CAS and CASCON Honouring 25 Years of IBM Research and Innovation

Edited by

Marin Litoiu
Kelly Lyons
Hausi Müller
Joanna Ng

Forward

The Centre for Advanced Studies (CAS) was established in 1990 at the IBM Toronto Software Laboratory. The primary aim of CAS is to facilitate the transfer of research ideas into the products and services in IBM. The first CASCON was held in November 1991 at IBM Toronto Lab. In 2006, the CAS model for research collaboration was recognized by NSERC with the 2006 Leo Derikx Synergy Award for Innovation awarded to IBM and 12 university partners. In recognition of the 25th Anniversary of CAS and CASCON, we solicited short chapters for a book to document and highlight the importance and influence of CAS within IBM, academia, and Canada.

We invited people from the CAS ecosystem to write chapters that articulate the various ways in which CAS and CASCON have impacted their collaborative research, innovation, and technology exchange over the past 25 years. We were looking for submissions from different perspectives that highlighted and documented individual experiences with CAS and CASCON as well as for descriptions of mechanisms, research results, technology transfers/exchanges, influential results, and HQP training for which CAS and CASCON are so famous.

The response was enthusiastic and the result is 28 high-quality chapters written by 33 past and current CAS Research Staff Members, CAS Directors, IBM Executives, IBM Developers, CAS Students, CAS Faculty Members, and industry collaborators. Through the chapters in this book, readers can trace the history of CAS and read stories of early CASCON experiences and large-scale CAS projects. The breadth of perspectives in this book articulates the full scope and influence of CAS and CASCON, describing significant research resulting from CAS collaborations and documenting the impact of CAS and CASCON on individuals and institutions. The personal accounts pay tribute to the significant and lasting effect of CAS and demonstrate a passion for an organization that is rare and extraordinary. There are stories of friendships and relationships that have been built and sustained over the past 25 years—because of CAS. There are descriptions of significant innovations and influential outcomes that resulted from CAS projects. There are also suggestions for the way forward and mechanisms for measuring future CAS success.

We set out to document the influence and impact of CAS and CASCON. This book does that and much more. Through these chapters, the authors have illustrated how this one organization initiated within IBM Canada has changed the IBM research and development landscape, the Canadian computer science and software engineering communities, the Canadian software industry, and the lives of the people in this amazing CAS ecosystem.

We would like to thank every single person who contributed to the great success story of CAS and CASCON over the past 25 years. We especially would like to thank all the authors of this volume who shared their very personal experiences for this wonderful tribute to research and innovation at IBM CAS Canada.

Marin Litoiu
York University

Kelly Lyons
University of Toronto

Hausi Müller
University of Victoria

Joanna Ng
IBM Canada Software Lab

Editors

Table of Contents

CAS Leadership

The Early Days of CAS

CAS Students and Highly Qualified People

CAS Projects: Distributed Systems

CAS Projects: Software Evolution

CAS Projects: Software Engineering

International Collaboration

CAS Leadership

Let the History of CAS Shape its Future Direction

Jacob Slonim
Faculty of Computer Science
(emeritus)
Dalhousie University
slonim@cs.dal.ca

Michael McAllister
Faculty of Computer Science
Dalhousie University
mcallist@cs.dal.ca

John Botsford
CAS Principal Investigator
(retired)
botsford@sympatico.ca

CAS has left an indelible stamp on the IBM Toronto lab and on research partnerships between academia and industry over its 25 years. It has helped to drive innovation, has built long-lasting research relationships, and has influenced models of research at IBM, at other companies, and at the public-funding agencies in Canada. We look back at how CAS grew to its current state and examine what we can learn from the first 25 years that could help CAS in the years to come.

1. WHERE CAS CAME FROM

Knowing the ideas behind how CAS started is important to understand the choices that were made for CAS. The primary role of CAS was to provide a vehicle to inject new ideas and directions into the existing work at the IBM Toronto Lab. Those ideas and directions needed to be shaped by, but not shackled by, the strengths and initiatives within the lab and needed to provide effective knowledge transfer to the lab beyond the opportunity to hire the participating students upon their graduation. It aimed to leverage the strengths of both the academics and the IBM staff who participated in CAS.

Truly joint projects between academia and IBM became the cornerstone of CAS. Some projects stretched and explored new paradigms with large teams and led to innovation in areas such as distributed databases, directory services, and electronic commerce. Other projects led to close-knit work between students and developers, which evolved into CAS student fellowships and the matching CAS internships. In both instances, success came from helping all participants, both IBM and academic, to understand and help work through the issues, items of value, and stresses faced by their counterparts. All participants needed to understand and embrace each other's goals: engineers learn in order to build; scientists build in order to learn.

Engaging "the right" members for the projects and fostering mutual understanding among the teams was a critical role of the CAS Director. Success came from recognizing the potential in ideas and in individuals, interacting with academics at conferences and wherever else they gathered, finding the right mix of team members for new ideas, and fostering a culture of sharing ideas across the perceived academia-industry divide in a way that each could appreciate the intent and constraints around the ideas. The ability to integrate new talent into CAS opportunistically, whether as large or small projects, provided an agility to CAS.

Meanwhile, CASCON originated as a celebration of CAS achievements, a vehicle to deepen the network of individuals from which CAS could draw for talent, and a means to foster a community around topics that simultaneously impacted research and development. CASCON was a means to an end and never and end of its own.

Impediments to collaboration eventually arose in the projects. We outline some of the more significant factors from the early days of CAS.

Change is an ever-present factor in development projects and uncontrolled change is the biggest single killer of development projects. Change can include the need to handle issues not previously considered important, the need to consider additional classes of customers, and mistakes in early requirement statements. Change also arises in a positive manner as projects uncover new directions and opportunities. Projects needed to adapt in what we might currently call agile development or agile research; that iterative industry-academic adaptation was not a prevalent mode of industry-based research at the time. Having the students spend significant time in the Toronto lab, especially over an academic term, proved fundamental in adapting to the ongoing changes and in growing an agile mindset among the researchers and the developers.

Differing environments, both technological and organizational, provided further impediments. The technological differences were more easily overcome through the flexibility of researchers (typically PhD students) who became embedded in the commercial development environment for a period of time. Organizational structures were a different story. While commercial software developers would grudgingly adapt to the world of lawyers, accountants, and multi-tiered reporting structures, researchers found few incentives to engage in those elements. The CAS staff needed to take on the role of smoothing out these organizational issues.

A potential lack of continuity of involvement in a project was a concern as researchers sought long-term dedication in a research topic to understand and overcome the intricacies of a problem. Meanwhile, it is common for professionals in commercial development to change development groups before a project is finished for reasons like career advancement or urgent needs for additional people in priority areas. It was critical to have the developers' involvement become part of their job and suitably recognized as part of the assessment of their work by their managers. Additional support and incentive was then provided by recognition at CASCON and the development of, and participation in, recognized software engineering education by partner academic institutions for senior technical staff.

The nature of CAS is to bring experts together. In commercial development, it is common to have one or two individuals who emerge as a "guru" because of their special knowledge or leadership qualities, without regard to their title. It became essential that researchers develop strong and positive bonds with the developers' gurus, especially as the researchers themselves either are or have gurus of their own. Everyone needed to find ways to support and build upon one another's knowledge rather than become territorial around their knowledge base.

Accountability is important in any project. What constitutes accountability varied across CAS members. Product managers are measured by how well his or her product does in the marketplace and success of product managers is accomplished through, and inherited by, those to report to them. Researchers face other measures such as publications, funding, training of students, and demonstrable impact. In all of the projects, the underlying mindset needed to be a winning situation for all participants to which everyone contributed.

The developers, researchers, and CAS staff faced pressure to drive the project outcomes to be product-ready or to link CAS activities to IBM promotion and marketing initiatives. Giving in to the pressure to generate product-ready outcomes results in a loss of the potential new directions, a requirement of students to be developers which is askew from their role in the project, and a withdrawal from chasing new ideas opportunistically. On the other hand, resisting the pressure could result in a reluctance of key developers to participate since their contributions were needed on products. Of the two directions, CAS needed to resist the "productization" pressure to foster the generation of innovation. The buy-in and participation from high-level managers was essential in maintaining that stance.

Key elements in the early success of CAS were its focus on the best people from across the world and its development of a social network in addition to the projects. First, we needed to engage strong students with select and knowledgeable developers on mission-oriented (not mission-constrained) work. All the individuals needed to be collaborative, open-minded to new ideas, and able to see the potential in ideas. Second, we needed to use each member to his or her strengths. We didn't want to make students into product-level developers or require that developers chase down every esoteric twist of a problem solely for the sake of completeness. Third, we needed to set clear expectations around the nature of the team, such as a mutual understanding and respect of the differing pressures, timelines, and capacity for change, among the team members and their supervisors or managers. Fourth, we needed to develop a mutual sensitivity for what each team member in the project wanted and needed from the project. Everyone needed to be onboard with the diversity of definitions of "success" for the project and to buy in to helping one another achieve their success points.

2. WHERE CAS CAN BE

Academia and industry have evolved since the first days of CAS. We believe that the original vision of CAS as an agent to infuse new ideas and directions into IBM remains viable. Changes in the environment necessarily require changes in approaches to contribute to that vision. Some of the key environmental changes to account for include:

- Restrictions (perceived or real) on the amount and flexibility of funding and other resources, both within IBM and through external funding agencies;
- Changes in the nature of development and the role of development in academia and in industry, which includes a decline in product development from first principles in favour of increased development from existing software, frameworks, and toolkits;
- Increases in accountability, transparency, process, and structure to manage the growth in interest in CAS from its early successes and to address issues around openness, impartiality, and return-on-investment;
- Long-term relationships between faculty, students, and staff through CAS projects in which people have invested in learning or adapting to one another's ideologies, values, and environments; and
- An increased industry trend to acquire start-up companies to bring in large-scale innovation and direction.

Amid those changes, we single out three directions on which to focus. These directions are repeatedly subject to a slow, but ultimately significant, shift: focusing on the right team, keeping CAS mission-oriented, and expanding on exploration.

Team composition must remain at the core of CAS. As in the early years, we must recruit open-minded, collaborative individuals. On the academic front, we must develop and maintain connections to a diversity of academic institutions, both locally and globally. Despite the long-term relationships built in CAS to date, there must be a cycling of researchers through the program to infuse fresh direction to projects. A consequence is a need to develop pathways for the long-term CAS participants to evolve their engagement with IBM beyond CAS fellowships. On the IBM front, we need to renew the active participation of the senior technical staff and managers and ensure that the most knowledgeable and productive developers maintain the incentive and encouragement to participate in CAS projects. These IBM employees need to become participants in identifying and pulling in researchers for projects. CAS needs to find ways to support that activity within IBM, including sponsoring these employees to attend conferences and other venues where the top researchers network and present their ideas. Both IBM and academia need to ultimately see the senior technical staff as the face of CAS.

CAS has evolved into being the primary contact point of university researchers in Canada with IBM. CAS must foster an active role in that relationship as an entity that forms teams around mission-oriented directions. Early CAS successes stemmed from large initiatives around distributed databases and directory services where CAS actively sought out a diverse selection of researchers to tackle significant directions. Such a role is different from responding to requests from researchers seeking partners for government research grants.

Both of the activities mentioned in focusing on the research team rely on investing in the CAS leadership and senior technical staff to ensure that these individuals are able to:

- engage the researchers on their terms;
- recognize and encourage potential in research students and ideas;
- engage and draw researchers and IBM developers into ongoing projects; and
- help foster and develop the work connections and networks within projects, which can require a significant investment in the early stages of projects.

One element that contributes to being effective in these latter areas is a nimbleness within CAS to connect new individuals with existing teams and projects as the opportunity arises.

The second direction of focus is keeping CAS mission-oriented, which is different from saying that all CAS activities must be product driven. The traditional view of being mission-oriented is that projects need to tie directly to innovation within the products developed in the IBM lab. That element remains valid, but it is not the whole of the picture. CAS can be used to help a lab stretch its mission and gain resilience as the marketplace changes. It can open new directions to a lab that can expand the scope of the lab's activities. CAS also has the potential of providing a vehicle to collaborate with other companies, through the researchers and projects, in ways that do not threaten internal development. These collaborations can be best achieved through larger projects that target uncharted technical landscapes where each of the collaborating companies can establish a future niche or foothold through the collaboration.

The third direction of focus is a re-investment or expansion into exploratory projects. Risk is an inherent element of innovation. CAS provides a vehicle through which higher-risk ideas can be explored while lowering the direct risk within the lab. Researchers and academia are expected to push the boundaries of what we know. In that realm, failures that stem from the complexity of a problem are acceptable and normal events. The same failures are not as forgiving for a developer who is facing development deadlines. Use CAS projects to chart new directions where uncertainty is greatest, even to including projects on more basic topics whose tie to the lab mission is a larger stretch. Acknowledge that some of these projects will fail; accept and learn from those that fail from the complexity of the problem (but not those projects that fail due to negligence or a lack of collaboration).

Innovative students today are often interested in or lured by entrepreneurial activities. CAS can provide the vehicle to retain these students within IBM who seek to explore and push the boundaries as a complement to the developer career stream or the competition to qualify for one of IBM's research labs.

As a means for exploration, we need to encourage the adoption of ideas for CAS projects that originate within the universities. IBM always has the ability to pursue internally-generated product ideas with developer groups. However, those ideas can be constrained by the surrounding corporate pressures. Partner universities can break away from those constraints more easily if CAS remains open to their project ideas. The key is to have CAS bring those ideas to a proof-of-concept point, whether within or without the IBM context, to gauge their long-term potential.

As at its genesis, CASCON must keep a complementary role to the CAS projects. While easy to state, the amount of time, energy, funding, and profile that goes in to CASCON will always have the pull of positioning CASCON as an entity of its own with goals that may not add to those of CAS. CASCON remains at its best when it focuses on providing a networking platform between diverse communities that don't ordinarily meet, on organizing workshops in which the participants leave with an agenda and action items on which to build future CAS projects, and on sharing and celebrating the milestone achievements of existing CAS projects.

Workshops remain at the core of CASCON. We need to use the workshops to spark and develop new ideas that can touch a broader audience and that have an impact following CASCON itself. Some directions that could be tried follow:

- Create targeted workshops to draw in international talent. While this is already to goal of many CASCON workshops, we need to provide incentives and means for more international talent to participate, whether in recruiting and supporting their involvement or providing the means for meaningful remote participation in workshops.
- Set a target of 2-3 new CAS projects to be approved from the best future plan that comes out of workshops. Use the workshop times to explore, develop, and refine the project idea.
- Define workshops for researchers (students and faculty) who are looking to become involved in CAS to have a mix of ideas, introduce them to senior technical developers, refine mission-oriented project ideas, and have a meaningful chance to the researchers to become connected with a CAS project. We need to introduce and feed a flow of researchers through CAS.

- Host workshops on education directions and promote innovative ways to support and deliver IT education. Many of our educational programs continue in the same mode as was used when IT first appeared in universities. However, the reach of IT and the number and diversity individuals who rely on IT has grown. CAS can take a leadership role in addressing that growth.

CASCON's footprint has the chance to grow by using even more of the networking systems to reach a global audience. Keynote speakers are regularly recorded now. Those recordings and live workshop feeds could be broadcast more broadly to involve IBM staff at other labs and global researchers. With appropriate collaborations, some workshops could tie in with graduate class events at universities or a technology-driven analog to the TEDx movement.

The relationships between the researchers and the IBM developers were always at the heart of CAS. Those relationships provided the instrument and goodwill to bridge between the academic and industrial environments. They fueled CAS's synergistic approach to innovation. However, all relationships including those in CAS need attention and need to be valued and worked on by all parties. Should the relationships and synergy fall from their central roles in CAS then we are faced with a harsh question: has CAS's part come to an end, possibly to be replaced by something else, or can we reinvest in CAS, preserve the elements of CAS that work, and reshape the elements that aren't serving their purpose to have CAS emerge stronger yet?

There is no doubt that barriers exist to many of these ideas for CAS and for CASCON. However, these barriers will be no more insurmountable than those faced in the early days of CAS. CAS has demonstrated 25 years of successes. At its inception, IBM felt that its mandate was important, as did the students and researchers who became and remain partners in CAS. Despite the differences in their environments, the developers and researchers made CAS work as witnessed by the number of students who have been involved in CAS over the years. A renewal of CAS, including creative thinking in how to attract and engage new talent and renewed support in time, ideas, and staff from senior technical leaders, can direct CAS for further years of success.

There is More Than One Way to Create Research Momentum

John G. Schwarz
Founder and CEO of Visier Inc
Unit 300, 575 W 8th Ave
Vancouver, BC V5Z 0C4
1 888-277-9331
john.schwarz@visier.com

The predominantly popular approach to creating momentum in pre-competitive technology development is to fund a start up with angel money. The ingredients are simple. One needs a credible inventor with a great idea, a small team of collaborators and a willing investor to provide the initial capital. Thousands of such projects are started every year, with a reasonably predictable (but low) track record of success.

The outcome of these projects is an early proof point, a prototype, sometimes a patent application. The market discipline is very effective at weeding out failures early and provides a ramp for projects that show promise.

The downside of this process is that the discoveries are almost always proprietary and restricted to a narrow group of potential exploiters, typically those that were the initial inventors or the owners of the documented IP. But this approach is not the only way to generate good ideas for eventual inclusion in commercial development.

Another very innovative and collaborative way was pioneered in the 1990's by Larry Achtemichuk and Dr Jacob Slonim of IBM in Toronto, Canada, with the creation of the Center for Advanced Studies (CAS) and its attendant annual research conference CASCON. This format uses angel funds made available by a strategic technology player (in this case IBM), and brings together researchers from the funding firm and from universities and other research institutions to collaborate on projects of shared interest. Often, these projects are leveraged by applications for funding from established national and international granting organizations.

The benefit of this approach is that the IP that is created is shared by multiple groups and usually made available to people who wish to build on this base. Also, since the collaborators are often students, it is a very effective way for them to learn about the prospective employer and enable them to continue their research projects with commercial backing. There is a downside - since the IP is shared, one must be very fast in exploiting the invention in order to realize commercial advantage. Often, the firm that provides the strategic funding does not react quickly enough and can lose the first mover advantage.

CAS provides a further benefit that is usually not available outside of the mainstream investment centers, nor to people who do not have a successful track record of independent entrepreneurship. In San Francisco, London or Shanghai, with access to massive local venture fund sources, it is relatively easy to find money, and people who have the experience of successful start-ups flock there. However, in areas further away from these centers of gravity and for people who are first time inventors, getting access to high risk investment sources is far more difficult, often impossible.

Here, the CAS model comes into its own. Any organization that is interested in collaborative early development can set aside a small collaborative research budget to provide the seed money. This can be enough of leverage to attract academics, students and subsequently research grants from governments.

The great advantage of this approach is that these Centers can be created virtually anywhere where there is an academic center of gravity and an interested commercial partner, which is in most cities that are home to an accredited university.

