consortium and the 10th International Rule Challenge, which this year was dedicated to applications of rule-based reasoning, such as Rules in Retail, Rules in Tourism, Rules in Transportation, Rules in Geography, Rules in Location-Based Search, Rules in Insurance Regulation, Rules in Medicine, and Rules in Ecosystem Research.

This year's symposium featured four invited keynote and tutorial talks. Keynote speaker Richard Waldinger of SRI International presented Natural Language Access to Data: It Needs Reasoning. Keynote speaker Bruce Silver of Bruce Silver Associates spoke on DMN as a Decision Modeling Language. Tutorial speaker Neng-Fa Zhou, of City University of New York, presented Programming in Picat, while tutorial speakers Michael Kifer, Theresa Swift, and Benjamin Grosof of Coherent Knowledge Systems presented Practical Knowledge Representation and Reasoning in Ergo.

As a novelty this year, there was a highly successful colocation between RuleML 2016 and DecisionCAMP 2016, facilitated by Jacob Feldman and colleagues. A total number of 132 participants attended both conferences and the affiliated subevents. The colocation was a great opportunity for the rule-based community and the industrial decision-modeling community to mingle at one of the several joint events, including the joint reception on Tuesday July 6 and the Thursday July 8 conference dinner at the Hilton Garden Hotel. Other joint events included the joint keynote by Bruce Silver; the joint tutorial by Neng-Fa Zhou; and the RuleML industry session on Friday, July 9.

The RuleML 2016 Best Paper Awards were presented to Iliano Cervesato, Edmund Soon Lee Lam, and Ali Elgazar for their paper Choreographic Compilation of Decentralized Comprehension Patterns and ro Ho-Pun Lam, Mustafa Hashmi, and Brendan Scofield for their paper Enabling Reasoning with LegalRuleML.

The 10th International Rule Challenge Awards went to Ingmar Dasseville, Laurent Janssens, Gerda Janssens, Jan Vanthienen, and Marc Denecker, for their paper Combining DMN and the Knowledge Base Paradigm for Flexible Decision Enactment, and Jacob Feldman for his paper What-If Analyzer for DMN-based Decision Models.

As in previous years, RuleML 2016 was also a place for presentations and face-to-face meetings about rule technology standardizations, which this year covered RuleML 1.02 (System of Families of Languages and Knowledge-Interoperation Hub) and DMN 1.1 (OMG DMN RTF).

Details about the RuleML and DecisionCAMP 2016 programs can be found at the Rule ML website.[1,2,3] The proceedings are available from Springer. Workshop proceedings have been uploaded to the CEUR proceedings site. The RuleML videos are available from Youtube.[4]

We would like to thank our sponsors, whose contributions allowed us to cover the costs of student participants and invited/keynote speakers. We would also like to thank all the people who have contributed to the success of this year's special RuleML 2016 and colocated events, including the organization chairs, PC members, authors, speakers, and participants.

The RuleML community will join forces with the RR (Web Reasoning and Rule Systems) community for a joint conference in 2017, which will be held in London, UK. RuleML+RR 2017: International Joint Conference on Rules and Reasoning is being organized under the leadership of Fariba Sadri and Roman Kontchakov.

.

Notes

1. 2016.ruleml.org.

2. link.springer.com/book/10.1007/978-3-319-42019-6.

3. ceur-ws.org/Vol-1620.

4 . www.youtube.com/playlist?list=PLQkz10de_pG8xIi3t_aT0JHzrgkQhlOba.

Paul Foder was general chair of RuleML. He is a research assistant professor at Stony Brook University in New York.

Guido Governatori was a program cochair of Rule ML. He is a senior principal researcher and leads the research activities on business processes and legal informatics at data61 and the Commonwealth Scientific and Industrial Research Organisation in Australia.

José Júlio Alfers served as a program cochair of Rule ML. He is a professor at Universidade Nova de Lisboa, Portugal.

Leopoldo Bertossi served as program cochair of RuleML. He is a a professor at Carleton University, Canada.

AAAI News

Spring News from the Association for the Advancement of Artificial Intelligence

AAAI Announces New Senior Members!

AAAI congratulates the following individuals on their election to AAAI Senior Member status:

Alessandro Cimatti (Fondazione Bruno Kessler, Italy)

Xuelong Li (Chinese Academy of Sciences, China)

Nathan R. Sturtevant (University of Denver, USA)

This honor was announced at the recent AAAI-17 Conference in San Francisco. Senior Member status is designed to recognize AAAI members who have achieved significant accomplishments within the field of artificial intelligence. To be eligible for nomination for Senior Member, candidates must be consecutive members of AAAI for at least five years and have been active in the professional arena for at least ten years.

Congratulations to the 2017 AAAI Award Winners!

Rao Kambhampati, AAAI President, Tom Dietterich, AAAI Past President and Awards Committee Chair, and Yolanda Gil, AAAI President-Elect, presented the AAAI Awards in February at AAAI-17 in San Francisco.

Classic Paper Award

The 2017 AAAI Classic Paper Award was given to the authors of the paper deemed most influential from the Sixteenth National Conference on Artificial Intelligence, held in 1999 in Orlando, Florida, USA. The 2017 recipient of the AAAI Classic Paper Award was:

Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. *Dieter Fox, Wolfram Burgard, Frank Dellaert, Sebastian Thrun*

Fox and his coauthors were honored for pioneering the application of particle filtering to provide an effective and scalable method for robot localization. Dieter Fox presented an invited talk during the conference in recognition of this honor.