Another very valuable aspect of the CAS model is that multiple projects run in parallel in close proximity, so that experience and support resources can be shared, and there is a certain degree of cross-pollination of ideas and people. Such sharing gives rise to new ideas and early identification of problems and points of failure.

From the point of view of granting agencies, the CAS structure affords risk mitigation as the commercial partner typically delivers productive and secure facilities, experienced project management and financial discipline that is often missing in projects run purely from academic institutions or independent start ups.

In my experience with CAS, between 1991 and 1995, we initiated about 20 collaborative projects together with several universities, including Waterloo, Toronto, McGill, UBC, Simon Fraser, York, Dalhousie and Calgary in Canada, and Carnegie Mellon, Princeton, Technion and others internationally. The projects focused on computing languages, databases, user interfaces, communications, and applications. Several $ millions were raised by working together with the NSERC and US granting agencies to fund more ambitious multi-disciplinary work. IBM participated by providing seed money, project ideas, motivated professional developers, management oversight, computer equipment and facilities.

The work resulted in product inventions and many patents filed jointly by the researchers, students and universities. Every project underwent a rigorous peer review with the academics, and a project management review with IBM's lab leadership.

The annual culmination of the work was celebrated in the CASCON conference, an invitation-only prestigious event that brought together professors, students, researchers and commercial players from across the CompSci landscape. The conference featured joint projects that had reached a demonstrable stage, and invited speakers on a wide variety topics of shared interest. Papers describing the innovations derived from the work in CAS were presented by the researchers. CASCON created a prestigious award that was granted to the projects that were voted the top in their class.

To my knowledge, the CAS experiment was successfully replicated by other IBM labs in the US and internationally. The people who were engaged in this work in Toronto went on to share their excitement and success with the CAS model in their subsequent roles in other organizations. The network of professional/academic connections still thrives today.

I am very proud to have been one of the early participants and sponsors of this energetic and innovative approach to academia and industry collaboration. But the real credit for imagining and creating this unique engine for innovation must go to Dr Jacob Slonim, who was the head of Research at IBM Canada and head of CAS in the early 1990s, and later the Dean of Computer Science at Dalhousie University in Halifax.

Exporting the CAS Toronto Vision

Dr. Gabriel M. Silberman
Dell Research
One Dell Way
Round Rock, TX 78682
+1 914-216-3035
gabby_silberman@dell.com

Cheryl L. Morris
Dell Research
5480 Great America Parkway
Santa Clara, CA 95054
+1 408-219-4635
cheryl_morris@dell.com

ABSTRACT

As no doubt noted in other articles in this journal, the IBM Centre for Advanced Studies was established at the IBM Toronto Software Laboratory in 1990. It has played - and continues to play – a significant role in bringing together worldwide academia, government, and industry research organizations. It is a model for facilitating the transfer of advanced technology into strategic products. This article will describe how this successful model was "exported" to support collaborative research at local IBM development labs around the world.

1. THE CAS MODEL

As in many fruitful cooperative endeavors, collaborative research teams excel when their members represent diverse perspectives, bring to the table a blend of experiences, and share a clear, common goal. Building such teams to include members from industry, academia, and in some cases government, presents interesting challenges, but also great opportunities. The original design of IBM Toronto's Centre for Advanced Studies, or CAS Toronto, provided a winning formula for assembling a team from the various constituencies, around an agreed-upon goal, to carry out world-class research.

As we looked to replicate the CAS Toronto model at other locations around the world, it was clear we needed to start by identifying an insightful and well-connected local leader. Someone with the experience and credentials within the local IBM development organizations, as well as the reputation and connections within the relevant academic communities. This person should also have a good understanding of government funding programs for industry-academia collaboration, Intellectual Property (IP) policies for collaborative partnerships at various universities, and broad technical expertise. All these skills come into play when building a collaborative research program to solve problems from an industrial origin with a combined industry-academia team.

The next step involved procuring the commitment of local management for supporting a CAS organization. Clearly, the chosen leader's position and reputation went a long way toward securing the support and resources needed to launch a CAS. Also, many former IBM Toronto Lab executives, who knew and supported the creation and growth of CAS Toronto and had spread throughout IBM, served as strong advocates for CAS expansion among their peers in target locations.

Once a CAS begins operation, its leader needs to identify challenging problems facing some of the site's development teams. First and foremost, a problem needs to be intellectually challenging, to attract the attention of faculty who are experts in the problem's domain and provide the context for a graduate (MS or PhD) student's thesis. The problem should also tolerate the longer timelines typical of an academic setting, as opposed to the more accelerated tempo characteristic in industry. It is also critical to properly set the expectations on both sides with respect to this mismatch, while at the same time introducing fairly regular milestones to measure progress.

IBM development managers and technical experts, who will work closely with faculty and their students, play a key role to ensure the objectives of the project are met from the company's point of view. They also meet regularly with the research team to ensure any needed changes in direction prompted by new discoveries are identified and goals adjusted accordingly and in agreement with the team.

The selection of faculty to participate in a project relies on the CAS leader's extensive network of contacts. Engaging a professor for the first time may take some effort, and some faculty may not be amenable to working in an industry setting, but when a good match is made, it usually leads to a long-term relationship spanning many years. A discussion then takes place with the professor to jointly define a research project. This project provides the context for a graduate student's thesis topic, but is not necessarily identical to it.

Through the selected faculty member, access to talented graduate students, interested in tackling real world problems, is secured. The student gains the designation of CAS Fellowship Student and joins the research team to typically spend a summer, or longer depending upon availability and timing, working on the agreed upon project at a CAS location, as well as receive support through the academic year to carry out their thesis research. Because of the overlap between the project and thesis, the student actually works on both, albeit in different proportions, while at IBM or back at the university. If needed provisions were made for access to IBM systems and development environment.

It is interesting to note several PhD students who had been CAS students and chose an academic career, have become faculty collaborators upon completing their degree and securing an academic appointment.

One of the most important elements in a CAS collaboration is connecting research and development more directly, providing an experience for the student akin to working in industry with an advanced degree. The exposure to real business problems, and opportunities for networking with like-minded professionals are very valuable for a CAS fellowship student. Recruiting talented and tested candidates is significant for IBM, and the CAS model facilitates this effort.

CAS became the focal point for building relationships between IBM and academic institutions, working in harmony with IBM University Relations and other programs connected with academia.

Clearly, local customs, regulations and funding opportunities dictate variations on the CAS Toronto model, but generally speaking, the same basic principle holds everywhere: when developers and researchers share the same goals, and trust is established between the two communities, the results are almost always positive. These successful results are not solely based on measurements like new products or novel IP. CAS defines success based on establishing solid and enduring relationships with outstanding faculty and their students, access to leading-edge expertise, recruiting opportunities, and broadening awareness of the IBM brand.

2. THE GLOBAL VISION

CAS Toronto worked with universities from around the world, and talented students and faculty came from Spain (Barcelona), Russia (St. Petersburg), the United States, and Singapore, to mention a few.

From this international perspective, the idea was born to host students at CAS sites closer to their universities' home country or city. The CAS model could work as effectively in other geographies as it had in Toronto. A local CAS would also provide recruiting and networking opportunities on a more practical scale. The search for strategically located sites was begun.

By 1998 there were two CAS sites; the original in Toronto, and Austin CAS. The latter was established as an independent entity following a talk by John Swainson for the Research Management Council (RMC). One of the attendees at that meeting, David LaPotin, a manager at the Austin Research site, was intrigued by the description of the CAS model and contacted Gabby Silberman, who was Head of Toronto CAS. The groundwork was set for establishing Austin CAS, initially to solidify IBM's relationships with local universities.

In 1999, Gabby presented the concept of exporting the CAS model to Susan Puglia, Head of the IBM Toronto Lab. Susan was enthusiastic and offered her support. Later that year CAS Raleigh (RTP) was established under the leadership of Andy Rindos. Similar to the structure of Austin CAS, Andy's goal was to enhance IBM-academic relationships in the RTP region, and it was not yet involved in collaborative research projects between IBM and those academic institutions.

In 2000, Gabby's Toronto assignment ended, and he went back to the U.S., moving from IBM Research to the Strategy team of the IBM Software Group. He embarked upon his globalization plan by assembling a small team consisting of Gene Hoffnagle and Nancy Burns. Gene's role was to help evangelize the CAS model worldwide. Under Gabby's leadership, Nancy created the IBM Academic Initiative Program in collaboration with IBM University Relations. This program became an important element to increasing the awareness and use of IBM software in university curricula.

Gabby and Gene set about polling all the major IBM Software Group sites, reaching out to their extensive network of contacts at those sites. As word reached other IBM organizations like IBM Global Services and the Systems and Technology Group, people contacted Gabby for guidance on how to set up their own CAS.

As the reputation of CAS and CASCON grew in stature and became more internationally known, both in IBM and at academic institutions, the number of CAS sites also grew and shared best practices through annual meetings of CAS leaders.

One of the key attractions of establishing a CAS was the fact that each CAS was locally funded, supporting projects which were relevant to their home site and organizations. Other than sharing best practices and using the recognized CAS branding, each site was independent in formulating and executing its own strategy and project portfolio.

3. LEVERAGING THE GLOBAL PRESENCE

The Eclipse Technology Exchange (ETX) was an IBM initiative to broaden the use of open source Eclipse as an integrated development platform in academic institutions across the globe. To this end, a small team consisting of Connie Smallwood from IBM University Relations, and Cheryl Morris, Project Manager from CAS Toronto, was formed under Gabby's leadership to create and execute a comprehensive plan to accomplish this goal.

This plan included:

- organizing high quality workshops with peer-reviewed papers at major international software engineering conferences
- poster sessions at some of the same conferences
- a worldwide contest for the most original Eclipse applications, open to professional developers and students, the International Challenge for Eclipse (ICE)
- Eclipse Innovation Awards, granted to faculty based on a highly competitive selection process

In addition to broad goals, such as the ETX, the global CAS community was also leveraged to pursue specific research questions. A great example is the work Professor Margaret-Anne Storey from the University of Victoria (British Columbia, Canada) carried out during her sabbatical year.

A recipient of Eclipse Innovation Awards for three consecutive years, Professor Storey had been studying the tools and processes software development teams used to build and maintain their products. During her sabbatical she visited four IBM software development sites, in Dublin, Ireland, Cambridge, Massachusetts, as well as Toronto and Ottawa in Ontario, to familiarize herself with their practices. Besides serving as a bridge for best practices among those sites, Professor Storey gained valuable insights into the use of social networks, in particular tagging, within the software development context. These insights drove a number of Professor Storey's research projects over the following years, many of which were designed and developed using the Eclipse platform, thus enhancing its functionality.

4. SUMMARY

Through the expansion from its origin in Toronto, changes in leadership, and reach beyond software sites, the CAS model has proven its scalability, adaptability and resiliency. CAS has served each local site in its various missions, from fostering innovation to recruiting outstanding personnel and enhancing the image and standing of the hosting IBM entity. At the same time, the CAS community has become a platform for sharing best practices inside IBM, as well as a conduit for local faculty to collaborate with peers from other countries, with and without IBM involvement.

As of the date of this article, here are the global CAS sites:

- CAS Barcelona, Spain
- CAS Benelux, Netherlands
- CAS Böblingen, Germany
- CAS Brazil
- CAS Budapest, Hungary
- CAS Cairo, Egypt
- CAS Canada (Alberta, Toronto, Ottawa, Victoria)
- CAS Dublin, Ireland
- CAS France
- CAS HQ Region (New York) USA
- CAS India
- CAS Istanbul, Turkey
- CAS Japan
- CAS Portugal
- CAS Poland
- CAS Rochester (Minnesota), USA
- CAS Rome, Italy
- CAS RTP (North Carolina), USA
- CAS Russia
- CAS Tucson (Arizona), USA

The CAS legacy lives on in the enduring collaborative spirit of the people involved, at IBM and from universities around the world.

A CAS Retrospective

Joe Wigglesworth
IBM Canada Lab
Markham, Ontario, Canada
wiggles@ca.ibm.com

1. EARLY YEARS

I don't remember a time in my IBM career when CAS and CASCON did not exist. But I suppose that just means that I don't remember my first couple years at the IBM Toronto Lab in great detail. CAS and CASCON began about two years after I joined the IBM Toronto Lab as a full-time employee. For the earliest CASCON conferences, I remember making the trek from the 1150 Eglinton building over to the Ontario Science Centre to see the demonstration booths and sit in the giant lecture theatre to listen to the general session talks. There were breakout sessions, too, that took place in small rooms that were a challenge to find, scattered throughout the Science Centre. In following years, CASCON was a walk or shuttle ride away at the Four Seasons hotel near the lab, or at what is now the Radisson hotel on the east side of the Don Valley Parkway.

I really enjoyed the atmosphere at CASCON in those early years. The academic talks and interesting demonstrations were a nice change from my everyday tasks. I volunteered in different roles to be part the CASCON team. In particular, I recall having the task to attend paper presentations to take notes and write summaries to be shared with others who couldn't attend. I enjoyed being part of the team that was bringing new ideas to the Lab. I still have a paper sculpture of the Toronto skyline that was given to me by Jacob Slonim as a thank you gift for helping with the 1995 CASCON conference. To this day, that framed sculpture with its inscription occupies a place of pride in my home office.

Experience from my volunteer work at CASCON lead to me naively accept an offer to be program co-chair and local arrangements chair of a conference. The conference was called the "IEEE Symposium on the Assessment of Software Tools" and it took place in downtown Toronto in May of 1996. I really had gotten in over my head when I took on that work, and it was only by the generous and gracious support of CAS, and many people whom I had met through CAS, that the symposium avoided disaster. It was a crash course in all the details involved in running a conference: from forming a program committee, to setting and collecting registration fees, to picking the coffee break menus.

In early 2000 I was offered a surprise assignment. My manger asked me if I would like to become the CAS operations manager. Even better than that, the idea was that I would become the Head of CAS at the Toronto Lab when Gabby Silberman returned to New York after his assignment in Toronto ended! Although I was caught by surprise by the offer, I didn't hesitate at all and jumped at the chance to have a position where working with CAS would be my main responsibility and not just a volunteer activity when I had time to work on it. It was a huge responsibility and meant learning an entire new set of skills, but I was thrilled to have the chance.

Beginning in the spring and summer of 2000, I worked to learn all the details of the processes related to CAS visiting scientists and fellowship students. I also began recruiting new CAS research associates who were to become the glue to keep the academic projects connected with IBM product work. Having a strong staff of was going to be key to creating quality research projects that would matter to IBM product development teams. To keep CAS visibility up, I continued the work to set up CAS talks that would give the professors and fellowship students the opportunity to talk about their work in front of an IBM audience. And I usually brought along bags of cookies as an extra incentive to attend the events!

2. A NEW HOME

At the time I joined CAS, the Toronto Lab was going through a major change. The Toronto Lab had long been based in buildings at or near the corner of Don Mills and Eglinton in the city of Toronto, but these traditional facilities weren't ideal for the evolving work of software development that was becoming increasingly collaborative. Renovations were needed or a new home for the Lab was needed. The proposal to build a brand new Toronto Lab building in Markham was approved and the site on Warden Avenue just north of the 407 highway chosen.

The design of the new Lab was strongly influenced by the "dot com" boom that was taking place and the related battle to attract and keep top new technical talent. The idea was to move away from rigid, hierarchical office structure and introduce open concept design. I really wanted to take advantage of this desire for change and use it to create something special in the new are that would become the home for CAS. I worked closely with the Lab's facility manger, Steve Wright, and Turid Horgen, a professor of architecture from MIT on the design of the CAS area in the new Lab. We ran workshops with CAS staff, professors, and students to learn what people wanted and what they would use in the new space. The buckets of Lego used in the exercises are still likely to be found somewhere in a CAS storage room.

A result of the workshops was a CAS area that was noticeably more open than most of the other areas of the Lab. We didn't completely avoid using the new "super cubicles" that filled the rest of the Lab, but CAS did have a very flexible open area where fellowship student could meet and work together. My favourite parts of the CAS area were the two large tables – one near the entrance, and one near the staff offices – where people could gather for a discussion or just sit and eat their lunch. Sadly, space constraints in the Lab over the years eroded the open design of the CAS area and now it is filled with cubicles and most of the collaborative spaces are gone. But trends towards open areas for collaboration are picking up again, so.

3. THE CAS PROCESS

Moving to CAS from a product development team, there was a noticeable difference between the workings of CAS in the way that that CAS activities are driven by the calendar. Timing of activities within a product development team will vary depending on whether a new product release is targeted for March or June or October or whatever. The release dates become the anchor dates used in planning other activities. Also, product release dates can be reset and moved by a couple of months as business pressures require, but in CAS once you set the date for CASCON that's it – a schedule slip was not an option!

For CAS, the CASCON conference that takes place in late October or early November represented both the end and the start of the year-long process. As the beginning of the process, CASCON was an important event for forming new research project proposals that could be proposed for funding for the coming year. New ideas were commonly presented in workshops, or demonstrated in an exhibit, and then discussed in detail over lunch, or in the hallways during coffee breaks between presentations. Given the way that the CASCON conference brought together both academics and software development professionals, it provided an ideal opportunity to brainstorm and develop ideas.

New research proposals and proposals for projects seeking continuing funding were due a few weeks after CASCON. When all the proposals were in, the discussions began to choose the project that would receiving funding and the amounts they would receive. Thankfully, I was not at all alone in this work. I was extremely fortunate to have a staff of top notch CAS Research Associates, each with their own areas of expertise in which they were themselves considered to be influential. Each associate would work with academics and Lab developers to create their slate of proposals. Of course everyone was a fan of their own projects, but I recall that the discussions were always respectful and constructive. In the end, we ranked the projects and came out with an accepted list that we could all agree on and that fit within our budget.

The budget of CAS within the Toronto Lab was something that I always had to be aware of, but I had great support from the Lab Director of the time, Hershel Harris, who understood the goals of CAS and the many benefits it brought to the Toronto Lab. CAS always received the funds it needed to run its programs and I was very grateful for that.

Funding letters were prepared and mailed (yes, real postal mail with stamps and all!) to the professors and students to let them know about what funding they would receive. Funding was always in the form of CAS Fellowships that were awarded to specific students who would work on the project. The students were expected to spend time at the Toronto Lab during the late spring or summer. Preparing for the arrival of the students took a lot of effort, but again I was fortunate to have a wonderful, experienced staff who knew exactly what to do when it came guiding the students through the details of getting lab access, setting up their workstation, submitting and recovering living expenses, and finding their Lab product teams.

As summer came to an end, student returned to their university campuses and the attention of the CAS team focussed on the coming CASCON conference. All the funded research projects were expected to create papers, workshops, or breakout talks that described the outputs of the past year and submit them to CASCON. From this perspective, CASCON served as a showcase of the many varied joint research projects it had funded in the past year.

When Gabby Silberman left the Toronto Lab to return to New York, he took the assignment to spread the CAS model that had been pioneered at the Toronto Lab to other IBM locations around the world. As the "established" CAS location, the staff of CAS in Toronto provided the support needed to help other IBM development laboratory locations understand the CAS model and put it into practice. Some locations were more open to the model than others. If a lab had an established program to connect with academic institutions, establishing the CAS model was difficult. I recall visiting the IBM Hursley Lab and although they listened to what CAS had to offer, they preferred to stay with what was working for them. But for lab without such a program, Dublin being a very good example, the CAS model offered a tested way forward.

I had the opportunity to travel and speak at an IEEE conference on engineering management to present the paper that I had written to tell the story of the CAS model for bringing academic research results directly to product development [1]. The CAS story and its model were very well received. It was clear to me that IBM had developed a model for connecting the academic and product development communities that was unique within the industry.

Related to CASCON, an achievement of the CAS team during my time was to make all CASCON proceedings available online through the ACM Digital Library. It was not always the case that the important research done by CAS and its professors, students, and IBM software developers was easily found when other researchers performed literature searches. In order to be sure to be found and cited, CASCON proceedings had to be included in a major digital library. CAS staff worked with the ACM Digital Library team who were very helpful. Recent CASCON proceedings were already in digital form and were not difficult to add. But the earliest CASCON proceeding were only available in printed form, and so a copy of each early year's proceedings was sacrificed to be sliced apart and scanned into digital form. And so now even the papers from the very earliest CASCON conferences can be easily accessed online.

4. A PERSONAL VIEW

Personally, the work I did with CAS – both before I joined the CAS staff and while I was a CAS staffer –played a significant role in my career development. When I began, my professional work and network contacts were centered on what went on within the IBM Toronto Lab. CAS connected me with people and their work that broadened my network and the pool of ideas I could draw from. I especially enjoyed my trips to visit university campuses across Canada. I was able to tell the story of CAS and the IBM Toronto Lab and in return I learned about the ground breaking research that

As the manager of CAS, I was able to significantly grow my management skills. I was a member of the Lab's senior management team, the team of managers who reported directly to Hershel Harris, the Lab Director. All the other managers on the team lead very large organizations, most numbering in the hundreds, but CAS had about a dozen people. At our regular senior management team meetings, I felt a little bit like the representatives of Malta must feel at an EU meeting! Being a member of the Lab's senior management team gave me an entirely different view on the organization had been a part of for so many years. I learned from the others about how to handle the many difficult situations that inevitably occur in an organization so large. Personnel issues, real estate issues, and financial issues that were totally foreign to me were presented, discussed, and dealt with on a regular basis. Being part of that team was an amazing education that complemented the technical skills I was developing in other aspects of my work with CAS.

I knew that my assignment as Head of CAS was only intended to last a few years, so eventually the time came for me to leave CAS and move on. After about four years with working with all product development areas within the Lab, I had gained a much broader perspective of the opportunities available to me. I decided to join one of the new start up areas of the Lab that had been formed from an acquisition. The technical and management growth that CAS had supported served me well in my new position. I am no longer in management; a few years after leaving CAS I switched my career track to become a technical staff member. Now I am on the receiving end of the outputs of CAS research and continue to stay in contact with CAS and its programs.

REFERENCES

[1] Wigglesworth, J. Bringing academic research directly to development: IBM's Centers for Advanced Studies. *Proceedings IEEE International Engineering Management Conference (IEMC 2002).* Vol. 2, pp. 886-870 (2002)

All I Really Need to Know (About Work), I Learned in CAS

Kelly Lyons
Faculty of Information
University of Toronto
140 St. George St., Toronto, Canada
1-416-946-3839
kelly.lyons@utoronto.ca

ABSTRACT

I am in the enviable position of having held almost every possible role in CAS during my career. I started as a CAS Fellowship student in 1991 and became a CAS Research Staff Member when I graduated with my Ph.D. in 1994. In 1999, I held a development manager role and worked as the development contact for CAS researchers before serving as the CAS Director from 2004 to 2007. When I transitioned to academia, I was named a CAS Faculty Fellow. My time in CAS starting in 1991 prepared me well for all aspects of my career. In this chapter, I identify the ways in which being part of CAS taught me everything I need to know about work. In describing what I learned and how, I demonstrate the lasting impact and influence CAS has had on numerous students, faculty members, and IBM developers just like me.

Keywords

CAS, lessons, future career.

1. INTRODUCTION

There is a well-known book by Robert Fulghum entitled, “All I Really Need to Know I Learned in Kindergarten” [3]. Some of the key things that Fulghum identifies include sharing, being kind to one another, cleaning up after oneself, and balancing work, play, and learning. I argue that, while kindergarten may prepare one well for life, it is participation in CAS that prepares one well for work and a fulfilling career. The original CAS mission was, “to become a world-class applied research centre that facilitates the transfer of advanced research into strategic products.” [11]. In the late 1990s, the CAS mission became: to facilitate the exchange of academic research knowledge and real world industry challenges towards enhancing IBM products, processes and services. This mission is achieved through three goals:

- Establish IBM as the partner and employer of choice for top students as they learn and develop skills to create the technology of the future
- Build and foster relationships among researchers, funding agencies, IBM and customers
- Expose IBM developers to current research directions, and identify new and emerging technology issues for academic research

Students in CAS develop skills and expertise in working with industry people, positioning research in real-world contexts, and building and maintaining relationships. CAS Research Staff Members learn how to lead projects and must understand how different groups and people are motivated in order to work across academic and industry boundaries. IBM developers and development managers learn how to conduct research and apply research findings within products. Academic researchers who serve as CAS Faculty Fellows learn about the challenging problems facing IBM and its customers while developing an understanding of the pressures associated with delivering products on a schedule. CAS Directors are given the opportunity to set a vision and objectives for an important organization and mobilize resources, build relationships, and develop programs to achieve that vision. I feel very privileged to have learned and gained experience in all of these areas. In this chapter, I highlight a few specific examples of important lessons from my time in CAS that have helped me in my career.

2. WHAT I LEARNED IN CAS

My early experiences in CAS greatly influenced and shaped my future career and my values. Of the many lessons I learned in CAS, I have selected a few that I still practise today.

2.1 Articulate a vision and make decisions in line with that vision

As CAS students in the early 1990s, we were given responsibilities in the organization of CASCON. Looking back at that time, I realize now that, implicitly, we were implementing our tasks in line with the mission and vision of CAS and CASCON. At the time, I had not yet been explicitly taught how to articulate a mission and work with the mission and goals in mind.

Soon after I was hired (by Jacob Slonim, then Head of CAS) as a Research Staff Member (RSM) in 1994, Jacob asked me to travel to Montreal and Ottawa to meet with faculty members and determine which projects we should pursue in CAS. I was concerned that my limited experience would make it difficult for me to make decisions and so I enquired as to how I would know what to do when I was there. He told me that as long I understood the basic principles of what we were trying to do in CAS and made decisions in line with those principles, I would do the right thing. This was the first time I was explicitly told that I should consider my mission or principles and make decisions accordingly.

I was able to put this CAS lesson to the ultimate test when I became the Head of CAS in 2004. I met with the then IBM Toronto Lab Director, Hershel Harris, and he told me that I had been given a great opportunity to run an organization like CAS. It meant that I would be able to take the CAS mission and goals and make use of the resources available to me to achieve the mission and goals. I was immediately taken back to the lesson

Jacob Slonim had imparted on me ten years earlier. I understood the importance of having a goal in mind but also the satisfaction it brings to know why you are doing what you are doing and the ability to prioritize based on a well-defined, explicit mission or set of principles.

This important lesson has stayed with me. In work and life, I strive to make my principles and mission explicit and then make decisions with my goals in mind. I also teach this to my students. In each course and in my capacity as an advisor, I ask students to reflect on their own specific learning outcomes and goals and remind them to reflect on their objectives when making choices throughout the course or in their careers.

2.2 Build and Sustain Relationships

Inherently, I always knew that people and relationships were important (in fact, this is something we learn in kindergarten: "Without realizing it, we fill important places in each other's lives. … Good people who are always 'there,' who can be relied upon in small, important ways. People who teach us, bless us, encourage us, support us, uplift us in the dailiness of life." [3]). However, it wasn't until I was a CAS student that I experienced the ways in which strengths of relationships have an impact on the quality of work outcomes.

As a CAS student, I was a member of the Consortium for Research in Distributed Systems (CORDS) which was focused on the development, operation, and management of distributed applications [10]. During the summer of 1992, several CAS students worked together on the sixth floor of 895 Don Mills Road in one of the IBM Toronto Lab buildings. At the start of summer, Jacob charged us with building a prototype to develop a sample distributed application. We came from different backgrounds in computer science and our individual thesis research was extremely varied yet we worked together to develop a single prototype by adding mechanisms to the existing CORDS middleware, and integrating our own specific research into the prototype system. Through this process which was a difficult task, we became friends, socializing together and supporting each other in our individual research challenges. We dubbed our project, Operation Jumpstart, and not only produced a solid prototype but wrote and published a paper reporting on the project and the results [1]. The relationships we built during that summer enabled us to create something of value together. Furthermore, my interactions with fellow graduate students in CAS taught me that assembling a team of graduate students to work together on a shared goal enriches student research and relationship experiences and also enables research outcomes on a significant scale.

This book itself is an example of how strong relationships built over years in CAS can make great things happen. Everyone is busy with their own work commitments, yet when the editors of this book invited people to submit chapters, many people jumped at the chance. We want to create things together because of a strong mutual respect and desire to help one another.

The strength of relationships can often help when making difficult decisions. One of the unfortunate themes of my time as the Head of CAS was the reality of cost cuts. One year, the CAS team looked at all of our programs in light of our mission, goals, and available resources. We identified several ways in which we could best accomplish our goals while dealing with a decrease in financial resources. Building on the strong relationships we had with the CAS Visiting Scientists, I spoke with a group of key individuals (an ad-hoc advisory committee of sorts) and asked for their input and guidance. In the end, we decided to reduce the funding to the Visiting Scientist program, a decision that would have been very difficult to make and communicate without the strong relationships we had nurtured over the years.

2.3 Articulate Value

When I was the Head of CAS, I mistakenly expressed to a marketing person that CAS so was wonderful, we didn't need to market it. It turns out that I was wrong and he was correct when he told me that excellent services and products need to be marketed as much as or more than weak ones. Until then, I resisted quantifying the value of CAS and even posted a sign on my cubicle door at 8200 Warden Ave. that said, "Not everything that counts can be counted; not everything that can be counted counts." There were two times when I was the Head of CAS that we worked to formally articulate the value of CAS: when writing our nomination for the Leo Derikx NSERC Synergy Award for Innovation; and, when I was asked to justify the value of CAS to a new vice president after an IBM reorganization.

Both times, the CAS team worked together, bringing in others as needed to gather information, analyze data, reflect on past activities, and identify creative ways to articulate CAS value. This work brought the team together and we felt even prouder to work in such an incredible organization. We all intuitively understood the tremendous value of CAS but our exercise in quantifying and making that value explicit was an extremely satisfying and pride-inducing activity.

2.4 Honour and Document History

This book, assembled as a way to recognize 25 years of CAS is an example of a way in which we can honour and document our history. The CAS Model was celebrated in two issues of the IBM Systems Journal: Vol. 33, No. 3 in 1994 and Vol. 36, No. 4 in 1997. Each issue presents one article documenting the CAS model itself along with several articles reporting on key CAS projects and results. These documents are examples of how we can preserve the legacy of organizations such as CAS.

In 2005, under the leadership of Hausi Müller, CAS celebrated the pioneers of computing in Canada [8]. Honourees were selected using the following criteria: having spent a substantial part of their career at a Canadian University, contributing significantly to computing science education and research, and having received a PhD degree in 1972 or earlier. In total, 92 pioneers were honoured at a gala dinner held at CASCON in 2005. There was also a panel presentation in which pioneers reminisced about the past and made predictions about the future. For all of the pioneers, we documented their major achievements, memorable experiences, and words of wisdom[1]. Figure 1 shows a picture of the pioneers who were able to attend the gala dinner and dance event at CASCON 2005.

[1] Pioneers of Computing 2005: Honouring those who Influenced the History of Computing in Canada http://individual.utoronto.ca/klyons/files/pioneers.pdf

Figure 1: Canadian Pioneers of Computing honoured at CASCON in 2005

In 2010, we celebrated the 20th anniversary of CAS and CASCON. This milestone was marked in several ways including proceedings featuring fifteen past papers in a CASCON First Decade High Impact Papers proceedings [7] and a "vintage demos" showcase. Figure 2 shows me with friend and colleague, Wendy Powley, demonstrating our 1992-1994 graph layout demo called "GLAD". We were not able to compile our code from nearly a decade earlier but we were able to present screen shots and display some of the vintage CASCON bags from the early years.

It is possible to find ways to use research tools to help document the past. Inspired by the 20th anniversary of the CAS conference (CASCON) in 2010, my then doctoral student, Zack Hayat and I used Social Network Analysis (SNA) to characterize interactions in CAS by analysing co-authorship data over 19 years of CASCON (a total of 657 papers and 1101 authors) [5]. SNA provides a structural view of the CASCON co-authorship network as well as analysis of individual actors (authors) and their place in the network. Our analysis indicates that, in the CASCON community, social capital is maintained by the connections (cohesion) that exist among its members. While the average structural holes ratio in the CASCON community has decreased over 19 years, the cohesion of the network has increased.

Figure 2: Kelly Lyons and Wendy Powley pose beside their "Vintage Demo" celebrating 20 years of CASCON in 2010

Looking back, I have missed some opportunities to honour and document important events throughout my career. Putting these events and related artifacts together takes a great deal of time and effort but the lasting effects are priceless.

2.5 Have Fun and Make Friends

During my time at CAS I made lifelong friendships. This is partly because we worked together but also because we socialized together. Since my days as a CAS student, the CAS community has played soccer together, often ending up at someone's house (usually mine) to eat and share stories of the games.

One of the CAS traditions was an annual picnic where students, faculty members, CAS researchers and staff, and IBM developers took a day together for food and games. During my time as Head of CAS, budget constraints and new policies limited our ability to buy food for non-IBM employees, but we continued the tradition by making the CAS picnic a potluck affair.

The CAS team had favourite restaurants such as the Armenian Kitchen and Noodle Delight near the Don Mills Lab and Milestones near 8200 Warden Avenue. Each year, CASCON brings many old friends together again and the social events around CASCON are as important as the technical sessions.

I believe that one of the reasons that strong friendships are built through CAS is because we all believe so strongly in the benefit and value of CAS. We are united in our passion for the organization that is CAS. When the CAS partnerships won the Leo Derikx category of the NSERC Synergy Awards for Innovation in 2006, the awards ceremony was in Winnipeg. Several Chairs of Computer Science Departments travelled with me to Winnipeg, each finding their own funds and taking time out of their busy schedules to be there to celebrate together. We wanted to be together, to celebrate together winning an award for an organization that we all knew was so deserving.

Figure 3 shows the smiling faces of the group that travelled to Winnipeg that year (left to right: Pat Martin, Queen's University; Jacob Slonim, then Dalhousie University; Larry Achtemichuk, IBM Toronto Lab Director at the time CAS was founded, retired; Hausi Müller, University of Victoria; Kelly Lyons, then Head of CAS; Mike Bauer, University of Western Ontario; Garth Issett, IBM Canada; Johnny Wong, University of Waterloo; and, Craig Boutilier, University of Toronto). I have remained close friends with many of the individuals in this photo. We felt tremendous pride that day as we shared stories of CAS and the significant benefit that it has brought to IBM, universities, and Canada. Our friendship was strengthened through our mutual appreciation for the CAS model.

The friendships I made in CAS continue to this day. The CAS experience taught me the importance of building friendships with work colleagues. As a result, since joining academia, I have built new friendships with my colleagues at the University of Toronto. We find opportunities to go out together, enjoy potluck lunches, and identify common interests.

Figure 3: At the NSERC Awards Ceremony in Winnipeg, Manitoba after winning the NSERC Synergy Award, Leo Derikx Category for Innovation

3. CONCLUSION

Being part of CAS was a tremendous learning opportunity for me and others. In fact, the programs in CAS were specifically designed to provide those learning opportunities to the CAS community. In addition to the important lessons described above, my participation in CAS enabled me to remain connected to academia, setting the stage for an eventual career as a faculty member at the University of Toronto. I was able to learn about and contribute to a wide range of computer science research topics including: computational geometry and graph layout algorithms [6], distributed platforms and multimedia services [14], database management systems [2], distance learning systems [13], privacy systems [12], and service oriented computing [4]. Through CAS, I grew in a breadth of areas which gave me significant background and experience in methodologies, theories, knowledge, and application areas for my current interdisciplinary research focus in service science. And I'm just one of many; CAS has contributed to many people's careers.

4. ACKNOWLEDGMENTS

I wish to thank Hausi Müller for inviting me to contribute to this book and for everything he has taught me. I also wish to thank Jacob Slonim for being an incredible teacher and mentor. Most of what I do well today, I learned from him. I am grateful to Gabby Silberman for being a tremendous role model for what it means to be a kind, thoughtful, and strong leader. I also want to thank all of the CAS students, faculty members, and IBM colleagues who have been instrumental in helping me shape my career.

5. REFERENCES

[1] Attaluri, K., Bradshaw, D., Finnigan, P. J., Hinds, N., Kalantar, M., Lyons, K. A., Marshall, A. D., Pachl, J., & Tran, H. Operation jump start: A CORDS integration prototype using DCE. Published in the Proceedings of the 1993 CAS Conference Volume II, IBM Canada Ltd. Laboratory Centre for Advanced Studies (Toronto, ON, October, 1993), 621-636.

[2] Barbosa, D., Mendelzon, A., Keenleyside, J., & Lyons, K. (2002). ToXgene: An extensive template-based data generator for XML. Published in the Proceedings of the Fifth International Workshop on the Web and Databases (Madison, WI, June 2002), 49-54.

[3] Fulghum, R. All I Really Need To Know I Learned in Kindergarten. New York: Villard Books, 1988.

[4] Hawkins, J. L., Khusial, D., Lyons, K. A., McAllister, M. J., McKegney, R., McKenna, M. D., & Slonim, J. System and Method for Resolving Interdependencies between Heterogeneous Artifacts in a Software System, US Patent 8,707,260. Washington, DC: U.S. Patent and Trademark Office, Filed 2008; Awarded: April 2014

[5] Hayat, Z., & Lyons, K. The evolution of the CASCON community: A social network analysis. Published in the Proceedings of CASCON 2010 (Markham, ON, November 1-4, 2010), 1-12.

[6] Lyons, K. A. (1992). Cluster busting in anchored graph drawing. Published in the Proceedings of the 1992 CAS Conference, CASCON'92, IBM Canada Ltd. Laboratory Centre for Advanced Studies (Toronto, ON, November, 1992), 7-17.

[7] Müller, H. A., Ryman, A., & Kark, A. W. (Editors), CASCON First Decade High Impact Papers (CASCON'10), November 2010.