Dieter Fox is a professor in the Department of Computer Science and Engineering at the University of Washington. He grew up in Bonn, Germany, and received his Ph.D. in 1998 from the Computer Science Department at the University of Bonn. He joined the UW faculty in the fall of 2000. His research interests are in robotics and artificial intelligence, with a focus on state estimation and perception. He heads the UW Robotics and State Estimation Lab (RSE-Lab). Fox is a Fellow of the AAAI and IEEE, and he received several best paper awards at major robotics, AI, and computer vision conferences.

A Classic Paper Honorable Mention was also given to the following two papers:

Combining Collaborative Filtering with Personal Agents for Better Recommendations. *Nathaniel Good, J. Ben Schafer, Joseph A. Konstan, Al Borchers, Badrul Sarwar, Jon Herlocker, John Riedl*

Good and his coauthors were honored for developing an effective way to combine collaborative filtering and content filtering to provide better recommendations to users.

Dr. Nathan Good is Principal of Good Research and Lecturer at UC Berkeley's Ischool's Master of Data Science program. A fundamental goal of his work is create devices, models services and user experiences that are sim-

2017 Feigenbaum Prize Awarded!

AAAI is delighted to announce that Yoav Shoham (Stanford University, Google), has been selected as the recipient of the 2017 AAAI Feigenbaum Prize. Shoham is being recognized in particular for high-impact basic research in artificial intelligence — including knowledge representation, multiagent systems, and computational game theory — and translating the basic research into impactful and innovative commercial products.

The AAAI Feigenbaum Prize was established in 2010 and is awarded biennially to recognize and encourage outstanding Artificial Intelligence research advances that are made by using experimental methods of computer science. The associated cash prize of $10,000 is provided by the Feigenbaum Nii Foundation.

AAAI Distinguished Service Award

The 2017 AAAI Distinguished Service Award recognizes one individual for extraordinary service to the AI community. The AAAI Awards Committee is pleased to announce that this year's recipient is James A. Hendler (Rensselaer Polytechnic Institute, USA). Hendler is being recognized for his outstanding contributions to the field of artificial intelligence through sustained service to AAAI, other professional societies, and government activities promoting the importance of Artificial Intelligence research.

Jim Hendler is the Director of the Institute for Data Exploration and Applications and the Tetherless World Professor of Computer, Web and Cognitive Sciences at RPI. One of the originators of the "semantic web," Hendler was the recipient of a 1995 Fulbright Foundation Fellowship, is a former member of the US Air Force Science Advisory Board, and is a Fellow of the AAAI, BCS, the IEEE, the AAAS and the ACM. He is also the former chief scientist of the Information Systems Office at the US Defense Advanced Research Projects Agency (DARPA) and was awarded a US Air Force Exceptional Civilian Service Medal in 2002. Hendler is also the first computer scientist to serve on the Board of Reviewing editors for Science. In 2010, Hendler was named one of the 20 most innovative professors in America by Playboy magazine and was selected as an "Internet Web Expert" by the US government. In 2012, he was one of the inaugural recipients of the Strata Conference "Big Data" awards for his work on large-scale open government data, and he is a columnist and associate editor of the Big Data journal. In 2013, he was appointed as the Open Data Advisor to New York State and in 2015 appointed a member of the US Homeland Security Science and Technology Advisory Committee and in 2016, became a member of the National Academies Board on Research Data and Information.

ple, secure and respectful of people's privacy. On the commercial side, Dr. Good has codeveloped technologies and designs for privacy protection products that have grown to millions of users, and has worked with fortune 100 firms to develop privacy and security solutions. He is a coauthor of the UC Berkeley web privacy census, and contributing author to books on privacy and the user experience of security systems. He has published extensively on user experience studies, privacy, and security related topics and holds patents on software technology for multimedia systems and event analysis. Prior to Good Research, Nathan was at PARC, Yahoo and HP research labs. At Berkeley, he worked with TRUST and was a member of the 2007 California SOS Top-to-Bottom Review of Electronic Voting Systems. His research has been reported on in the *Economist, New York Times,* CNN and ABC and he has testified on his research before Congress and the FTC.

Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. *Ellen Riloff and Rosie Jones*

Riloff and Jones were honored for introducing a mutual bootstrapping technique for information extraction that simultaneously learns the semantic lexicon and the extraction patterns.

Ellen Riloff is a professor in the School of Computing at the University of Utah. She received her Ph.D. in computer science from the University of Massachusetts in 1994. Her primary research area is natural language processing, with an emphasis on information extraction, sentiment analysis, semantic class induction, and bootstrapping methods that learn from unannotated texts. She has served on the NAACL Executive Board, Human Language Technology (HLT) Advisory Board, Computational Linguistics Editorial Board, Transactions of the Association for Computational Linguistics program cochair for the NAACL HLT 2012 and CoNLL 2004 conferences.

For more information about nominations for AAAI 2018 Awards, please contact Carol Hamilton at hamilton@aaai.org.

AAAI-17 Program Committee Awards

AAAI-17 Program Cochairs Shaul Markovitch and Satinder Singh recognized the following members of the AAAI-17 Program Committee for their distinguished service on the committee. These individuals went above and beyond the expectations for the role, showing exceptional judgment, clarity, knowledgeability, and leadership in reaching a consensus decision while serving on the committee.

Outstanding Senior Program Committee Members

Thomas Eiter (Vienna University of Technology, Austria)

Jussi Rintanen (Aalto University, Finland)

Sven Koenig (University of Southern California, USA)

Outstanding Program Committee Members

Luis Ortiz (University of Michigan - Dearborn, USA)

Aris Filos-Ratsikas (University of Oxford, UK)

Ingo Pill (Graz University of Technology, Austria)

Miquel Ramírez (University of Melbourne, Australia)

Christer Bäckström (Linköping University, Sweden)

Richard Valenzano (University of Toronto, Canada)

AAAI-17 Outstanding Paper Awards

This year, AAAI's Conference on Artificial Intelligence honored the following two papers, which exemplify high standards in technical contribution and exposition by regular and student authors.