[8] Patrick, R. Canadian IT pioneers honoured at CASCON, IT World Canada, October 2005, Retrieved October 3, 2015 from: http://www.itworldcanada.com/article/canadian-it-pioneers-honoured-at-cascon/14181

[9] Perelgut, S. G., Silberman, G. M., Lyons, K. A., & Bennet, K. L. Overview: The Centre for Advanced Studies, IBM Systems Journal, 36(4), 474-488, 1997.

[10] Slonim, J., Bauer, M., & Larson, P. CORDS: status and directions. In Proceedings of the 1992 Conference of the Centre for Advanced Studies on Collaborative Research-Volume 2, IBM Canada Ltd. Laboratory Centre for Advanced Studies (Toronto, ON, October, 1992), 1-21, 1992.

[11] Slonim, J., Bauer, M. A., Larson, P. -A., Schwarz, J., Butler, C., Buss, E. B., & Sabbah, D. The Centre for Advanced Studies: A model for applied research and development, IBM Systems Journal, 33(3), 382-398, 1994.

[12] Smit, M., Lyons, K., McAllister, M., & Slonim, J. (2009). Detecting privacy infractions in applications: A framework and methodology. Published in the Proceedings of the IEEE International Symposium on Trust, Security and Privacy for Pervasive Applications (Macau SAR, P. R.China, October 12-14, 2009), 694-701.

[13] Smith, P. W., Lyons, K. A. & Bringelson, L. (2003). Multipoint video-over-IP seminar course. Published in the Proceedings of Canadian Association for Distance Learning (St. John's, Canada, June 2003).

[14] Wong, J. W., & Lyons, K. A., Evans, D., Velthuys, R.J., v.Bochmann, G., Dubois, E., Georganas, N.D., Neufeld, G., Ozsu, M.T., Brinskelle, J., Hafid, A., Hutchinson, N., Iglinski, P., Kerherve, B., Lamont, L., Makaroff, D., & Szafron, D. (1997). Enabling technology for distributed multimedia applications. IBM Systems Journal, 36(4), 489-507.

CAS: A Sustainable Industrial Research Collaboration Model

Joanna Ng
IBM Canada Software Laboratory, CAS Research
Markham, Ontario, Canada
jwng@ca.ibm.com

ABSTRACT

IBM formed the first CAS (Centre for Advanced Studies) twenty-five years ago in IBM Toronto Laboratory, with its original mission of *applied research transfer* from academic partners to IBM product development teams. A mutually beneficial CAS model was established, and formally published in IBM Systems Journal since its beginning. The CAS model has stood the test of time for the last twenty-five years. In the last decade, with confluences of transformations in the industry and within IBM as a company, coupled with fast emergence of multiple disruptive but impactful new technologies, in order to embrace the wave of a technological and industrial tsunami of changes, CAS has evolved from a model of academic research transfer to a model of *industrial research collaboration and partnership*. This paper summarizes how the CAS model has evolved over time to ensure research relevance and timely innovation impact amidst these major transformations. It also calls out the importance of establishing innovation metrics, in order to ensure effective communication of success. This paper also highlights a subset of successful outcomes as a result of these CAS evolutions.

Keywords

CAS, CASCON, industrial research, research collaboration, innovation, languages of innovation, innovation metrics

1. INTRODUCTION

IBM's Centre for Advanced Studies (CAS) was first established in 1990 in then IBM Toronto Software Lab as an applied research centre. Its primary goal is to "facilitate the transfer of research ideas into the various product groups" [1]. A CAS research operation model was established at the time, reflecting the understanding of the critical importance of interactions between academic research and software development communities. The original CAS model carefully considers the differences in the measures of success in academia and in IBM as an industrial partner, and designed a CAS operation model that focuses on the win-win of all parties.

I had the great honor to be appointed as the Head of Research for CAS from 2008 – 2015. When I first started my role in 2008, IBM Toronto Software Lab had just evolved to become IBM Canada Software Lab. It was also a time when forces of disruptive technologies such as cloud computing, mobile computing, social computing, big data analytics and Internet of Things started to emerge, drastically changing the industrial landscape and IBM as a company. In order to face such a time of massive changes, CAS has evolved beyond a model of *"applied research transfer"* into a model of *"industrial research collaboration and partnership"*.

In this industrial research collaboration model, IBM shares with CAS academic partners problem statements we receive from our enterprise customers, and IBM's business strategies and priorities. This shapes CAS research scope in all CAS initiatives; critical in ensuring our research relevance. It is also necessary in order to ensure our contributions and advancements out of our collaborative research effort have business and technological impact. In addition, finding an efficient approach to crisply articulate back to the business the return on research investment in CAS, in a manner that resonates, is of critical importance, as economics are getting tightened year over year.

The remainder of this paper is organized as follows. Section 2 re-captures the key elements of the classic CAS industrial collaborative research model. Section 3 reflects upon the major refinement of the CAS model in light of the vortex of emerging technologies that transform IBM from within and the industry at large. Section 4 highlights its successful outcomes. Section 5 concludes the paper.

2. THE CLASSIC CAS MODEL

The CAS model [1] is expressed as a set of fundamental CAS principles that operationalize the mission of *applied research transfer* from academic partners to IBM product development teams. A subset of these CAS principles are highlighted here, namely:

The principle of *Win-Win design* is focused on the selection of a research agenda that benefits and fits the business and academic interests of both IBM and academia.

The principle of *mission-driven projects* shapes research project selection criteria, focusing on research projects with direct innovation and business impact in the short-to-midrange timeframe.

The principle of *cooperative arrangement with experts and personal contact and networking* highlights the importance of research relationships built upon relevant expertise, with personal contacts and long term working relationship of collaboration history that builds trust.

The principle of *product development funding and leverage of matching funds* is designed for a low innovation and development cost from IBM with a high yield of research outcome by leveraging the matching of federal and provincial research fund matching, minimizing IBM's financial risk in organic innovation.

The principle of *focusing on prototypes* is designed to ensure that research outcomes from academic partnerships are implementable.

The classic CAS model also includes a set of CAS processes in identifying problems, defining the scope of research project, forming the collaborative research team over the established project scope.

A set of CAS initiatives was birthed since then. Centre for Advanced Studies Conference (CASCON) is one of these CAS initiatives. We are happy to be a part of its quarter-century anniversary celebration. Over time, it has evolved to become a premier, international industrial academic conference in computer science and software engineering [3]. CASCON proceedings are available in the ACM Digital Library and are indexed [3].

Changes are inevitable. In the last decade, IBM as a company has gone through major organizational changes. IBM Toronto Software Lab has evolved to become IBM Canada Software Lab, with multiple lab sites and a large collection of product portfolio with a wide spectrum of technologies.

In the last decade, multiple disruptive technologies have emerged as impactful forces that change the industrial landscape. Their confluence forms an industrial tsunami of transformations. When I was first appointed as the Head of Research for CAS in 2008, these disruptive technologies, namely, cloud computing, mobile computing, social computing, big data and analytics, internet of things, were just emerging and far from adoption maturity. Yet, we knew they were too critical to be ignored. CAS has since then evolved from a model of *academic research transfer* to a model of *industrial research collaboration and partnership,* in order for the entire CAS community together, from IBM practitioners, technologists and IBM product stakeholders, to CAS faculty academic partners, to work collaboratively as innovation and research partners, in order to produce research outcomes and thought leadership in these game-changing technologies.

3. KEY REFINEMENTS

In light of these major shifts in the industry, in technologies and within IBM as a company, several key refinements over the original CAS model had been put in place since 2008. These refinements are necessary in order to build a ***sustainable industrial research collaboration model*** that positions CAS Research's continual success going forward. These refinements also ensure CAS Research is able to bring in research-based thought leaderships that add value. These refinements are aimed to foster technology contributions that are significant to the history of technology and to the industry at large.

3.1. CAS Technology Incubation Lab

The CAS Technology Incubation Lab (TIL) was birthed in November 2008. It was formed out of the realization that software development after successful research outcome is absolutely critical in commercialization. Commercialization is key to the sustainability of research investment. TIL is missioned to produce implemented prototypes, with code quality and strength close to product-level implementation. Prototypes produced demonstrate the realization of innovation concepts, providing technology-adopters an actual hands-on iteration to try and to provide feedback in usability, bringing credibility to research outcomes that paper solutions cannot. TIL leverages government post-doctorate industrial matching funds to maximize software development capacity with minimized cost.

3.2. Research Portfolio by Technology Themes

Starting 2009, CAS research projects are categorized under major technology themes, grouped into larger research portfolio [7]. This was designed to discover how these projects relate to one another in providing solutions to a bigger technology space. When related projects are being abstracted to a higher level of consumptions, it opens up opportunity for more technology contributions. High scale, low touch component of IBM cloud was harvested into commercialization under this approach.

Laying out research scope per projects within a larger technology theme (e.g. Cloud Computing) enables us to identify technology gaps from within, leading to problem statements to formulate the next round of research projects within a technology theme.

Eight major technology themes and three major industrial solutions were established since 2009 [7]. It is interesting to call out an observation that IBM re-structured itself in the beginning of 2015, creating business units per these technology areas that we have called out in 2009.

3.3. Innovation Metrics in Four "P"s

Establishing innovation metrics that we can communicate efficiently in a manner that resonates among the business and the innovation community is critical in order to sustain the investment in the industrial research collaboration. It provides a very tangible expression of the CAS's win-win model.

The Four Languages of Innovation [2] has been established since 2013 as CAS Research's metric, expressed in units of measure in the number of patents filed, the number of publications and citation counts, the number of prototypes implemented and the number of productizations accomplished. Internally we call this the 4P metric model of innovation.

3.4. The Role of CAS Research Staff Member

The role of CAS Research Staff Member (RSM) has been expanded beyond the role of research coordinator into multiple additional roles such as research portfolio manager, subject matter expert as a technologist and architect of prototyped solutions, leading TIL to implement near-product level prototypes.

3.5. Position Paper Track in CASCON

Recognizing the fast speed and dynamics of these emerging technologies, we started a new "Position Paper Track" in CASCON 2013 in order to provide a time-stamped approach to capture thought leadership in positioning what may be too pre-mature for validated research results to be published as technical papers.

My personal thanks to Dr. Jenny Zou from Queen's University, Dr. Kenny Wong from University of Alberta, Dr. Hausi Müller from University of Victoria, and Dr. Jim R. Cordy from Queen's University, who are the pioneers in the formation of this position paper track, which we come to know today as "Emerging Technology Track" (ETT) in CASCON.

4. RESULTS OF INNOVATION

CAS has established a great heritage over the last quarter-century. The refinement of the CAS model, from a mindset of academic research transfer to a truly collaborative industrial research model, has since yielded fruitful outcomes. CAS is well positioned to take on another quarter-century, with continual evolution to ensure research relevance and timely innovation impact. The following is a highlight of some of its accomplishments.

4.1 People

The pillar of success for CAS's quarter century is the CAS community. This includes CAS Faculty Members and Visiting Scientists who go above and beyond what is required, time and time again, to make countless contributions to CAS's success.

CASCON's 25th Anniversary celebration is a testimony of such long-term commitment.

CAS Research's industrial research collaboration model also produces highly skilled personnel from CAS faculty students who are very much a part of the innovation journey. Not only they have proven their skills over a sustained period of time, they also gain skills in emerging technologies that are very difficult to find in the industry, making them prime candidates to hire into IBM.

4.2 Disruptive Innovation Outcome

With the establishment of CAS TIL and CAS Research portfolio, and a transition into an industrial research collaboration model, disruptive innovations have been led and birthed by the CAS Research Community. Critical product components in IBM cloud, called the High Scale, Low Touch cloud autonomic management is one of the poster-children of CAS's disruptive innovations. Other disruptive innovations birthed by CAS Research includes Social Relationships as a Service, Web Tasking, Talk to my Data, just to name a few. This is not intended to be an exhaustive list, but to highlight a glimpse of the innovation potential of CAS's "low-cost, high-yield" organic innovation model.

CAS Research has also started new research platforms, starting workshops in new research areas. Web Tasking [6], Smart Internet are a few examples [4].

4.3 Incremental Innovation Outcome

CAS Research continues to make fruitful contributions in incremental innovation outcome in strategic IBM product portfolios, including IBM Security products, IBM Next Generation Systems and Compilers, IBM Analytic Product suite, just to name a few.

4.4 Publications

Since the establishment of the innovation metrics in the Four languages of Innovation, there has been an exponential growth of publications under CAS Research, from ten publications in 2007 to the peak of one hundred and fifty-four publications in 2012, now an average of ninety publications annually.

CAS Research has also started to publish CAS Research books through Springer. We have so far published two CAS Research books under Springer: namely *"The Smart Internet"* [4], and *"The Personal Web"* [5]. With so many new technologies emerging, more books need to be published in the future to capture the impactful research outcome that CAS research as a community has contributed.

4.5 Patents

Patents have not been a focus item until the innovation metrics were established. Since 2012, a steady annual average of twelve patents have been filed by CAS Research.

5. CONCLUDING THOUGHTS

CAS Research has established a rich innovation and partnership heritage that is impactful for IBM as well as for the CAS academic community.

The CAS Model established since the beginning has proven its value over a quarter century. Refinements that are critical to CAS's continual success have been in place for CAS to embrace the next quarter century.

It has been my honor and privilege to serve this amazing CAS community in the last seven years. I want to take this opportunity to thank everyone for your commitments to CAS and wishing CAS all the successes in the future.

REFERENCES

[1] Slonim, J., Bauer, A., Larson, P.-A., Schwarz, J., Butler, C., Buss, E.B. and Sabbah, D. The Centre for Advanced Studies: A model for applied research and development. *IBM Systems Journal* 33(3):382-398 (Aug. 1994)

[2] Ng, J. The Four Languages of Innovation. (Apr. 2015) https://www-927.ibm.com/ibm/cas/canada/file/CAS_Research_The_Four_Languages_of_Innovation_April_2015_v2.pdf

[3] CASCON Wiki: https://en.wikipedia.org/wiki/IBM_CASCON

[4] Chignell, M., Cordy, J.R., Ng, J., and Yesha, Y. *The Smart Internet: Current Research and Future Applications*. LNCS 6400. Springer (Nov. 2010) http://www.springer.com/gp/book/9783642165986

[5] Chignell, M., Cordy, J.R., Kealey, R., Ng, J., and Yesha, Y. *The Personal Web.* LNCS 7855. Springer (May 2013). http://link.springer.com/book/10.1007/978-3-642-39995-4

[6] *First International Workshop on Personal Web Tasking (PWT 2013). IEEE 9th World Congress on Services (IEEE SERVICES 2013)*. IEEE. (Jun. 2013) http://www.servicescongress.org/2013/pwt.html

[7] CAS Research Technology Themes: https://www-927.ibm.com/ibm/cas/canada/research/theme.jsp

The Early Days of CAS

Personal Reflections on the Influence and Impact of CAS Data Visualization and Query Research

Arthur Ryman
IBM Canada Lab (Retired)
arthur.ryman@gmail.com

ABSTRACT

This article presents the personal reflections of the author on the impact and influence of data visualization and query research carried out in the Centre for Advanced Studies (CAS) at the IBM Toronto Lab. This research was a main thread of the Advanced Software Design Technology (ASDT) project which was one of the founding research projects at CAS. The article describes the creation of the ASDT project, CAS, and the first CAS conference (CASCON). It then traces the influence and impact of the data visualization and query research thread on IBM Rational products and industry standards including Open Services for Lifecycle Collaboration (OSLC) and W3C Linked Data. It concludes with some reflections on the impact of the late Prof. Alberto Mendelzon who was a key contributor to this research.

1. INTRODUCTION

I'd like to thank the organizers of this volume for inviting me to contribute my thoughts on the influence and impact of the IBM Centre for Advanced Studies (CAS) on the occasion of its 25th anniversary. I am especially grateful that they have requested that contributors document their personal experiences. As an early proponent and co-founder of CAS, I am obviously positively biased towards it. Nevertheless, I will attempt to be objective in my assessment the influence and impact that CAS and my research there has had.

1.1 Organization of this Article

The remainder of this article is organized chronologically. Section 2 covers the prelude to the creation of CAS. It describes the environment at the IBM Toronto Lab and the conditions that led to the creation of CAS. Section 3 describes my years in CAS as its Associate Head and the Principle Investigation of the Advanced Software Design Technology (ASDT) project. It also describes the origin of CASCON. Section 4 describes my return to product development and how my CAS experience impacted IBM products. Section 5 contains my concluding remarks.

2. PRELUDE TO CAS

I joined the IBM Toronto Lab in 1982. At that time, the Lab had around 300 employees and was mainly a subcontractor to IBM labs in the USA. The Lab was engaged in both hardware and software projects. Back then software was often regarded as something necessary in order to sell hardware. That began to change. Software was becoming a major source of revenue. The Lab leadership team was actively looking for new software missions and was very supportive of advanced technology (adtech) projects.

Prior to entering industry I had done several years of postdoctoral research so, naturally, I gravitated to the Lab adtech program. Some adtech projects involved only Lab developers while others, including mine, were collaborative efforts with university research partners. The adtech program established a strong culture of innovation and academic collaboration within the Lab and laid the foundations for CAS. The Lab adtech program was in many respects the beta version of CAS and it is neither easy nor useful to separate the two.

2.1 The ASDT Project

By 1989 I had set up the ASDT project. The goal of the ASDT project was to create new tools that would help developers understand complex software artifacts such as design specifications and source code. The ASDT project had several research threads and many wonderful academic collaborators. These included formalization of object-oriented design with Prof. David Lamb at Queens University, metaprogramming with Prof. Jim Cordy at Queens University, optimization of Prolog queries with Prof. Nigel Horspool at the University of Victoria, stereoscopic display of networks with Prof. Paul Milgram at the University of Toronto, and application of text markup languages to software artifacts with Prof. Frank Tompa at the University of Waterloo. I would have liked to describe them all, but space here is limited. I have therefore decided to limit my remarks to the data visualization and query thread and to tell the story of my collaboration with Prof. Alberto Mendelzon and his student Mariano Consens at the University of Toronto.

Within the Artificial Intelligence community, it was well known that certain kinds of knowledge could be conceptualized as directed, labeled graphs called semantic networks. The nodes of the graph represented things and the arcs between nodes represented relations between those things. Semantic networks were ideal for representing knowledge about many kinds of software artifacts. For example, a class hierarchy diagram is a semantic network in which the nodes are classes and the arcs are inheritance relations. The logic programming language Prolog could be used to represent and query semantic networks. Each arc of a semantic network could be encoded as a Prolog fact. The entire semantic network corresponded to a Prolog knowledge base which could be queried using Prolog rules.

The ASDT project was motivated by positive development experiences using Prolog to represent and query information about design specifications, source code, and code overlay maps. The vision of the ASDT project was to create tools that used semantic networks as a unifying framework for representing, querying, and visualizing knowledge about software artifacts. Queries arose in two contexts, namely as constraints and filters. A constraint is any rule that a semantic network is required to satisfy. For example, design rules can often be phrased as constraints. A developer would use a constraint query to check for violations to the design rules, and then take corrective action as required. A filter is a rule used to eliminate parts of a semantic network. Filters allow a developer to reduce the data volume to a manageable level so they can focus on just the relevant information. The ability to filter a semantic network is crucial for effective visualization since real-world semantic networks contain huge amounts of data.