AAAI-17 Outstanding Paper Award

Label-Free Supervision of Neural Networks with Physics and Domain Knowledge. *Russell Stewart and Stefano Ermon*

AAAI-17 Outstanding Student Paper Award

The Option-Critic Architecture. *Pierre-Luc Bacon, Jean Harb and Doina Precup*

2017 Innovative Application Awards

Each year the AAAI Conference on Innovative Applications selects the recipients of the IAAI Innovative Application Award. These deployed application case study papers must describe deployed applications with measurable benefits that include some aspect of AI technology. The application needs to have been in production use by its final end-users for sufficiently long so that the experience in use can be meaningfully collected and reported. The 2017 winners are as follows:

Large-Scale Occupational Skills Normalization for Online Recruitment. *Faizan Javed, Phuong Hoang, Thomas Mahoney, Matt McNair*

Phase-Mapper: An AI Platform to Accelerate High Throughput Materials Discovery. *Yexiang Xue, Junwen Bai, Ronan Le Bras, Brendan Rappazzo, Richard Bernstein, Johan Bjorck, Liane Longpre, Santosh K. Suram, Robert B. van Dover, John Gregoire, Carla P. Gomes*

Special Computing Community Consortium (CCC) Blue Sky Awards

AAAI-17, in cooperation with the CRA Computing Community Consortium (CCC), honored three papers in the Senior Member track that presented ideas and visions that can stimulate

Congratulations to the 2017 AAAI Fellows!

Each year a small number of fellows are recognized for their unusual distinction in the profession and for their sustained contributions to the field for a decade or more. An official dinner and ceremony were held in their honor during AAAI-17 in San Francisco, California.

Ronen I. Brafman (Ben-Gurion University, Israel)

For significant contributions to algorithms, representation, and theoretical foundations of automated decision making in the areas of preference handling, planning under uncertainty, multiagent planning, and privacy.

Eduard H. Hovy (Carnegie Mellon University, USA)

For significant contributions to natural language processing, including text summarization, semantic analysis, entity/event coreference and sentiment analysis.

Tommi S. Jaakkola (Massachusetts Institute of Technology, USA)

For significant contributions to the fields of machine learning, computational biology and natural language processing.

Maurizio Lenzerini (Università degli Studi di Roma "La Sapienza," Italy)

For fundamental contributions to knowledge representation, description logics, ontologies, and AI and databases, which have become mainstream in AI.

Fangzhen Lin (Hong Kong University of Science and Technology, Hong Kong)

For significant contributions to formal theories of knowledge representation, in particular nonmonotonic logics, answer-set programing, and theories of action.

Dale Eric Schuurmans (University of Alberta, Canada)

For significant contributions to machine learning, including foundational methods for model selection, on-line learning, unsupervised learning and sequential decision making.

Munindar P. Singh (North Carolina State University, USA)

For significant contributions to multiagent systems, especially via seminal formalizations of the interactions, communications, trust, and commitments among intelligent agents and services.

2017 Robert S. Engelmore Memorial Lecture Award

This award was established in 2003 to honor Dr. Robert S. Engelmore's extraordinary service to AAAI, AI Magazine, and the AI applications community, and his contributions to applied AI. The annual keynote lecture is presented at the Innovative Applications of Artificial Intelligence Conference. Topics encompass Bob's wide interests in AI, and each lecture is linked to a subsequent article published upon approval by AI Magazine. The lecturer and, therefore, the author for the magazine article, are chosen jointly by the IAAI Program Committee and the Editor of the *AI Magazine*.

AAAI congratulates the 2017 recipient of this award, David W. Aha, Naval Research Laboratory, who was honored for pioneering research contributions and high-impact applications in autonomous systems, machine learning, and case-based reasoning, and for extensive contributions to AAAI, including educating the broader AI community through AAAI doctoral consortia and video competitions. Aha presented his award lecture, "Goal Reasoning: Emerging Applications, a Foundation, and Prospects," at the Innovative Applications of Artificial Intelligence Conference in San Francisco.

David W. Aha (UCI 1990) is a member of NRL's Navy Center for Applied Research on AI. His group conducts basic and applied research on intelligent agents, ML, case-based reasoning, and related topics; their current projects concern goal reasoning or deep learning. He has mentored 12 postdocs, served on 20 PhD committees, was a AAAI Councilor, cocreated the AAAI AI Video Competition, and created the UCI Repository for ML Databases. Aha has co-organized mor than 30 events (including AAAI-17 DC, ICCBR-17, IJCAI-17 Workshop on XAI), serves on several PCs, and led or leads the evaluation team for four DARPA or ONR Programs.

the research community to pursue new directions, such as new problems, new application domains, or new methodologies. The recipients of the 2017 Blue Sky Idea travel awards, sponsored by the CCC, were the following:

The AI Rebellion: Changing the Narrative. *David W. Aha, Alexandra Coman*

Moral Decision Making Frameworks for Artificial Intelligence. V*incent Conitzer, Walter Sinnott-Armstrong, Jana Schaich Borg, Yuan Deng, Max Kramer*

Getting More Out of the Exposed Structure in Constraint Programming Models of Combinatorial Problems. *Gilles Pesant*

AAAI-17 Best Technical Demonstration Award

Two technical demonstrations were honored as part of the AAAI-17 Technical Demonstration Program. Votes for these awards were cast by all AAAI-17 technical registrants. The winners were as follows:

Best Technical Demonstration Award

Arnold: An Autonomous Agent to play FPS Games. *Devendra Singh Chaplot, Guillaume Lample*

Honorable Mention: Technical Demonstration

Deep Music: Towards Musical Dialogue. *Mason Bretan, Sageev Oore, Jesse Engel, Douglas Eck, Larry Heck*