I had become an Adjunct Professor at the University of Waterloo, working with Prof. Kelly Booth in the Graphics Lab. I planned my visits to the university to coincide with departmental seminars so that I could understand the state-of-the-art in graphics research. On one visit I attended a talk by Alberto Mendelzon where he presented his research on GraphLog, his visual query language. GraphLog was a visual representation of Datalog, a logic programming language closely related to Prolog. Datalog was the subject of active research within the database community. GraphLog provided a way to formulate complex Datalog queries using the visual metaphor of matching subgraphs of a semantic network. GraphLog had built-in path operators which let you express recursive queries, a task which required a lot of programming skill when using Prolog.

Alberto's presentation had a huge impact on me. First, I recognized that Datalog was more natural than Prolog for representing data. Second, I believed that GraphLog was so intuitive that software developers could learn it with minimal effort. In contrast, Prolog rules, although powerful, required a lot of effort to write. However, I was concerned that GraphLog might lack enough expressive power for the kind of queries that arose in software development. I approached Alberto after the talk and I showed him my Prolog rules. He was confident that they could be expressed in GraphLog. Alberto arranged to have Mariano Consens, his PhD student at the time, translate them. Mariano was successful and both he and Alberto began to participate in the ASDT project.

2.2 Lobbying for CAS

In 1989 IBM had several large research labs, such as T.J. Watson in Yorktown Heights, and many more so-called Scientific Centers around the world, but none in Canada. I felt strongly that Canada should also have a formal research role within IBM. Several of my colleagues and I began to advocate for the creation of a Scientific Center in Canada. We argued that Toronto was the ideal location for a Scientific Center because it was surrounded major universities that had strong Computer Science programs.

Fortunately, this idea was already in the air. The Lab's leadership team had established a University Advisory Board with representatives from many Canadian universities. One of their recommendations was that the Lab should establish a permanent, formal program for collaborative university research. The Lab's leadership team accepted this recommendation and created CAS in 1990.

3. THE CAS YEARS

Jacob Slonim was appointed as Head of the newly established CAS. He had recently joined the Lab and had excellent relations with the academic community. Jacob asked me to serve as Associate Head and to continue as Principle Investigator of the ASDT project. All of the other adtech projects were also brought under the management of CAS.

Jacob established a streamlined model for funding collaborative research. This model enabled me to run the ASDT project for several more years during which I had the opportunity to work with many great researchers and help advance the careers of many talented graduate students, some of whom were later hired by IBM. I recall that when people asked me what I did, I told them that I had "The best job in IBM."

Mariano, Alberto, and I published the GraphLog work in 1991 at the 14th International Conference on Software Engineering (ICSE)[2] and at the first CASCON[3]. A comprehensive survey of our graphical query and visualization work was published in 1994 in the IBM Systems Journal[1].

3.1 Immediate Product Impact

The ASDT project had one immediate product impact. The ASDT project included software that could transform C++ source code into Prolog facts. The Lab owned the C++ compiler mission for IBM and developers on the compiler team liked the idea of using Prolog to represent software artifacts. They added an option to the XLC++ compiler to generate Prolog facts directly from its internal parse tree. This information was later used for query and visualization in an Integrated Development Environment (IDE) for C++.

The other components of the ASDT project, especially those for graphical visualization, were considered to be too exotic to be picked up by products at that time. Both GraphLog and 3D network visualization required fast graphical rendering chips which were only available in advanced UNIX workstations such as the IBM RS6000. Typical desktop PCs were inadequate for this purpose. However, many ASDT concepts influenced me later in my career and had a much larger impact on IBM products and industry standards.

3.2 Standards

Jacob actively encouraged all CAS members to participate in standards bodies. I joined an ISO committee on software engineering standards. Regrettably, this particular committee had no direct impact on the Lab, but it was a great learning experience for me and helped me to be an effective contributor in future standards work.

3.3 CASCON

CASCON, the annual CAS research conference, is arguably one of the most influential aspects of CAS. I'd therefore now like to tell the story of how CASCON got started. As I previously mentioned, the Lab had established a University Advisory Board. Some of its members advised us on how to run CAS. During one meeting Kelly Booth suggested that we hold an annual conference. A few days later Jacob asked me to plan and manage our first event. I started to send out emails and needed a subject line. Around that time science fiction conferences were very popular and people referred to these simply as "Cons." So I started using the subject line "CASCON" as a placeholder until we picked an official name for the conference. I guess that people thought CASCON was the official name, or maybe everyone liked it. In any case the name stuck.

The first CASCON was held in the Lab. It was a modest affair with around 100 attendees. Everyone knew everyone and there was a lot of hallway conversations and networking. As CAS grew, we needed more space and started to hold CASCON in external conference facilities. Fortunately, the increase in size did not have a negative effect on the way attendees interacted with each other. CASCON continued to be a great place to renew old acquaintances and to make new ones. Whenever I attend CASCON I am struck by the deep sense of community among the attendees. The fact that CASCON will be held for the 25th time this year is a testament to the special place it holds within the Canadian and international software communities.

4. AFTER CAS

Jacob's personnel policy for CAS was that Lab members should rotate through it to facilitate technology transfer and to bring in fresh blood, so after around four years as Associate Head I moved back into product development. My immediate goal was to transfer some ideas from the ASDT project into IBM products.

4.1 Productization Efforts

As mentioned above, the ASDT project used Prolog, and later Datalog, to represent software artifacts. There were two main reasons for this technology choice. First, both Prolog and Datalog do not require schemas making it very easy to add new types of data. Today we would refer to these as NoSQL technologies. Second, the kind of queries that arise for software artifacts often involve recursion, e.g. find all functions that directly or indirectly call a given function. Recursive queries be expressed in both Prolog and Datalog, but were impossible to express in early versions of SQL. However, by 1995 IBM DB2 had added support for recursion.

I believe it is very plausible that academic research on Datalog, including that done by Alberto Mendelzon, influenced the DB2 team to add recursion to SQL. In any case, I made the decision to migrate the ASDT project onto DB2 since it was a strategic IBM product, thereby eliminating one potential barrier to adoption. However, there still remained the problem of graphically rendering the semantic networks. Typical PCs were still not powerful enough and advanced RS6000 workstations were too expensive for the average developer. ASDT was simply not consumable at that time.

Investment in further ASDT development stopped and I returned to mainstream product development projects. I remained convinced that the basic idea of querying and visualizing semantic networks had great potential for software developers. However, I also learned that in order to move research results into products, the enabling technology had to have a very low barrier to adoption.

4.2 The Web, Java, and Eclipse

My CAS experience had a significant impact on my future career. Through my inability to transfer ASDT results into products, I had learned to be a much better judge of what would succeed in the marketplace. In particular, I learned the importance of simplicity and consumability. If a technology was not consumable, it was doomed to failure.

CAS had also opened new doors for me. My role as Associate Head had helped me get elected to the IBM Academy of Technology where I was exposed to the best technical minds in IBM. There I had early exposure to many emerging technologies. I recognized the great potential in the Web and Java to displace the incumbent technologies and as a consequence was able to propose and then lead development tool projects for those technologies. Both the Web and Java were simpler alternatives to the prevailing distributed computing (e.g. CORBA) and object-oriented technologies (e.g. C++, Smalltalk), and I had learned that simplicity was often the overriding factor for market acceptance. I also recognized the power behind the Open Source movement and was able to propose and lead a major project at Eclipse (the Web Tools Platform project).

My earlier less-than-impactful CAS experience in an ISO standards committee prepared me to be successful in Web and Java standards at W3C and the Java Community Process (JCP).

4.3 Jazz, OSLC, and Linked Data

By 2008 the focus of work on development tools had clearly shifted from the desktop to the Web. This movement was enabled by the pervasiveness of the Internet and was driven by the need for globally distributed development teams to collaborate. My division, Rational, embarked on the creation of a new server-based tool platform called Jazz and a new integration architecture called Open Services for Lifecycle Collaboration (OSLC).

My CAS experience turned out to profoundly influence my involvement with OSLC. Martin Nally, then the Chief Technology Officer of Rational, had proposed that OSLC use a Semantic Web technology called Resource Description Framework (RDF). Martin had proposed RDF specifically because it did not require a schema and was therefore very suited to integrating diverse types of information about software artifacts. This was exactly one of the motives for using Prolog and Datalog in ASDT.

I had in fact previously been exposed to RDF through my standards work at W3C but had not regarded it as very compelling at the time. However, the RDF technology stack had since evolved and now included good databases known as triples stores and the powerful query language SPARQL. Remarkably, SPARQL was closely related to Datalog so it resonated with me. My positive experiences with Datalog in the ASDT project influenced me to reverse my opinion of RDF. I recognized in SPARQL the potential to express the kind of queries that typically arose in software artifacts and advocated for its adoption as a query language for OSLC.

The architectural pattern of RDF messages exchanged in HTTP requests was referred to at W3C as Linked Data. This pattern was precisely what Martin was proposing as the basis for OSLC. I became a strong proponent of the adoption of Linked Data, contributed to several OSLC specifications, and implemented OSLC support in several IBM products. One of the OSLC specifications that I contributed to was Resource Shapes which allowed you to easily express constraints on RDF graphs. This was one of the main use cases for queries in ASDT.

OSLC specifications are currently being standardized at W3C and OASIS. I contributed the OSLC Resource Shape specification to W3C[4]. This led to the formation of W3C Data Shapes Working Group where I represented IBM. After I retired from IBM, I continued to participate in this Working Group as an Invited Expert.

There is also an interesting personal CAS connection to this later work. Mariano Consens earned his PhD and went on to become a professor at the University of Toronto and a CAS Faculty Fellow. He later independently did research on Linked Data, reflecting the natural progression from Datalog to SPARQL as a topic of research. Mariano supervised John Liu who developed XML tools for Eclipse as a CAS MSc student and was later hired by the Lab. John then joined my development team to work on a SPARQL gateway server which became the core of the Jazz Reporting Service. So three generations of CAS researchers have had an impact on data query and visualization aspects of Rational products.

5. CONCLUSION

Did CAS data visualization and query research have a significant influence or impact on anything? I think so. It certainly had a very positive impact on my career and on the careers of many of the graduate students that contributed to it.

Did it have a significant impact on the research community? That is hard for me to judge, but there is some independent evidence that it did have a significant impact.

In 2010, as part of its 20th anniversary celebrations, CASCON included a “First Decade High Impact Paper Award”. A committee of 14 experts reviewed the 425 papers that had been published in the first decade of CASCON. The committee selected 14 papers that were judged to have high academic and industrial impact. The ASDT 1991 CASCON data visualization and query paper[3] was among those selected for this special award.

I am confident in saying that, although it had limited early product impact, my CAS experience had significant influence on the adoption of Linked Data by Rational and on the development of OSLC specifications which are now becoming standards at W3C and OASIS.

In closing, I'd like to take this opportunity to say that I felt privileged to have known and worked with Alberto Mendelzon who in 2005 at the peak of his career was tragically taken from us by pancreatic cancer. The Latin American research community celebrates the influence of Alberto annually in the “Alberto Mendelzon International Workshop on Foundations of Data Management.” On this the 10th anniversary of his death, let his many CAS friends and colleagues also celebrate his life and work.

6. REFERENCES

[1] M. Consens, F. Eigler, M. Hasan, A. Mendelzon, E. Noik, A. Ryman, and D. Vista. Architecture and applications of the hy+ system. *IBM Systems Journal*, 33(3):458–476, 1994.

[2] M. Consens, A. Mendelzon, and A. Ryman. Visualizing and querying software structures. In *Proceedings of the 14th ICSE: International Conference on Software Engineering*, May 1991.

[3] M. Consens, A. Mendelzon, and A. Ryman. Visualizing and querying software structures. In *Proceedings of 1991 CASCON: The Centre for Advanced Studies Conference*, pages 17–32, Sept. 1991.

[4] A. Ryman. Resource shape 2.0. W3C Member Submission, World Wide Web Consortium, Feb. 2014.

The Improbability of CAS

Bill O'Farrell
IBM Linux on System Z Open Source
Ecosystem
IBM Canada Lab, Markham
billo@ca.ibm.com

ABSTRACT

In this chapter I present some personal reminiscences of CAS and CASCON, and my personal experience with the CAS model of low-risk high yield development and the training of Highly Qualified Personnel.

1. EARLY DAYS OF CAS

In preparation for this paper I have been reminiscing about what can sometimes seem like "the ancient of days." I was hired directly into CAS from Syracuse University in 1991. My formal offer was on letterhead that listed IBM's website – hold on, I'm lying. The world-wide-web had not yet arrived. In fact the letterhead listed IBM's cable and telex numbers (cable: INBUSMACH, telex: 06-966574); what the heck are those? Cellphones were rare, and we did email on VM/CMS green screen technology (which was actually 100% reliable).

The technology seems dated now, but the vision for the future embodied in CAS was vibrant and modern. CAS originated at a time when IBM was about to enter a financial crisis, and was said to be stodgy. How improbable. Upon starting with IBM, I learned that the CAS idea was primarily the brainchild of Lab director Larry Achtemichuk. He got the whole thing started in 1989, and brought on-board Jacob Slonim to provide leadership and vision to CAS, and that he did in spades. Jacob further developed the founding principle of working with leading University Professors and bringing their students to IBM as part of an assistantship program. Tied-in with various government grants available to academics, IBM's investment was often doubled or tripled and applied to research that directly benefited IBM's products. Jacob also pioneered the idea of professors and students (that is, non-IBMers) co-authoring IBM patents with IBMers. This was virtually unheard of at the time and was one of the times that Jacob had to wrestle with the prevailing bureaucracy to achieve something innovative. I personally benefited from this with several patents (as well as papers) done jointly with Academic partners, including several from the early days with Professor Eshrat Arjomandi from York University [8], [9], [15], [16], [17], [18], [19], [20], [21]. After Larry Achtemichuk stepped down, the new Lab Director, John Schwarz, helped carry on the CAS vision with renewed support for Jacob's vision, and (importantly!) renewed funding.

Among the first professors involved with CAS were Eshrat, who worked with me on parallelism, David Taylor, who researched distributed systems (the POET system), Alberto Mendelson, Toby Teorey, Michael Bauer, Hanan Lutfiyya, Irwin Pressman, Richard Holt, Guang Gao, Paul Ward, Yelena Yesha, Kostas Kontogiannis, Pat Martin, Bill Cowan, Stephen Cook (Turing Award winner), Charles Rackoff, Allan Borodin, John Mylopoulos, and many others.

I apologize for not remembering all of the names. I think all of these professors impacted IBM products and people, some a great deal. Both the length and the esteemed nature of the list show the level of interest CAS generated in the academic community.

Of course I most keenly remember Eshrat and her students, who worked with me on ABC++, which was eventually used in the concurrent administration feature of DB2 Parallel Edition. I also closely followed David Taylor, as his research was used to develop new products in the area of Autonomic Computing. IBMers that were involved in CAS at that time include Arthur Ryman (recently retired Chief Data Officer of Rational), Erich Buss, John Henshaw, Ivan Kalas, John Botsford, Patrick Finnigan (who retired after 31 years with IBM), Rich Helms, Jan Pachl, and Ron Holt. I believe it was Erich who told me how developers were measured by lines of code per week, and that it might go against me if I replaced a longer algorithm by a shorter, more efficient one. In fact, I never experienced anything like that at IBM. Far from stodgy, the IBM I have experienced over the years has been a crucible of innovation, and I think that CAS has been a key part of that in the Toronto Lab.

The Operations Manager at CAS when I came was John Maillard, after which came Nick Cooper. Polly McPherson was Jacob's assistant and Cindy Butler was administrator. One thing we all had to cope with – a by-product of CAS growth and shifting Lab priorities in a rapidly changing world – was frequent moving. In what seemed like a very short time, we were first in one tower at 895 Don Mills Road, then the next tower, then we moved to 844 Don Mills Road, then 1150 Eglinton. Not knowing that we were going to move, we even had a super high-speed fiber network installed in one tower of 895 Don Mills. I assume it's still there, buried in the plenum above the ceiling. Believe it or not, I even once won a small IBM award for my knowledge of plenums (the award was an IBM umbrella, which somehow seems ominously ironic considering the plenum is where all the water pipes are).

It was CAS which first brought the world-wide web to the Toronto Lab. We got an internet connection through the phone company, and I personally registered the ibm.ca domain. I owned ibm.ca for many, many years until someone at IBM actually noticed that it wasn't the IBM Company who owned it – it was some private individual named Bill O'Farrell. Lucky for them, that individual worked for IBM, and was happy to get rid of it.

In those early days we had just a few machines hooked up (no firewalls, no regulation), and our students were among the first in Canada to surf the world wide web. We developed our own website (no IBM guidelines then), and I poked around with some new websites that were called "search engines." One of them claimed to index an unbelievable huge number of web sites - over 70,000!

(Of course, most search engines now index billions). Nonetheless, CAS has always been on the forefront of tomorrow's technology.

2. STUDENTS ARE KEY TO CAS

One thing that Jacob impressed on us was the importance of the students. Today we use the acronym "HQP" – i.e. the training of highly qualified personnel. Here are a few IBM HQPs (in no particular order) that started as CAS students under my (at least partial) direction:

Ilene Seelemann, who is now on the cutting edge of cloud with the Bluemix Garage, started by developing object transport protocols under Professor Jay Black (University of Waterloo).

Gita Koblents, who has been a key JIT developer, worked on ABC++ under Eshrat (York University). She also co-authored an IBM Systems Journal article with me [17].

Michael Xiao, who worked for several years on AppScan at the lab, started out studying requirement compliance for business processes in BPEL under Professor Jenny Zou (Queen's U). He co-authored a conference paper with me [13].

Kien Huynh, who is now helping to drive the future with our Analytics Platform, started with a project on BPEL4WS under Professor Franck van Breugel (York University). She is also a co-author with me [10].

Shu Tan, who for more than a decade was a key developer in IBM's SOA products, started out researching message queue debugging under Eshrat (York University). She co-authored an astounding five patents with me [1], [3], [5], [6], [7].

Yuan Gan, who worked nearly a decade at IBM on Business Process Execution Language (BPEL), started out researching verification of temporal properties for debugging BPEL (winner of Best Student Paper Award at CASCON!). Her research was under Professor Marsha Chechik (U of T), and they are both co-authors with me [11], [12].

John Liu, who has been with the Lab a decade advancing our middleware, started out researching XML debugging under Professor Mariano Consens (U. of T.) He has co-authored a patent and a paper with me [4], [14].

Maria Koshkina, who has her head in the cloud (development, that is) and has been at IBM for more than a decade, started out researching verification of business processes under Professor Franck van Breugel (York University). She has co-authored a patent with me [2].