AAAI-17 Student Abstract Awards

Two awards were presented to participants in the AAAI-17 Student Abstract Program, including the Best Student 3-Minute Presentation and the Best Student Poster. Nineteen finalists in the Best Student 3-Minute Presentation category presented one-minute oral spotlight presentations during the second day of the technical conference, followed that evening by their poster presentations. Votes for the Best Student 3-Minute Presentation award were cast by senior program committee members and students, and all AAAI-17 technical registrants were eligible to cast their votes for the Best Student Poster Award. The winners were as follows:

Best Student 3-Minute Presentation

Fast Electrical Demand Optimization under Real-Time Pricing. *Shan He, Mark Wallace, Campbell Wilson, Ariel Liebman*

Honorable Mention: Student 3-Minute Presentation

Robust and Efficient Transfer Learning with Hidden Parameter Markov Decision Processes. *Taylor W. Killian, George Konidaris, Finale Doshi-Velez*

Best Student Poster

Evolutionary Machine Learning for *RTS Game StarCraft. Lianlong Wu, Andrew Markham*

Honorable Mention: Student Poster

An Advising Framework for Multiagent Reinforcement Learning Systems. *Felipe Leno da Silva, Ruben Glatt, Anna Helena Reali Costa*

2017 AI Video Competition Winners

The eleventh annual AI Video Competition was held during AAAI-17 and several winning videos were honored during the awards presentation. Videos were nominated for awards in four categories, and winners will receive a "Shakey" award. Our thanks go to Charles Isbell and Scott Niekum for all their work on this event.

The winners of the four awards were as follows:

Best Video

HEALER: Using AI to Raise HIV Awareness among Homeless Youth. *Amulya Yadav, Eric Rice, Robin Petering, Jaih Craddock, Bryan Wilder, Milind Tambe*

Best Robot Video

Aquatic Micro Air Vehicles. *Robert Siddall, Alejandro Ortega, Raphael Zufferey,*

Talib Al-Hinai, Mirko Kovac

Best Student Video

HEALER: Using AI to Raise HIV Awareness among Homeless Youth. *Amulya Yadav, Eric Rice, Robin Petering, Jaih Craddock, Bryan Wilder, Milind Tambe*

Most Entertaining Video

How to Cut Cake Fairly. *Jack Fisher, Marcus Strom*

The organizers would like to thank these and all the authors of the nominated videos for their participation in the 2017 AI Video Competition. Other nominated videos included the following:

Top Drone. *Philip Moore, Michael Floyd, Justin Karneeb, David Aha*

Using Constraint Programming on a Single-Arm ABB Robot. *Mathieu Collet, Arnaud Gotlieb, Inger Karoline*

Active Video Summarization: Customized Summaries via On-line Interaction with the User. *Ana Garcia del Molino, Xavier Boix, Joaquim Bellmunt*

Congratulations to all!

ICWSM-17 Registration Now Available!

The Eleventh International AAAI Conference on Web and Social Media will be held at the Hyatt Regency Montreal in Montreal, Canada from May 15–18, 2017. This interdisciplinary conference is a forum for researchers in computer science and social science to come together to share knowledge, discuss ideas, exchange information, and learn about cutting-edge research in diverse fields with the common theme of online social media. This overall theme includes research in new perspectives in social theories, as well as computational algorithms for analyzing social media. ICWSM is a singularly fitting venue for research that blends social science and computational approaches to answer important and challenging questions about human social behavior through social media while advancing computational tools for vast and unstructured data.

ICWSM-17 will include a lively program of technical talks and posters, invited presentations, and keynote talks by Laurie Faith Cranor (Carnegie Mellon University), Hilary Mason (New York University), and Matthew J. Salganik (Princeton University). The ICWSM Workshop program will continue in 2017 with 4 half-day and four full-day workshops, and the Tutorial Program, comprising 2 half-day and 2 full-day tutorials, will run in parallel. Both will be held on the first day of the conference, May 15. For complete details about these programs, please see icwsm.org/2017/.

Registration information is available at the ICWSM-17 website (www.icwsm.org/2017/attending/registration). The early registration deadline is March 31, and the late registration deadline is April 21. For full details about the conference program, please visit the ICWSM-17 website (icwsm.org) or write to icwsm17@aaai.org.

AAAI/EAAI 2017 Outstanding Educator Award

The AAAI/EAAI Outstanding Educator was established in 2016 to recognize a person (or group of people) who has (have) made major contributions to AI education that provide long-lasting benefits to the AI community. Examples might include innovating teaching methods, providing service to the AI education community, generating pedagogical resources, designing curricula, and educating students outside of higher education venues (or the general public) about AI. AAAI is pleased to announce the 2017 award is being given to Sebastian Thrun (Udacity, KittyHawk, Stanford University, Georgia Tech) for his pioneering efforts on the creation of high-quality, widely available, and affordable online courses, including seminal artificial intelligence courses, and for demonstrating the excitement of AI research in self-driving cars and navigation. This award is jointly sponsored by AAAI and the Symposium on Educational Advances in Artificial Intelligence.

Sebastian Thrun pursues research on robotics, artificial intelligence, education, and human computer interaction. He founded Google's self driving car team, after winning the DARPA Grand Challenge. Together with Peter Norvig, he also developed the very first global MOOC with 160,000 students enrolled. His company, Udacity, has educated over 5 million students, and has been valued at more than one billion dollars. Google Scholar ranks Thrun's publication h-index #14 worldwide in all of computer science. Thrun also founded Google X, where he founded Google Glass among many other projects. He was elected into the National Academy of Engineering and the German Academy of Sciences at age 39. Fast Company named Thrun the fifth most creative person in business, and Foreign Policy touted him Global Thinker #4. He won numerous awards, including the prestigious Max Planck Research Award.

Interesting in Hosting ICWSM-19?

AAAI, in cooperation with the ICWSM Steering Committee, is currently seeking proposals for a host city for the Thirteenth International AAAI Conference on Web and Social Media (ICWSM-19). The conference is typically held Monday - Thursday during the timeframe of mid-May through mid-June. Final selection of a site will be made by August 2017. For more information about proposal requirements, please write to icwsm19@aaai.org.

Note: ICWSM-18 will be held at Stanford University in Palo Alto.

Courtesy, iStock

Join Us in New Orleans for AAAI-18!

The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18) and the Thirtieth Conference on Innovative Applications of Artificial Intelligence (IAAI-18), will be held in New Orleans, Louisiana, USA, February 4-10. The technical conference will continue its 3-1/2 day schedule, followed by the workshop and tutorial programs. AAAI-17 will arrive in New Orleans just prior to Mardi Gras and festivities will already be underway. Enjoy legendary jazz music, the French Quarter filled with lively clubs and restaurants, world-class museums, and signature architecture. New Orleans' multicultural and diverse communities will make your choices and and experience in the Big Easy unique. The 2017 Call for Papers will be available soon.

Please join us in 2017 in New Orleans for a memorable AAAI!

www.aaai.org/aaai18

AIIDE-17 to be Held in Snowbird, Utah

Please join us for AIIDE-17, to be held October 5-9, 2017 at the Cliff Lodge at Snowbird in Little Cottonwood Canyon, Utah, USA. AIIDE-17 is the next in an annual series of conferences showcasing interdisciplinary research on modeling, developing, and evaluating intelligent systems in entertainment. AIIDE-17 provides a meeting place for academic AI researchers and professional software developers to discuss the latest advances in entertainment-focused AI. The conference has a long-standing history of featuring research on artificial intelligence in computer games. We also invite researchers, developers, and digital artists to share ideas on topics at the intersection of all forms of entertainment and artificial intelligence broadly. AIIDE-17 will feature invited speakers, paper sessions, workshops, tutorials, playable experiences, panels, posters, the Starcraft AI Competition, and a doctoral consortium.

This year, the AIIDE conference will feature a special topic of "Beyond Games." In addition to general topics of interest in game AI, we welcome submissions featuring innovative forms of interactive digital entertainment, including but not limited to human-robot interaction, computer music, generative art, physical computing, procedural animation, and digital improvisation. The special topic will also connect to keynote speakers, panels, paper sessions, and other aspects of the conference program.

Submissions for all programs are due May 25, 2017. For more information, please visit www.aiide.org, or write to aiide17@aaai.org.

Join Us in Quebec City for HCOMP-17

The Fifth AAAI Conference on Human Computation and Crowdsourcing will be held October 24 – 26, 2017 at the Hilton Regency Quebec in Quebec City, Canada. HCOMP-17 will be colocated with the User Interface Software and Technology (UIST). HCOMP is the premier venue for disseminating the latest research findings on crowdsourcing and human computation. While artificial intelligence (AI) and human-computer interaction (HCI) represent traditional mainstays of the conference, HCOMP believes strongly in inviting, fostering, and promoting broad, interdisciplinary research. This field is particularly unique in the diversity of disciplines it draws upon, and contributes to, ranging from human-centered qualitative studies and HCI design, to computer science and artificial intelligence, economics and the social sciences, all the way to digital humanities, policy, and ethics. We promote the exchange of advances in human computation and crowdsourcing not only among researchers, but also engineers and practitioners, to encourage dialogue across disciplines and communities of practice.

Submissions are due May 4, 2017. For more information, please visit

humancomputation.com, or write to hcomp17@aaai.org.

2017 Fall Symposium Series

Mark Your Calendars! The 2017 AAAI Fall Symposium Series will be held Thursday through Saturday, November 9-11 2017, at the Westin Arlington Gateway in Arlington, Virginia, adjacent to Washington, DC. Proposals are due April 21, and accepted symposia will be announced in late May. Submissions will be due July 21, 2017. For more information, please see the 2017 Fall Symposium Series website (www.aaai.org/Symposia/Fall/fss17.php).

AAAI Executive Council Elections

Please watch your mailboxes for an announcement of the 2017 AAAI Election. The link to the electronic version of the annual AAAI ballot will be mailed to all regular individual AAAI members in the spring. This year, the membership will vote for four new councilors, who will each serve three-year terms. The online voting system is expected to close on June 16. Please note that the ballot will be available via the online system only. If you have not provided AAAI with an up-to-date email address, please do so immediately by writing to membership17@aaai. org.

Member News: In Memoriam

Adele Howe

AAAI deeply regrets to report the sad news of Adele Howe's passing on January 20, 2017. Howe, a AAAI Fellow, was a professor of computer science at Colorado State University (CSU), and was named a Professor Laureate in the College of Natural Sciences at CSU in 2010. She attended The University of Pennsylvania and in 1983 received her B.S.E. in computer science and engineering. She then joined ITT's Artificial Intelligence Research Group. She completed her Ph.D. in Computer Science at the University of Massachusetts in 1992. While at the University of Massachusetts she did pioneering work on autonomous agents operating in complex environments, and simulated firefighting in Yellowstone National Park.

Howe's later research focused on autonomous agents, planning systems, human-centered computing, as well as search and combinatorial optimization. One of her many projects was helping to develop the first metacrawler search engine for navigating the web (before there was Google). In 2000 and 2001 she served the United States as a member of the U.S. Defense Science Study Group, a selective and small group of academics charged with reviewing all branches of the military, and anticipating how changes spurred by advanced research will impact the military. She also helped to develop a satellite communication scheduling system for the U.S. Air Force.