Without a doubt these student/HQPs have made a huge impact on the Lab, and IBM intellectual property. CAS obviously played a key role in this. Please forgive me if I've forgotten someone – twenty five years is a long time for my poor brain.

Not all the students have stayed with IBM, but they have all made important contributions in the universities and companies to which they went. For example, Frank Eigler (along with Karen Bennet, former CAS ops manager) have advanced open source Linux which is now a vital part of IBM's future. Another example was a student under Eshrat named Neda. She was in the Lab on a CAS project for only a short time (and unfortunately I cannot confirm her surname). However she did a vital piece of speculative development – using an Object Oriented Java-based database to prototype a Java version of the component broker CDM meta-data store. This turned out to be key, because when we demonstrated it, IBM management immediately wanted a full-scale version, which turned out to be one of the first major Java-based projects in the Lab. This is surely a prime example of low-risk high reward investment.

Other students I've lost track of over the years – where are you Henry Lee (ABC++) and Cassandra Liu (reflection in ABC++)? Others have become professors and researchers in their own rights. For example, Jocelyn Simmonds (who as a student worked on web service conversations) is now a professor at University of Chile. Other former students (not on my projects) who have gone on to academia include Joanna McGrenere (UBC) and Wendy Powley (Queen's U). They carry with them an understanding of how technology can be inspired by and can drive industrial applications. In this list we certainly cannot forget Kelly Lyons, who started out as a student from Queen's University (best student paper at CASCON '92), and went on to development at IBM, became Program Director at CAS, and is now a professor at U of T (also on the program committee for CASCON 2015). Her impact on CAS and IBM is unmatched.

3. CASCON

This is the twenty-fifth CASCON, and I'm one of the few around who can boast that I was there from the beginning. In the early days, CASCON was at the Ontario Science Centre. That first year was certainly a trial by fire. We had to setup a number of demos connecting to each other and to servers (but not the world-wide web – as I mentioned it did not exist yet!). For the demos we used mainly XTerms borrowed from the Lab. In order to connect them we used Ethernet, but not wireless (that had not been invented yet either). The Ethernet was provided by coaxial cable connected to BNC connectors on the devices. Getting it working was, well, let's just say, hmm, <expletive deleted>. However, somehow we did. One of the more interesting challenges was switching the Xterms from Token-Ring (used at the lab) to Ethernet. The switches to do this were so small, only one diminutive student named Susan had fingers slender enough to do the job.

The demos that year were exciting, with the biggest attention-getter being a visual analysis of the design principles of the Taj Mahal by Ron Lane-Smith, (U.B.C.). Interestingly, during those years of doing CASCON at the Science Centre I found that the server in the Science Centre control room ran on a brand new open-source operating system: has anyone heard of something called Linux? Subsequently, as CASCON grew, naturally it became more complex to coordinate. One year we were loaned devices to help us coordinate: brick-sized cell phones. Also, it was during our set-up for CASCON on October 23, 1993 that the Blue Jays won the World Series. We had the game on a TV there in the Science Centre and we all shared big cheers for every run the Blue Jays scored. Hopefully the Jays can do it again this year! We could not have guessed that it would take the Jays twenty-two years to again have a real chance, and that in the interim, CASCON would grow to be arguably the largest technical conference in Canada.

Among the growing complexities of the CASCON technology showcase was keeping track of all the computers. Increasingly demos required their own machine, sometimes multiple, and sometimes a large server. We used movers, who in turn needed bigger and bigger trucks. One year once CASCON was over and we had everything back at the lab and had the machines back in place, we noticed that there was one missing – and not a little one. We were missing a server about the size of a kitchen range. How could that be? Was it stolen? Well we went back to the Science Centre and there on the floor, sitting amidst the decorative greenery with hundreds of the regular Science Centre guests milling about, was our server, all crated up and looking very out of place. Amazingly no one had noticed it there. Another year we wound up with an extra server, not IBM's. Obviously one of the university attendees had brought it, but it was completely unmarked. We assumed they would eventually ask us if we had found their server. No one ever did.

Moving things to and from CASCON was hectic, perhaps a little chaotic. One year there was a server that, when we got it back and powered it on, it immediately caught fire! Fortunately there was a student nearby with the good sense to quickly pull the plug. One time, the ever-resourceful Cindy Butler thought it a good idea to store a pallet of CASCON materials (proceedings, bags, pens, etc.) in the IBM automated warehouse behind 844 Don Mills Road. She arranged it and they gave her a numeric code needed to retrieve the pallet. However, I think that may have been the year she left, and I often wonder if that pallet is still in that warehouse, now irretrievable as no one knows the code.

4. ACM COMPETITION

CAS has also been instrumental in bringing the IBM brand and reputation for innovation to literally hundreds of thousands of students around the world. Since 1997, under the auspices of CAS, IBM has been the sponsor of the ACM international collegiate programming contest. In those eighteen years, participation has grown 1500 percent! Brenda Chow and Debbie Kilbride have guided it into a global phenomenon. In the year 2000, Stephen Perelgut and I had the good fortune of attending the world finals in Florida. Instead of flying we drove down to Orlando, and just for fun, brought a combined two spouses and three children with us in a minivan. That was a fun trip – and I mean that! The kids entertained us the whole way.

5. IMPROBABILITY

I have mentioned how challenging the CASCON networking was to set up. Getting the demos themselves to work was a non-linear function in the complex plane. It always seemed infinitely improbable to get them all working at the same time. Anyone who has read Douglas Adams' *Hitchhikers Guide to the Galaxy* will know that future spaceships will be powered by improbability drives. "As soon as the ship's drive reaches infinite improbability, it passes through every conceivable point in every conceivable universe simultaneously." At CASCON, as one by one the demos started working, each one adding to the total improbability, I fully expected that at any moment space-time itself would start to rip, pulling us all into a vortex of causality.

Well maybe I'm exaggerating a little, but CAS itself continues to move on the power of improbability. How improbable it seems that a large company like IBM, that had been around since before computers were invented, could have dreamt-up such a vibrant and successful institution as CAS – with the goal of knocking down the ivory tower and getting the world of academia to work on things that would be useful, practical, and eventually profitable. How improbable that it could have lasted more than twenty-five years. How improbable it seems that we could have been so lucky as to hire so many of those talented (and highly qualified!) students. It turns out Douglas Adams was right after all. He just didn't know he was talking about CAS. CAS ***is*** the improbability drive, taking us forward at the speed of light.

6. REFERENCES

[1] O'Farrell, William, Shu Tan. "United States Patent 8645938 - System and method for replacing code", Issued February 14, 2014.

[2] O'Farrell, William, Dorian Birsan, Mihnea Galeteanu, Maria Koshkina. "United States Patent: 8549481 - Web-based integrated test and debugging system", Issued October 1, 2013.

[3] O'Farrell, William, Shu Tan, Evan Mamas, Simon Moser. "United States Patent: 8473912 - Computer readable medium, system and method of debugging software applications modelled using flow paradigms", Issued June 25, 2013.

[4] O'Farrell, William, Mariano Consens, John Liu. "United States Patent: 8321845 - Extensible markup language (XML) path (XPATH) debugging framework", Issued November 27, 2012.

[5] O'Farrell, William, Grace Lo, Shu Tan, Jane Fung. "United States Patent: 8196109 - Common debug adaptor in a multiple computer programming language environment", Issued June 5, 2012.

[6] O'Farrell, William, Shu Tan, Alex Seeleman, Vivian Mak. "United States Patent: 7287246 - Debugging optimized flows with breakpoints using stored breakpoint trigger indicators", Issued October 23, 2007.

[7] O'Farrell, William, Shu Tan. "Canada Patent: 2345416 - High performance debugging in a message flow environment", Issued May 3, 2005.

[8] O'Farrell, William, Eshrat Arjomandi. "United States Patent: 5999987 - Concurrent processing in object oriented parallel and near parallel systems", Issued December 7, 1999.

[9] O'Farrell, William, Eshrat Arjomandi. "United States Patent: 5655101 - Accessing remote data objects in a distributed memory environment using parallel address locations at each local memory to reference a same data object", Issued August 5, 1997.

[10] O'Farrell, William, Kien Huynh. "Fast and Accurate Path Highlighting for Graph Models of Business Processes", ip.com, June 2010.

[11] Simmonds, Jocelyn, Yuan Gan, Marsha Chechik, Shiva Nejati, Bill O'Farrell, Elena Litani, and Julie Waterhouse. "Runtime monitoring of web service conversations." IEEE Transactions on Services Computing, vol 2, no. 3 (2009): 223-244.

[12] Simmonds, Jocelyn, Marsha Chechik, Shiva Nejati, Elena Litani, and Bill O'Farrell. "Property patterns for runtime monitoring of web service conversations." In Runtime Verification, pp. 137-157. Springer Berlin Heidelberg, 2008.

[13] Xiao, Hua, Brian Chan, Ying Zou, Jay W. Benayon, Bill O’Farrell, Elena Litani, and Jen Hawkins. "A framework for verifying SLA compliance in composed services." ICWS'08. IEEE International Conference on Web Services, pp. 457-464. IEEE, 2008.

[14] Consens, Mariano P., John WS Liu, and Bill O'Farrell. "XPlainer: An XPath Debugging Framework." ICDE'06. Proceedings of the 22nd International Conference on Data Engineering, pp. 170-170. IEEE, 2006.

[15] Tan, Shuxia, Eshrat Arjomandi, Richard Paige, Evan Mamas, Simon Moser, and Bill O'Farrell. "Extending the message flow debugger for MQSI." In Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research, p. 15. IBM Press, 2001.

[16] Arjomandi, Eshrat, William G. O'Farrell, and Gregory V. Wilson. "Smart Messages: An Object-Oriented Communication Mechanism." COOTS. 1996. USENIX Conference on Object-Oriented Technologies. June 17-21, 1996.

[17] Arjomandi, Eshrat, William O'Farrell, Ivan Kalas, Gita Koblents, F. Ch Eigler, and Guang R. Gao. "ABC++: Concurrency by inheritance in C++." IBM Systems Journal 34, no. 1 (1995): 120-137.

[18] Kalas, Ivan, Eshrat Arjomandi, Guang R. Gao, and Bill O'Farrell. "FTL: a multithreaded environment for parallel computation." In Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research, p. 33. IBM Press, 1994.

[19] Arjomandi, Eshrat, William O'Farrell, and Ivan Kalas. "Concurrency Support for C++: An Overview." In C++ Report. January 1994.

[20] Arjomandi, Eshrat, Ivan Kalas, and William O'Farrell. "Concurrency abstractions in a C++ class library." In Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: distributed computing-Volume 2, pp. 919-932. IBM Press, 1993.

[21] Arjomandi, Eshrat, and William O'Farrell. "Concurrency issues in C++." In Proceedings of the 1992 conference of the Centre for Advanced Studies on Collaborative research-Volume 1, pp. 347-358. IBM Press, 1992.

Ever Onward - The Early Days of CAS and CASCON and How they Evolved

Patrick J. Finnigan
Instruments & Information
(M.Math, P.Eng., Senior Member
IEEE, Retired from IBM) 8 Bellemeade
Lane Toronto, Canada M2H 1Y9 +1
416-492-8587
Patrick_Finnigan@ieee.org

ABSTRACT

This paper presents a rather personal and idiosyncratic memoir about IBM Toronto Lab's Centre for Advanced Studies (CAS) by someone involved from its beginning and who had three different roles: as a student, research team leader, and operations manager. These are differing viewpoints of CAS from the perspectives of the key stakeholder groups at that time – i.e. students, researchers and IBM staff, all working to try and make it successful. Then, with the benefit of hindsight, a summary of what has been achieved over the years, and perhaps what could have been done differently – after all we did learn a lot as things evolved.

Keywords

CAS, CASCON, IBM Toronto Lab, Collaborative Research, Technology transfer.

1. BACKGROUND

The IBM Centre for Advanced Studies (CAS) was initiated in 1990. Until that time, technology innovation and technology transfer at the IBM Toronto Lab had been restricted to collaboration with IBM Research and other development labs within IBM Software Group and the Mid-Range Systems Lab in Rochester, MN. At that time, there were also a number of third-party contractual efforts such as with Ada® compilers from the TeleSoft company in San Diego, and CASE tools in collaboration with Hewlett–Packard, as well as initial contact with the Rational Company in California which later became IBM's Rational division. Because IBM was a small Lab at that time, I was able to be involved in all of these efforts.

In order that the proposed collaborative research efforts in CAS were relevant to the Lab departments, a number of staff (including me) were seconded to work at CAS. That is not to say there were "no strings attached". In fact, we had to report to both departmental and CAS management in a matrix fashion. Furthermore, problems which we had helped to create in previous projects remained ours to correct.

I was seconded from the "Compiler Tools Group", where we had been working on CASE environments that worked portably across the new IBM OS/2 "Warp", Microsoft Windows®, and the new IBM RS/6000 AIX® (Unix) workstations[1]. Initially John Maillard kept the small group in line and focused.

It is interesting to note that in 1990 [3], the latest in computing was the IBM Power/1 processor. IBM and Microsoft stopped working together on operating systems – and IBM went it alone with OS/2. Most importantly Tim Berners-Lee successfully set up the first web server at info.cern.ch on December 25, 1990. Ivan Kallas at CAS was in the forefront of investigating this "open", web-based technology. We couldn't believe how much information was "instantly" available in such a short period via the internet. The internet protocols ran over the ubiquitous "VNET" IBM Intranet. Our research collaborators were also very quick to adopt this technology. IBM, especially in its development labs, was rightly very conservative and concerned with security and confidentiality. These concerns did not mesh well with CAS's desire to work in a very open and collaborative fashion.

IBM CAS moved twice before the "big move" to Warden Ave. First we moved from the old IBM Lab building at 1150 Eglinton Ave. in Don Mills, Ontario to 895 Don Mills Rd ("the Tower of Babel", twin towers with distinctive stepped architecture now occupied by Morneau-Sheppell). The next move was to 844 Don Mills Road – the old IBM manufacturing plant, re-purposed to offices. Working late one night, Jacob Slonim, Phil Ford and I were startled by a construction worker in a protective "moon suit" riding on a "mini-mo" tractor knocking down one of our office walls as 844 was renovated! This all settled down after the move to the wonderful new facility on Warden Ave. in Markham in 2001.

[1] "The RS/6000 family replaced the IBM RT computer platform in February 1990 and was the first computer line to see the use of IBM's POWER and PowerPC based microprocessors. RS/6000 was renamed eServer pSeries in October 2000" [8]. I had the great pleasure of meeting John Cocke – the IBM Fellow who invented these architectures and Marty Hopkins who managed the group writing the compilers which were integral to the success of the architecture – the perfect marriage of hardware and software. Just to keep things in relative technological perspective, the first iPod® didn't appear until 10 years later in October, 2001.

2. THE ADVENTURE

EVER ONWARD -- EVER ONWARD!
…We can't fail for all can see
That to serve humanity has been our aim!
….
We've fought our way through -- and new
Fields we're sure to conquer too
For the EVER ONWARD I.B.M.

- From the famous IBM Song [1]

The sentiments of the song echo the general level of enthusiasm and commitment of the staff and visitors to CAS. It is not uncommon in an R&D setting for the teams, technologies and outcomes (i.e. products resulting from the research) to be ill-defined. However, as the song says, nobody involved wanted to "fail", and indeed we all saw the CAS setting as a chance to "fight our way through" in "new fields". We all seemed to share the vision which Grady Booch states so wonderfully in a recent article [9] and I think those involved in CAS today still share this vision (maybe even hoping for a miracle to occur now and then).

> "Software is the invisible writing that whispers the stories of possibility to our hardware, yielding computing systems of exquisite complexity and promise. And we, insiders to computing have the privilege of making the possible manifest"
>
> - Grady Booch, "Then a Miracle Occurs" [9]

As to complexity – it has grown exponentially over the years and shares many forms. See my recent article musing on the full extent of what complexity in IT Systems has become [10].

3. CASCON VERSION 1

Because CAS was focused on *applied* research, we all felt very strongly that publishing papers and presenting at conferences was just not enough to excite our colleagues – even though this gave us credibility as a research organization The Lab departments were looking for technology to enhance the existing products for which they were responsible, or, at higher levels within the lab, potentially new products that could be commercialized.[2]

For this reason, most of our research efforts had a "demo" component built-in. In the early 1990s there was a definite move toward iterative development and prototyping, as well, so having demos early on in a project also helped to change thinking and direction before things had proceeded too far down various blind alleys. We were also having trouble getting any "traction" (maybe even "respect") from those development engineers who, nose-to-the-grindstone were keeping our numerous existing products afloat. Also, the "catchers" of our new technologies were very busy with the aggressive demands to get compilers working for the soon-to-be released RS/6000 workstations and servers, as well as compilers and tools for the forthcoming release of OS/2 (V1.3, December, 1990) [3]. Of course, our CAS teams made use of these evolving platforms – some Beta releases of the OS and tools were better than others!

CASCON, with its papers and demos was seen as the vehicle to get our research and researchers better known, both within IBM and externally. Also, as a uniquely Canadian cross-country gathering of mainly software researchers, it aligned well with the goals of groups like Industry Canada and National Research Council (NRC). After some brainstorming, we picked for the CASCON opening day speeches and demos the closest large enough venue to the "old" IBM Lab at 1150 Eglinton Avenue, which happened to be the Ontario Science Centre. This is a very impressive venue in the "great hall" (the dangling sculptures in the picture in Figure 1 was not there back then), but no real facilities were available for paper presentations and tutorials. Such facilities are limited there, since they run high-school classes regularly.

Figure 1. Ontario Science Centre Great Hall

The decision was taken to mount the kick-off and executive speeches in the great hall together with the demos (to give our research teams "maximum exposure"). The hope was to run this the first day, then retreat back to 1150 Eglinton, where we had the Amphitheatre (seats 300) and other classrooms, for paper presentations and tutorials.

3.1 Team

Frank Ch. Eigler (student) managed all of the software configuring, installing researcher's code on our machines, and the networking. He had wonderful help from other grad students, especially John Buchan from UBC. Phil Ford managed to "borrow" many additional RS/6000 and OS/2 workstations since we needed some to port and verify researcher code as well as have machines setup at the Science Centre.

3.2 Equipment

We moved existing equipment (begged, borrowed, and ?) in order to have as much IBM equipment on display as possible. Hidden

[2] I had some sense of what was involved in commercialization from my role as a "product planner" for IBM's Telesoft Ada compilers for VM and MVS on our S/390 mainframes – it was not a pretty process within IBM then, nor, indeed can it likely ever be. As an aside, we helped Dr. David Taylor of Waterloo disclose some of his technology in a way that it could be licensed by IBM.

away here and there were some researcher machines from other competitive manufacturers. The network was a stand-alone fixed IP address tangle of CAT cables hidden as much as possible (wireless was just a dream back then, although IEEE Std 802-1990 "IEEE Standards for Local and Metropolitan Networks: Overview and Architecture" was indeed published in 1990.).

3.3 Notable and Memorable Incidents

At 2:00 AM, after watching the penultimate available demo come live in the Great Hall, I offered to give John Buchan a lift back to his hotel (Inn on the Park in those days). Believe it or not, my 1986 Chevrolet (Maroon colour) would not start. I remember driving up to the entrance of the Science Centre and John could see immediately from the look on my face there was a problem. Luckily he was an accomplished mechanic and showed me how to remove the air filter, prop open the butterfly valve, and bingo it started right away. I have since used this trick many times on lawn mowers etc. Thank you, John! This is a vignette from CAS in general – we all helped each other out in innumerable ways.

But, little did we know our problems were just beginning. Since attendance had been so low for the opening, it was hastily decided to move all of the demos back to the "gathering area" in front of the Amphitheatre in the Lab at 1150 Eglinton for the second day. This involved moving everything back to 1150 overnight by getting rental trucks and finding the owners and security with access to the model shops at the Lab in order to get power and network to the display machines. This was a consequence of no power or network wired into the "gathering area". Many cables were strung along the corridors back to "machine rooms" close by and carefully taped to avoid any trips (disastrous for the victim as well as the network).

Meanwhile, "reminders" went out to the local managers that, indeed there were things worth seeing, and researchers worth talking to right in the 1150 building on the second day. Attendance increased markedly.

The showcase demo at the first CASCON was the "Taj Mahal" 3D exploded view demo[3]. Several of us spent an anxious week plugging in 3D graphics cards into our most powerful RS/6000 workstation to make this work. I managed to borrow these "on pain of serious injury" from a casual acquaintance at IBM Research with a promise to return them immediately after the conference. Taking them out and restoring the machine to "normal" was actually a trickier job than the original install.

4. CASCON EVOLVES… 25x

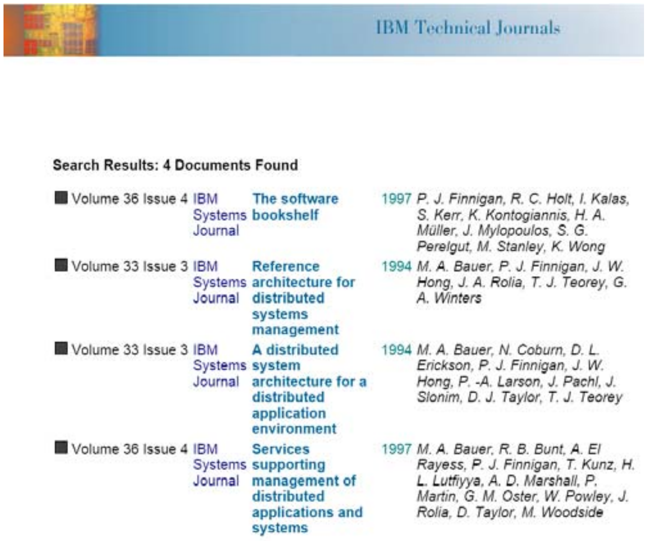
Because we have an archive [5,6] of CASCON proceedings, we can see how the research and research teams evolved over time. It is also worth noting the number of downloads and citations for each of these, which cumulatively must be very large. Having personally helped organize several CASCONS and having attended many subsequent versions of CASCON, it can be safely said that the exhibit hall grows larger and more complicated each year – even though efforts to modularize and standardize have improved by orders of magnitude (thank the IEEE for those wireless 802.xx standards).

The remarkable thing is that it has been run consistently for these last 25 years, especially in the early 1990s when some of us wondered if, indeed IBM would survive our competitive pressures and organizational difficulties (Thank you Mr. Louis V. Gerstner, former CEO of IBM, for saving us from ourselves!).

5. THE PAPERS

CASCON has a tradition of quality papers, which were published in Proceedings starting in 1991. I am happy to have submitted one for the first issue with Kelly Lyons: "Narratives of space and time: visualization for distributed application" (pp. 363 – 391). The content seems now in retrospect a bit obscure, but putting the paper together with Kelly led me to pursue an M.Math. degree at the University of Waterloo, where my thesis explored some of these issues of visualization for large distributed system with Dr. Bill Cowan[4]. On another level, it was very good to see collaboration between IBM and an up-and-coming researcher. Kelly and I put this together in record time, with several trips between Kingston and Toronto As I recall it was stifling hot in Pat Martin's lab at Queen's. We also cleverly used existing public domain examples in the paper which could be easily demonstrated – no programming required for a respectable CASCON demo.

Very early on, working with Morven Gentleman and Anatol Kark from National Research Council (NRC), we established that the review and acceptance process for the papers would be managed by the expert staff at NRC. Without them and their early automation for this, the papers part of the conference would have collapsed.



IBM Technical Journals

Search Results: 4 Documents Found

■ Volume 36 Issue 4	IBM Systems Journal	The software bookshelf	1997	P. J. Finnigan, R. C. Holt, I. Kalas, S. Kerr, K. Kontogiannis, H. A. Müller, J. Mylopoulos, S. G. Perelgut, M. Stanley, K. Wong
■ Volume 33 Issue 3	IBM Systems Journal	Reference architecture for distributed systems management	1994	M. A. Bauer, P. J. Finnigan, J. W. Hong, J. A. Rolia, T. J. Teorey, G. A. Winters
■ Volume 33 Issue 3	IBM Systems Journal	A distributed system architecture for a distributed application environment	1994	M. A. Bauer, N. Coburn, D. L. Erickson, P. J. Finnigan, J. W. Hong, P. -A. Larson, J. Pachl, J. Slonim, D. J. Taylor, T. J. Teorey
■ Volume 36 Issue 4	IBM Systems Journal	Services supporting management of distributed applications and systems	1997	M. A. Bauer, R. B. Bunt, A. El Rayess, P. J. Finnigan, T. Kunz, H. L. Lutfiyya, A. D. Marshall, P. Martin, G. M. Oster, W. Powley, J. Rolia, D. Taylor, M. Woodside

Figure 2. Four collaborative research papers

I also was privileged in the early 1990s to work with four different research teams who reported results in the two special issues of IBM Systems Journal devoted to CAS (Vol. 33 No. 3, 1994 and Vol. 36 No. 4, 1997). Many of the final edits for these articles were done in U. Michigan in Ann Arbor and at Western in

[3] The owner of this demo just escapes my memory, and CAS records only go back so far. If you are him, or know who he is, please drop me an email.

[4] In a nutshell: users can utilize their inherent "parallel visual recognition" to very quickly distinguish items laid out in grids (and some other formats). This is hundreds of times faster than "serial visual recognition" that we use when, say, reading. This can be measured scientifically in "reaction time experiments" used by psychologists to measure user cognition.

London in the winter time. Several trips in icy snow storms should have been avoided, but face-to-face meetings led to much better quality articles.

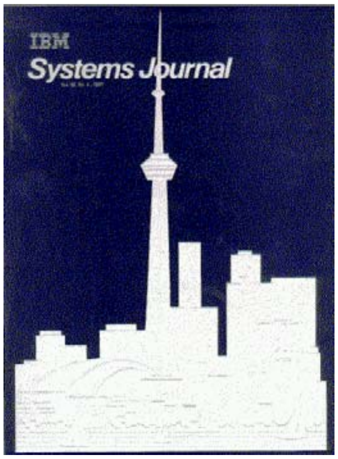


Figure 3. IBM Systems Journal, CAS special issue, 1997

It should be noted that the IBM Systems Journal effectively ceased publications in 2008, although articles of an "applied" nature still appear in the Journal of IBM Research. Fortunately the archive of wonderful articles is still available and being referenced – there are some classics here [7]. Of course, the CAS research teams have published various significant results of their works in other leading journals and conference proceedings – another indicator of successful research sponsored at CAS.

6. WHAT HAS BEEN ACHIEVED

I will explain what CAS and CASCON meant to me personally, then try and speculate what other good things occurred.

In terms of a career, I left CAS and IBM Lab in 1995, to pursue what I felt at the time would be very exciting – IBM Global Services Call/Contact Centre. Certainly, the marriage of telephone and computer technology was very exciting at that time – IVRs, Voice recognition etc. The idea of working in a diverse team, using iterative design techniques etc. which worked so well at CAS, also worked wonders with IBM and joint IBM-customer teams. All of my experience with "Software Architecture" working in the distributed systems projects at CAS led me to pursue certification as an "IT Architect" with the services organizations. Little did I suspect that this evolved naturally into doing initial designs for bids and proposals (ah, the endless hours and deadlines for completing proposals). Working collaboratively on research papers is an invaluable skill when drafting RFP responses. As I suspected when moving to IBM Global Services, the projects there were on scales that could not be duplicated in research projects. For example, at the Province of Manitoba, we had 250 IBMers doing design work for their social services system in an overall team approaching 1,000 people. I was able to put some CAS researchers in contact with these projects which had huge networks, datasets etc. that could be studied.

In the bigger picture, I suspect that time spent as a CAS researcher is personally rewarding for the many who were involved. For example, how else would a world-leading expert on computation theory spend any time speculating on whether you got wetter in the rain by rushing to lunch at Swiss Chalet with the team, or just proceeding at a normal pace? Having these researchers and students on site at IBM Lab made lab staff aware of the pool of outside expertise and sheer enthusiasm available "outside". In collaborative research efforts, it forced many lab teams to document and/or publish details of our existing processes and technologies (e.g. The Software Bookshelf project with the Compiler team). These joint teams also encouraged IBM staff (such as me) to pursue higher education and work collaboratively with "external" organizations. The skills we developed at CAS and CASC|ON certainly were applied in future research projects, but also in product development and other leadership roles. As the song says: "Ever Onward: We can't fail, ... we've fought our way through!". Finally, I know the experience of working with CAS and attending CASCON exposed researchers to the global collaboration possibilities across IBM, as well as issues inherent in very large scale, multi-site software development and maintenance.

7. ACKNOWLEDGEMENTS

Thanks are due to all of the research collaborators (visiting scientists, faculty fellows), graduate students, post-docs and staff at IBM Canada Lab (some now sadly deceased) who made CAS and CASCON what it has become over these 25 years.

8. REFERENCES

[1] IBM Rally Song, "Ever Onward", IBM Archives, http://www03.ibm.com/ibm/history/multimedia/everonward_trans.html

[2] Computer Hope History of Computing http://www.computerhope.com/history/1990.htm

[3] OS/2 Museum Timeline, http://www.os2museum.com/wp/os2-history/os2-timeline/

[4] Great Hall at the Ontario Science Centre (now the Proctor & Gamble Great Hall) http://www.ontariosciencecentre.ca/Private/GreatHall/

[5] CASCON Proceedings Archive. . https://www-927.ibm.com/ibm/cas/cascon/archives.jsp

[6] ACM Digital Library. http://dl.acm.org/

[7] IBM Systems Journal Archive, researchweb.watson.ibm.com/journal/sjindex.html

[8] Wikipedia article on "RS/6000" , https://en.wikipedia.org/wiki/RS/6000

[9] Booch, G. 2015. And Then a Miracle Occurs. IEEE Computing Edge (IEEE Computer Society). (Aug. 2015), 32-34.

[10] Botchkarev, A., and Finnigan, P, Complexity in the Context of Information Systems Project Management, Organisational Project Management 2015, 2(1): 15-34- http://dx.doi.org/10.5130/opm.v2i1.4272 https://epress.lib.uts.edu.au/journals/index.php/opm/issue/view/349

The Case for CAS and its Significant Impact

Paul Sorenson
Professor Emeritus
University of Alberta

paul.sorenson@ualberta.ca

ABSTRACT

IBM's contribution to academic Computer Science research in Canada over the past 25 years has been unmatched by any other company. Much of this contribution has come through the Center for Advanced Studies (CAS) collaborations between members of the IBM Toronto Development Lab and academic partners. In this paper I will provide my personal reflections on how the CAS phenomenon occurred and why it has been so successful and important to Canada.

Keywords

Impact; Canadian Computer Science Research; CAS (Center of Advanced Studies); IBM research; WestMOST; IBM CAS Alberta; ConGESE

1. BACKGROUND

My personal history with IBM started much before the creation of CAS. It began in the late 1960s when I was a Masters student in Computing Science at the University of Alberta. I had the opportunity to work online on information retrieval systems using APL through connections to Yorktown Heights. I completed my MSc thesis on pattern recognition algorithms on an IBM 360 model 67, the first virtual memory system IBM produced. During my PhD studies at the University of Toronto I used high performance IBM 7094 systems and gained significant expertise using JCL (Job Control Language).

In my first academic position at the University of Saskatchewan I was once again using predominantly IBM equipment (various machines from the 360 and 370 families). In 1979-80 I was fortunate to take my first sabbatical at IBM's San Jose Research Center where I joined a project on developing database design tools for a newly developed relational database called System R[1]. This was a tremendously important experience for me, not only because it provide an excellent opportunity to pursue a new area of research, but it also introduced me to the how the industrial research environment operates and sets priorities. This experience helped me later in understanding better how to work with industry in general and CAS in particular.

2. MY EARLY EXPERIENCE WITH CAS

In 1989 I made the decision to move from the University of Saskatchewan to my alma mater, the University of Alberta, taking the position of Chairman of the Department of Computing Science. In this position I was invited to be part of an Executive Advisory Board[2] for the Toronto IBM Development Lab. It was through the discussions with the then Director of the Toronto Lab, Larry Achtemichuk, and Board Members that the idea of forming a research arm of the Development Lab was born. This arm, called CAS (Centre for Advanced Studies), was unique within IBM and unique as an industry-sponsored research lab in computer science in Canada. A number of challenging goals were set for CAS and an excellent article that describes these goals and how the centre was formed to realize these goals can be found in the IBM Systems Journal, Vol. 3 1994 [1].

The overall purpose of CAS was to improve the Toronto Lab's capability to produce high quality software products by enhancing communication between research and development, reducing time from idea to product, and exploiting the window of opportunity of Lab professionals to incorporate new research ideas in the creation of a product. Canadian universities viewed CAS as an opportunity to gain access to industry professionals in order to vet their ideas and test out the practicality of their theories, to have their graduate students placed in a dynamic, industrial-oriented research environment and to more easily collaborate with other academic researchers on large scale, important problems that are of interest in industry. The primary method of engagement for CAS with Canadian universities was through the financial support (CAS scholarships) of excellent graduate students (typically PhD students) of researchers who were doing research in areas of high interest to the Toronto Development Lab. Jacob Slonim, the first Director of CAS and John Schwarz who had succeeded Larry Achtemichuk as Director of the Toronto Development Lab, worked out the funding model for these scholarships with the universities (not an easy task) along with certain provisions such as the scholarship holder spending some months at the Toronto Lab during their PhD studies. The model has evolved over the years and questions related to intellectual property have arisen but overall it has been a great success and stood the test of time.

With the formation of CAS it was quickly recognized that a forum for exchanging ideas between lab staff and academic researchers was a critical success factor. The idea of holding an annual

[1] System R was developed at IBM San Jose Research Center in the late 1960s. It was the Center's first major successful software prototype that later formed the basis for DB2, IBM's first relational database management system.

[2] In 1989, the EAB (Executive Advisory Board) consisted of the Chairs of 11 CSc Departments in Canada with PhD programs along with representatives of the T.J. Watson and Almaden IBM Research Labs.

meeting to facilitate this exchange was born and this forum called CASCON (short I assume for CAS Connections) was started in 1991. I was fortunate to attend the first and all most all of the other CASCON's up until 2012 when I retired. It was one of the most important and productive conferences that I regularly attended. For me, the major benefits of attending were:

- *a meeting place* - almost since its inception CASCON was recognized as the conference in which Canadian academic researchers could meet and share ideas with other academics and with industry practitioners. Over the years, various Networks of Centres of Excellence have met in conjunction with CASCON. Perhaps the largest and longest running such group has been CSER (Consortium for Software Engineering Research). IBM has demonstrated its long-term commitment to CSER by being a major sponsor of CSER projects since its inception in 1997.
- *a demo mart* - CASCON has sponsored, since its inception, a major project demonstration area that has allowed researchers to show their research prototypes to industry practitioners and other academics. For many, this is the primary reason to attend CASCON. I have participated in this part of CASCON on several occasions demonstrating our prototypes for building metacase tools and hook development tools for object-oriented framework.
- *an idea incubator* - I found that CASCON inspired many new ideas for me and my graduate students because of the many excellent keynote presentations from the top researchers from IBM's research labs. These presentations not only described future directions of IBM and industry more generally, but laid out the challenges that needed to be met in pursuit of these future directions. This was a real benefit that I did not receive in going to many international conferences in software engineering.

I would now like to discuss two important "spin-off" activities from CAS and CASCON that I was directly involved in, namely the formation of WestMOST and Alberta CAS.

3. FORMATION OF WESTMOST

In the early 1990s when CAS was created it was recognized during discussions at the IBM EAB that there was a shortage of programs in software engineering in Canada. Many of the existing IBM staff had backgrounds in computer science but limited background in software engineering. There was also a real shortage of faculty who specialized and taught software engineering courses. To overcome this problem ConGESE (Consortium for Graduate Education in Software Engineering) was formed to provide education specially structured for software professionals in Ontario industries. The ConGESE program led to a Masters of Applied Science degree, with a specialization in Software Engineering. This program was offered jointly by several Ontario universities (Carleton, Queen's, Western, York and Waterloo) with strong IBM and other industrial support. Each of the universities offered courses in this program. Because of the extensive uptake by IBM employees, many of the courses were offered at the IBM Development Lab. As time went on, each university belonging to ConGESE developed their own software engineering capabilities and programs, the need for ConGESE died out and it is my understanding that ConGESE no longer offers courses.

A consortium similar to ConGESE was launched in 1993 in western Canada. WestMOST (Western Canada Masters of Software Technology) was a consortium of eight universities (Victoria, UBC, Simon Fraser, Technical University of BC, Calgary, Alberta, Saskatchewan, Regina and Manitoba) that agreed to share graduate courses in software engineering designed for working software professionals across western Canada. The courses would lead to a non-thesis Masters Degree in Software Technology that each institution would offer. Courses could be offered at various locations in western Canada to students registered in a Masters program at an institution of their choosing. To get programs started, the initial courses were offered through the Open Studies Program at the University of Alberta. Both the University of Alberta and the Technical University of BC developed official programs in 1997.

After some challenges in advertising the program and getting strong commitment from most of the WestMOST institutions, a syllabus of twelve courses was agreed to and I became Chair of the WestMOST Steering Committee. Commitment from six industry partners (including IBM) and WED (Western Economic Diversification) aided in developing good initial interest in WestMOST. In the period 1994 to 2001, over 200 students took WestMOST courses. However, like ConGESE the participating universities began to develop their own programs with the hiring of additional faculty in software engineering. The WestMOST Board decided in December 2001 to dissolve WestMOST primarily because the Technical University of BC was closing, leaving only the University of Alberta to offer a MOST degree. University of Alberta committed resources to continue offering courses for two more years to allow those students in the program to complete. Approximately 20 students graduated with MOST degrees from the University of Alberta before the program was terminated.