In 2015, Howe was named a Fellow of AAAI, and was honored for her significant contributions to the theory, practice and evaluation of automated planning, scheduling and other AI technologies, as well as service to the AI community. In 2016, she was the inaugural recipient of the ICAPS (International Conference on Automated Planning and Scheduling) Distinguished Service Award.

Howe was a longtime volunteer for AAAI in a host of roles, and most recently served as the cochair for the Senior Member track at AAAI-17. Her commitment to fostering research in artificial intelligence and her passion for building diversity in the field of computer science and artificial intelligence will be sorely missed.

Ranan Banerji

AAAI also reports the passing of AAAI Fellow Ranan B. Banerji, professor of math and computer science at St. Joseph's University in New Jersey. Born in Kolkata, Banerji earned his undergraduate degree in physics at Patna University in Bihar, India, in 1947, and his doctorate in physics at the University of Calcutta in 1956. Before joining St. Joseph's, Ranan Banerji also held professorships at Case Western University in Ohio and Temple University in Philadelphia. At the time of his election as a AAAI Fellow, AAAI honored Banerji for his pioneering and continuing work in the formal foundations of problem-solving, game-playing and machine learning. He conducted seminal early work in search, game playing, and game trees, and wrote widely, including five books, on artificial intelligence.

AAAI Executive Council Meeting Minutes

The 2016 AAAI Executive Council meeting was held November 28, 2016, via teleconference.

Attending: Subbarao Kambhampati, Tom Dietterich, Yolanda Gil, Ted Senator, Sonia Chernova, Vincent Conitzer, Boi Faltings, Steve Smith, Charles Isbell, Diane Litman, Jennifer Neville, Blai Bonet, Michela Milano, David Smith, David Leake, Ashok Goel, Carol Hamilton.

Not Attending: Mausam, Kiri Wagstaff, Qiang Yang.

Rao Kambhampati brought the meeting to order at 9:00 AM and thanked everyone for joining. He reviewed the documents in Google and Blue Jeans. In addition, he gave an update on recent activities in which he has been involved.

Presidential Issues

Kambhampati and Carol Hamilton initiated the AAAI Affiliates program, whereby all individuals who have served on AAAI conference program committees or attended AAAI conferences as nonmembers are automatically signed up as AAAI Affiliates. Affiliates do not have member privileges, but do receive the AI Alert and other announcements about AAAI activities. Currently, there are 3,541 affiliates, so there are as many people helping with and attending AAAI programs as there are members.

Kambhampati has also been working with the China Computer Federation to work out a Memorandum of Understanding that will be of mutual benefit to AAAI members and to the Chinese AI community, including mutual advertising and other cooperative efforts. Kambhampati sent a congratulatory note to the CCF upon their 40th anniversary. Tom Dietterich also spoke about AAAI in a recent meeting in Beijing.

Kambhampati, representing AAAI, will serve on the advisory board of the newly formed Partnership in AI, recently established to study and formulate best practices on AI technologies, to advance the public's understanding of AI, and to serve as an open platform for discussion and engagement about AI and its influences on people and society by drawers. He will join other industry leaders from Apple, Amazon, Facebook, Google, DeepMind, IBM, and Microsoft, as well as representatives from UC Berkeley, the ACLU, the MacArthur Foundation, and the Peterson Institute of International Economics. The first physical meeting will be February 3 in San Francisco.

Kambhampati is also investigating the possibility of a joint conference with ACM on AI ethics and will be following up on this with the Ethics Committee.

The AAAI office had to migrate their website quite unexpectedly in the past few weeks, and Kambhampati reported that the migration had been completed successfully, just in time for the submission of close to 800 AAAI-17 proceedings papers.

Finance Committee

Ted Senator reviewed the budget approval process for the upcoming year. The budget is officially approved by the Council prior to the beginning of the next fiscal year (the calendar year). Most of the work is done by the staff and most of the data is based on historical trends during recent years. However, where known, the unique costs associated with chosen venues for programs are incorporated into the budget. If new programs or activities have been requested by committees, those are also incorporated into the budget. Senator drew the Council members' attention to the line in the program that explains how much of the operating reserve will need to be used to support the current projections. He noted that as long as this number is in the 3-5 percent range, it is an acceptable amount. A full set of budget guidelines was circulated to the Council prior to the meeting.

Senator also noted that the projection for 2016 is a surplus. 2017 will be a more expensive year due to the higher costs for the main conference. In addition, there will be a rehaul of the AAAI website, resulting in a one-time expense for the Association. Kambhampati added that he would like the association take on further commitments, given the current large surge in interest in AI, and would like to use the full 5 percent if opportunities arise to do so.

After some further discussion, Tom Dietterich moved to approve the budget, Senator seconded the motion, and the budget was approved unanimously.

In further financial news, Senator noted that he had circulated the tax return to the Council. This is a good practice required by our auditor to retain transparency in the process. Finally, Senator announced that he will be transitioning out of the Secretary-Treasurer position at the end of 2017, and in conjunction with other members of the AAAI Executive Committee, has worked out transition plan. David Smith, who has served on the Finance Committee for several years, will take over as Secretary-Treasurer upon his retirement from NASA, and noted that Smith will attend this and all future meetings during the transition period. Kambhampati thanked Senator for his 14 years of service as Secretary-Treasurer, as well as all his other service for the association.

Membership Committee

Kambhampati announced that Blai Bonet will be taking over as the Membership Committee chair. Bonet noted that he will be working with the committee on several issues, including how to develop the affiliates program. The committee will also be reviewing the current list of people who have applied for the Distinguished Speaker program, and will report on this at the February meeting. It is anticipated that a budget will have to be established to pay for some travel for the speakers. Kambhampati noted that we need to reach out to industry members more effectively, as several people have mentioned to him that they were not sure if AAAI is open to them.

Conference Committee

Carol Hamilton reviewed the major differences between the 2016 conference and the 2017 conference. The conference technical program will be significantly bigger with close to 150 additional technical papers accepted for presentation, including the main track, special tracks, senior member, and demo tracks. Other activities will remain similar to 2016, although the robotics program will not be held. The other big difference will be budgetary, as San Francisco will be significantly more expensive than Phoenix. This was reviewed during the budget discussions. In addition, Yolanda Gil noted that plans were underway for an Industry Day event, comprising a series of accessible talks geared toward industry. Possible days for this event will be Sunday or Friday following the main conference. Gil also mentioned that it would be a missed opportunity to not reach out to industry, given that AAAI-17 is in the Bay Area. Gil has agreed to spearhead this, but is seeking volunteers to help design the best program and promotion strategy.

The committee discussed other programs or initiatives that would be of interest to industry, including the AI Job Fair, also scheduled for Sunday, IAAI-17, and highlighting workshops or tutorials that might be accessible. All agreed that good promotion will be important to reach beyond the normal attendees at AAAI. Further discussion revolved around possible fees for the event, the targeted audience, and the scope of the event. Gil said she would do some further investigation to clarify these issues. Kambhampati would like to establish a strong connection to industry and this could be the jumping board to do so.

Symposium

Carol Hamilton noted that Gita Sukthanker will be stepping down as Symposium Chair, and will be replaced by Christopher Geib. The committee is currently seeking a new cochair, and may seek a person outside North America. The Council agreed that international outreach is very important, and that the resulting slight increase in travel expenses was fine.

Government Relations

Steve Smith reported that the commit-

tee will be responding to an RFI from the British government, and had earlier put in a response to the OSTP about the challenges and risks of AI. The committee will be spending some time reflecting on what they can do during the new administration.

Publications

David Leake reported that Ashok Goel took over as editor of the magazine officially on October 1. Goel noted that he is just getting familiar with the full process, and that his workload associated with the magazine is greater than he thought it would be. He would like to spend some additional time during this familiarization period with the current publishing structure, including the web and digital versions. He will then tackle the electronic edition. He reviewed the plans for the upcoming issues, and noted that there is the possibility of an *AI Magazine*-sponsored panel at AAAI-17, which will be converted to an article in a future issue. He hopes to start a dedicated column for past presidents of AAAI, as well as AAAI Fellows. Leake thanked Goel on behalf of the Publications Committee for all his work and the energy that he is putting into the magazine.

International

Rao Kambhampati announced that Qiang Yang will serve as the new chair for the International Committee. He noted that Yang has been very helpful in making connections for AAAI in China, especially with the recruitment of program committee members for AAAI-17.

Conflict of Interest Policy

Ted Senator reported that he will be developing a Conflict of Interest policy statement and agreement for members of the Executive Council before the next meeting. He reviewed the three duties of care briefly, and noted that a COI policy is highly recommended for all nonprofits. It is a standard request from the AAAI auditor.

The IBM Watson AI XPrize

Kambhampati noted that he is on the steering committee for this event that challenges teams to attempt their own Moonshot achievements, demonstrating AI-human collaboration to address humanity's Grand Challenges. Professional organizations will provide experts to help teams.

AIHub

Tom Dietterich reported that he is continuing to work with Sabine Hauert on the AI Hub project, a site that will provide one-stop shopping for AI-related items. The goal is still to create a cooperative effort among 4-5 different organizations who will each contribute $15-20K each to support the venture. Although he requested an initial $5K to get the project off the ground, it was determined that Council approval was not needed for this amount and it will be considered separately as needed. Dietterich asked for volunteers to help with the development of the project, and will have an update at the February meeting.

Kambhampati thanked everyone for coming, wished everyone a happy holiday season, and the meeting adjourned at 10:45 AM.

AAAI Conferences Calendar

This page includes forthcoming AAAI sponsored conferences, conferences presented by AAAI Affiliates, and conferences held in cooperation with AAAI. AI Magazine *also maintains a calendar listing that includes nonaffiliated conferences at www.aaai.org/Magazine/calendar.php.*

AAAI Sponsored Conferences

AAAI Spring Symposium Series. The AAAI 2017 Spring Symposium Series will be held March 27–29, 2017, at Stanford University adjacent to Palo Alto, CA USA.

URL: www.aaai.org/Symposia/Spring/sss17.php

Eleventh International AAAI Conference on Web and Social Media. ICWSM-17 will be held May 15–18 in Montréal, Québec, Canada.

URL: www.icwsm.org/2017

The Thirteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. The Thirteenth Annual AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment will be held October 5 – 9, 2017 at the Cliff Lodge at Snowbird in Little Cottonwood Canyon, Utah, USA.

URL: aiide.org

Fifth AAAI Conference on Human Computation and Crowdsourcing. The Fifth AAAI Conference on Human Computation and Crowdsourcing will be held October 24 – 26, 2017 at the Hilton Regency Quebec in Quebec City, Canada.

URL: humancomputation.com

AAAI Fall Symposium Series. The AAAI 2017 Fall Symposium Series will be held November 9–11, 2017, in Arlington, Virginia adjacent to Washington, DC. USA.

URL: www.aaai.org/Symposia/Fall/fss17.php

The Thirty-Second AAAI Conference on Artificial Intelligence. AAAI-18 will be held February 4–10 at the Hilton New Orleans Riverside Hotel, New Orleans, Louisiana USA.

URL: www.aaai.org/aaai18

Thirtieth Innovative Applications of Artificial Intelligence Conference. The IAAI-18 Conference will be held February 4–10 at the Hilton New Orleans Riverside Hotel, New Orleans, Louisiana USA.

URL: www.aaai.org/iaai18.php

Conferences Held by AAAI Affiliates

The 16th International Conference on Autonomous Agents and Multiagent Systems. AAMAS 2017 will be held May 8–12, 2017 in São Paulo, Brazil.

URL: aamas2017.org

Thirtieth International Florida AI Research Society Conference. FLAIRS-2017 will be held May 22–24, 2017 on Marco Island, Florida, USA.

URL: www.flairs-30.info

The 27th International Conference on Automated Planning and Scheduling. ICAPS-17 will be held June 18–23, 2017 in Pittsburgh, PA USA.

URL: icaps17.icaps-conference.org

Conferences Held in Cooperation with AAAI

The 16th International Conference on Artificial Intelligence and Law. ICAIL 2017 will be held 12-16 June, 2017 in London, UK

URL: nms.kcl.ac.uk/icail2017

Thirtieth International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems. IEA/AIE-2017 will be held June 17–21, 2017 in Arras, France.

URL: www.ieaaie2017.org

Thirteenth International Conference on Logic Programming and Nonmonotonic Reasoning. LPNMR'17 will be held July 3-6, 2017 in Espoo Finland

URL: http://lpnmr2017.aalto.fi

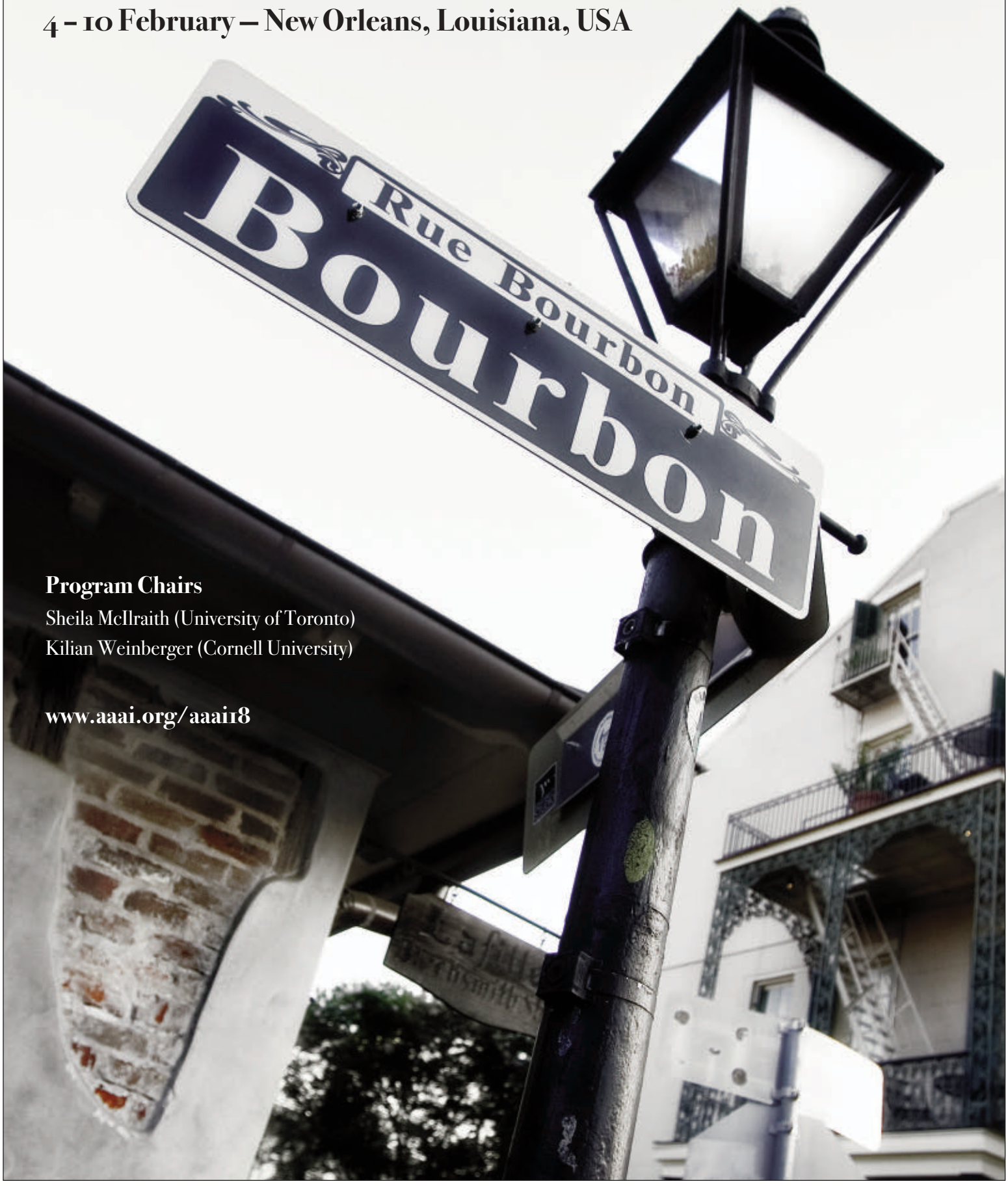
Thirteenth International Conference on Intelligent Environments. IE'17 will be held August 21-25, 2017, in Seoul, Korea

URL: www.intenv.org/?q=conferences/ie17

 ISSN 0738-4602

AAAI-18

4 – 10 February – New Orleans, Louisiana, USA



Program Chairs

Sheila McIlraith (University of Toronto)

Kilian Weinberger (Cornell University)

www.aaai.org/aaai18


ICWSM
Join Us in Montréal, Québec, Canada
on May 15–18, 2017
The Eleventh International AAAI Conference on
Web and Social Media (ICWSM-17)
www.icwsm.org/2017