4. FORMATION OF ALBERTA CAS

In the years 2003-2004 IBM was actively working with the Government of Alberta to find ways to increase its presence and investment in the province to help diversify the Alberta economy. The notion of creating an IBM software development centre specializing in service delivery was discussed but never came to fruition. Instead, IBM worked together with the University of Alberta to examine how the company could leverage some of the Government of Alberta's applied research funding to form a capability in software services that could focus on the existing and emerging strengths in the Alberta economy.

I was fortunate to work with Bernie Kollman, IBM VP, Public Sector Alberta to propose the creation of the IBM Alberta Centre for Advanced Studies (CAS). The operation of the proposed CAS was to be similar to the operation model used in IBM Toronto CAS but with some differences because there was no major development lab in Alberta that could be used to leverage the academic research in Alberta. Instead emphasis was to work with several existing IBM Research labs and, as the opportunities arose, with IBM Toronto CAS. IBM CAS Alberta was formed in 2005 in collaboration with the Government of Alberta, Advanced Education and Technology and the University of Alberta to engage in projects of mutual interest to the province and IBM. Renewed in 2009, the mission of IBM Alberta CAS expanded beyond its initial three year scope to include the University of Calgary with an updated project portfolio which was again strongly aligned with Alberta's research agenda and IBM's

Smarter Planet strategies. Based on its success as an industry-university partnership and its evolving applied research agenda, CAS Alberta was renewed for a third term in 2014.

The following provides an overview of IBM CAS Alberta as taken from its website: https://casalberta.sitecore.ualberta.ca/

"The pillar themes of IBM Alberta CAS's Innovation Framework includes modeling and simulation based research in Health and Bio-Systems, Energy and Environmental Systems and Service Systems with underlying infrastructure aspects of high performance computing, sensor networks, visualization, analytics, big data and cloud computing. Since its inception, IBM Alberta CAS has delivered over $12 million of cash and in-kind funding contributions (from all sources) to Alberta researchers and continues to make a very positive impact on research and economic development in the province.

"The mission of IBM Alberta CAS is to enable strategic, multidisciplinary collaborations of mutual interest and benefit between Alberta's research community and IBM's worldwide research and development staff. As a member of the IBM Worldwide CAS community, IBM Alberta CAS provides a means for global outreach and leadership for Alberta's research community.

"Alberta researchers are currently engaged in initiatives with IBMers from CAS Canada Research and Academic Partnerships in Toronto and Ottawa, CAS Research Triangle Park in North Carolina, IBM's T.J. Watson Research Centers in New York and Massachusetts, and IBM's Almaden Research in California. Discussions are active with IBM's new Brazil Research Center and with other IBM laboratories."

5. CONCLUSION

For me, IBM CAS and CASCON was the most important and innovative research initiative during my 38 year academic career in Canada. While I benefitted early in my career from opportunities provided to me through interactions with IBM research centres, the formation of CAS in Canada was a major positive factor in my research career and those of my students and colleagues. CAS interactions also directly influenced the creation of two major initiatives that I help to start, WestMOST and IBM Alberta CAS.

6. ACKNOWLEDGMENTS

I would like to personally thank many people in IBM for supporting me in my career. It is dangerous to provide a thank you list in fear of leaving out key people I should have thanked. But I will. Special thanks go to Vincent Lum and Mario Schlonick who supported me in my first sabbatical at IBM San Jose Research Center 1979-80. To Larry Achtemichuk, John Schwarz and Gabby Silberman for their leadership and many great discussions about the directions of CAS in its early years. To Jacob Slonim, first CAS Director, who I believe really got CAS off the ground and made things happen. To Maria Klawe, former IBM researcher at Almaden and former Chair of Computer Science at UBC, for her spirit, leadership and belief that CAS could happen. To Kelly Lyons who, as IBM Toronto CAS Director, supported and provided significant input into the creation of IBM Alberta CAS. To Paul Maglio and Jim Spohrer for your insights into Service Science and for hosting me at IBM Almaden Research Lab in October and November 2008 during my last sabbatical. To Cheryl Kieliszewski, Kirk Jordan and Doug Kimelman of IBM Research for their commitment on the IBM Alberta CAS Board and their great support and insights into how Alberta IBM CAS can successfully operate. To Monica Sawchyn, Executive Director of IBM Alberta CAS, for keeping us going and well organized and to Nelson Amaral, for taking over my Co-Chairmanship of IBM CAS Alberta and doing such a great job. Finally and most of all, to Bernie Kollman (my Co- Chair of the Alberta IBM CAS) who I admire greatly and enjoyed working with so much on IBM CAS activities.

7. REFERENCES

[1] Slonim, J., Bauer, M. A., Larson, P. -A., Schwarz, J., Butler, C., Buss, E. B., & Sabbah, D. The Centre for Advanced Studies: A model for applied research and development, IBM Systems Journal, 33(3), 382-398, 1994.

The Professor and the Corporation

David Taylor
University of Waterloo
Waterloo, Ontario
N2L 3G1
1-519-888-4567 x34432
dtaylor@uwaterloo.ca

ABSTRACT

Technology transfer from university to industry is difficult, whether it occurs through creation of a startup company or by working with an existing company. This paper describes the latter possibility, attempting to make some general observations, but largely describing the curious journey of the POET monitoring/debugging prototype, originating in research at the University of Waterloo and ending with the Object Level Trace feature of IBM Component Broker. The Centre for Advanced Studies at the IBM Toronto Laboratory initiated and guided this activity. Its unusual and highly productive model for interaction between industry and university made the technology transfer possible.

Keywords

Technology transfer, distributed computing systems, software debugging, software monitoring.

1. INTRODUCTION

Those of us who work in the “systems” side of computer-science research generally believe that some of the work we do creates or, at least could lead to, practical software artifacts that should be useful to others. If a researcher wants to do more than simply put a prototype onto a web site, for the world to download if interested, there are essentially two possibilities: be an entrepreneur and start your own company to create a commercial version or work with an existing company to get your bright idea into their product line.

It appears the likelihood of succeeding with the last possibility is inversely proportional to the size of the company, so working with IBM should be a near impossibility. One of the great achievements of CAS was to make it possible for me and numerous other researchers to get their ideas and software prototypes into IBM and have them used internally or become part of the product line.

I have structured this account largely around the process of taking some of our work at Waterloo and having it become a feature in an IBM product. Section 2 provides some background on our work at Waterloo and my earlier interaction with industry, then Section 3 describes my sabbatical, which initiated the overall process of technology transfer. Section 4 is a digression on multi-university research projects, as facilitated by CAS (and the way our Waterloo research was involved). Section 5 describes the somewhat arduous process that, beginning with the sabbatical, eventually resulted in a product version of our research software and Section 6 offers some concluding remarks.

2. BACKGROUND

2.1 Research Activity

As of 1991, research at Waterloo on the monitoring and control of distributed applications had been occurring for some time, with the work viewed primarily as facilitating the debugging of distributed applications. Henry Cheung’s Ph.D. thesis (supervised by J. Black) [3] had established important principles for the use of abstraction in distributed debugging but his thesis work did not include any implementation. Subsequent to the completion of Henry’s thesis, our research group used the ideas it contained to build a prototype debugger.

Fred Brooks’s famous dictum that you should plan to throw one away [2] has been argued about (and qualifications have been added by its originator), but it was most certainly true in our case. We built the prototype for use with the Waterloo-developed Shoshin testbed system [10], but with the intention that it could be easily modified for use with another “target” (application-execution) environment. By the time the prototype was operational, it was clear that adapting it to a new target environment would be nearly impossible and that we had made numerous bad design decisions, so that it was barely useable with the Shoshin testbed. As a bonus, we had built the GUI using InterViews [8] and during the time we were building the prototype it became clear that the world had settled on OSF/Motif [4] as the way to build a GUI in an X-windows environment.

It was thus clear that we needed to build a new prototype using our experience in building the first one, but likely not using a single line of source code from it. Since the funding we had been using to pay a research associate ended around this time, it was not at all clear how we could move on to another prototype.

2.2 Industry Interaction

My interaction with industry goes back to graduate school, since my graduate supervisor was very much involved in consulting and industrial research contracts. As a faculty member, I had also done some consulting with industry, but did not find the experience particularly satisfactory. Some of the consulting work was supposed to include aspects directly related to ongoing research at Waterloo, but the reality was that the companies primarily wanted a well informed individual who could provide advice. For me, the interactions were also of fairly short duration, so that I never developed much insight into the goals and operation of any company (and, most likely, they did not develop much trust in the short-term, hired-gun consultant).

Although a faculty member occasionally develops a long-term consulting or contract relationship with a company, such relationships are quite rare. The Centre for Advanced Studies (CAS) was founded to allow IBM to operate on a very different principle–long-term relationships between CAS and researchers, based on mutual interest in specific issues, but not based on a set of deliverables. As I discovered during my sabbatical at CAS, the difference in approach made a huge practical difference.

3. SABBATICAL AT CAS

For all of my other sabbaticals, I have decided what location might be appropriate and then approached someone about the possibility of a sabbatical there. For my 1991-92 sabbatical, however, Jacob Slonim contacted me and asked me if I would be interested in spending my sabbatical at CAS. He had become aware of our work in distributed debugging (and that I was due for a sabbatical). His proposal was that I use my time at CAS to build a second version of our debugger prototype and, facilitated by CAS staff, try to "sell" it to a development group so that the prototype could move into the product line.

I was concerned that he was over-selling the flexibility of a sabbatical at CAS and I would find myself much more management-directed than he was asserting, but eventually decided that it was worth the risk. To my pleasant surprise, the sabbatical proceeded exactly as advertised (other than my involvement in CASCON 1992, as described below).

During the sabbatical, I was able to establish a good architecture for the debugger prototype, based on hard experience with the first prototype, and write a substantial quantity of code. This not only provided a working prototype to demonstrate our ideas, it also provided a sufficiently solid core that graduate students and co-op students could subsequently add features to the code in a reasonably orderly way, avoiding the chaos that can easily occur in a multi-student software project. At some point, in a defensive maneuver, we gave the prototype a name, POET, for "Partial-Order Event Tracer" [7], since the lack of a name was causing colleagues at Waterloo and at IBM to refer to it in odd ways.

The environment at CAS was also good for feedback on the prototype as it developed. I now had a substantial group of people at IBM who were interested in the work and willing to comment on it, in addition to my colleagues back at Waterloo.

During the sabbatical, while there was much useful interaction both inside and outside CAS, it didn't seem to be leading toward a product, but I was not concerned. Aside from being aware that these things take time, even if nothing commercial resulted, we still had a solid prototype for use in our research and the CAS intellectual-property arrangements meant there were no obstacles to that use of the software. As well, I was having a generally enjoyable sabbatical because of the overall atmosphere at CAS.

I presume because Jacob Slonim was trying to bootstrap interaction with university researchers, there were very many sabbaticants at CAS in 1991-92. Because a broad diversity of research areas was represented, direct research interaction was not particularly common, but we could discuss our research with each other at a high level and offer occasional suggestions or, at least, encouragement. In general, it was simply a great opportunity to interact with a substantial number of other academics who were in the less-stressful-than-usual mode of a sabbatical.

In any environment, some service tasks are to be expected and, for my sabbatical, the main one was CASCON 1992. The technical program was organized as papers from faculty, students, and IBM staff, with a co-chair for each. Jacob Slonim was overall PC chair but had so many other responsibilities that he could give only limited attention to the PC. As co-chair for the faculty papers I found myself frequently acting as de-facto PC chair, an exciting prospect in the context of a conference that was taking place for only the second time and was still trying to work out many practical and operational issues. The potential for problems and conflict was quite worrying. In practice, I found that the other two co-chairs, John Botsford and Arthur Ryman, were extremely collegial. Working with them was a very pleasant experience and I believe that, collectively, we succeeded in appropriately managing the creation of the CASCON technical program and established useful precedents for future CASCONs.

4. MULTI-SITE RESEARCH

In addition to facilitating the transfer of ideas and prototype software into IBM, CAS also facilitated multi-university research projects. One side of this is simply that putting together a multi-university team takes work, first to identify a set of researchers who could work together on a particular project and then to convince them that participation is worthwhile. A proactive CAS Director, having identified a particular research project as being of potential interest to IBM can reasonably expend time and effort trying to put together such a team, although the individual researchers might worry that such initial organizational efforts, which can easily fail, are not a wise use of research time.

An obviously practical matter, in the Canadian context, is that much of the research funding is dependent on industrial participation or, at least, expressions of interest. A project that starts with identification of interest by the CAS Director, with universities then "signing on" to participate is very well positioned to obtain grant funding of the scale required to make such a large, multi-university research project succeed.

The Waterloo work on distributed debugging became part of two such large projects centred at CAS: CORDS (COnsortium for Research on Distributed Systems) and MANDAS (MANagement of Distributed Applications and Systems). These were two successive projects, involving slightly different sets of universities. In retrospect, CORDS can be viewed as an initial exploration of issues in distributed systems, using the research expertise of the researchers at multiple universities, leading to a more focused project specifically dealing with monitoring, management, and control of distributed systems. While we originally viewed our work at Waterloo as being an aspect of distributed debugging, we had come to see it as being more broadly about monitoring and, potentially, management and control. Our work thus fit very well with the goals of MANDAS and allowed us to explore how it might fit with work looking at other aspects, such as performance monitoring (our work deals almost exclusively with functional monitoring).

An obvious expectation for CAS-centred and -funded projects was that they present demos at CASCON. For CORDS and MANDAS, there was also a desire that the project have a demo rather than multiple, separate demos by the participating universities. As anyone who has presented a demo will know, it is challenging to structure a presentation so that those dropping by will be able to grasp the problem being addressed and the solution that has been developed. Anyone who has presented a demo will also know that demos cause software to break.

CORDS and MANDAS demos had the added problem that they were, in one respect, single, very large demos and, in another respect, a set of individual demos by the participating universities. There were also significant logistical problems, simply ensuring the required software was installed on the individual machines and that it could communicate with the relevant other software in the demo.

In practice, the great bulk of the equipment used for the demos came from our lab at Waterloo. (This was, unfortunately, still at a stage of technology development that did not allow a demo to be done with a set of laptops, so multiple machines with large CRT monitors needed to be transported.) As a result, Waterloo ended up coordinating the overall demos. For some demos, a mockup was constructed in our lab, with the various universities setting up there in advance so that, in principle, the machines just needed to be shut down, moved to CASCON, and powered up there.

The reality, of course, was more complex. It is in the nature of research projects that software will still be under development until the last moment, so changes were being made even as the demo was being set up at CASCON. Some of these last-minute changes were necessitated by difficulties discovered during the "dry run" at Waterloo. As well, problems would appear only at CASCON because of issues that should be trivial, but aren't, such as changing IP addresses.

We also had the special problem that our demos were heavily dependent on networking and the demo network, which had to be built by IBM staff under extraordinary time constraints, might not be available until well into the demo-setup period. We thus had to bring our own networking to enable initial setup of our demo and then transition to the main demo network. One year, a network hub we had brought remained part of the overall demo network and turned out to have a fault that brought down the entire network. I am still astonished by the calm and professional way that this problem was reported to me (and then quickly resolved).

Although the demos were very high-stress experiences, they provided an opportunity to present our work, in combination with the work from other universities, to a large audience. They were also an opportunity to bond with researchers from other universities as we jointly struggled to deal with the difficulties inherent in presenting a large-scale demo.

5. PURSUIT OF PRODUCT

The intention was, of course that our work at Waterloo would lead to a prototype during my sabbatical, which would then lead to incorporation of our ideas and, possibly, our software into the IBM product line. The process was a longer one than, I think, anyone expected and at many points it was unclear that it would eventually succeed.

Rather than attempt a chronological account of the many events, which would not be very interesting and difficult to reconstruct accurately after many years, the material below is structured around the general issues that arose and that may well be issues in other transfers of technology from a university to a company.

5.1 Importance of Persistence

As already alluded to, the process took a long time. During that time, I tried to maintain perspective and tell myself that if product never happened, that wouldn't be a disaster for me. I also soon came to the conclusion that, while maintaining that sense of perspective, a lot of persistence was required. Discouragement, at times would have been easy. During most of this time, I was visiting CAS once a week as a Visiting Scientist. One week, I made contact with a developer who was interested in our work and we agreed to talk a week later on my next visit. A week later, he no longer worked for IBM.

CAS staff, over time, located many development groups inside IBM who might be interested in our technology. Eventually, I came to the expectation that any given contact was unlikely to be successful, but if I pursued all the possibilities eventually one could be successful. As well, each contact was an opportunity for feedback, potentially leading to changes in POET or new research issues to be pursued.

5.2 Importance of Relationships

The relationship between me (or, more broadly, our research group) and CAS staff was obviously important. The Director of CAS and other CAS staff, most notably Patrick Finnigan, were vital in navigating the corporation. They located numerous development groups that might have an interest in our work, determined appropriate contacts in those groups, and convinced them it was worth taking time out of their busy schedules to talk to an academic whose research and prototype software might be of interest to them.

Developing relationships with willing developers was also key. POET was eventually adopted for use in the Distributed Application Development Toolkit (DADT). A successful conclusion seemed to be imminent. Then, DADT was cancelled. Because DADT only reached the alpha-release stage, it appears that almost all vestiges have disappeared from the web, except for some mentions in papers arising from the MANDAS project [1] and an ancient Developer Support News that is still on-line, mentioning "Distributed Application Development Toolkit for OS/2 Event Trace User's Guide" [6].

The cancellation of DADT could easily have meant going back to the beginning, looking for a new "home" for our work. What actually happened was that I had developed a good working relationship with a key DADT developer, Eric Labadie. He had become convinced of the value of our prototype for IBM and worked very hard to find a suitable new position for it in an IBM product. His efforts eventually led to its inclusion in IBM Component Broker.

5.3 "We Don't Do That"

Any large organization needs to have policies and procedures. It also develops a large body of precedent determining whether and how something can be done. CAS was rather different from anything IBM had done before and didn't always fit nicely with established ways of doing things. This was sometimes an issue with our work, most notably with respect to patents.

POET contained a mixture of well understood principles, research we had already published, and novel ideas conceived during its development. IBM is interested, for obvious reasons, in patenting novel ideas included in products, so there was a desire to consider patents related to POET. Standard processes for invention disclosure were followed and one of the disclosures was viewed as significant enough to warrant pursuing a patent.

Then, someone noticed that the sole inventor was not an IBM employee and immediately said, "We don't do that." If the CAS Director had not been Jacob Slonim, that might have been the end of the patent. He pursued the matter forcefully, pointing out that patents from CAS work should be expected and that some of them would be joint between IBM staff and university researchers, but others would not have an IBM co-inventor.

The "We don't do that" response actually occurred more than once, but Jacob was not deterred and successfully transformed it into "We've never done that before" [9].

5.4 The Distance Between Research Software and Product

I think all academics who produce software prototypes as part of their research understand that there are significant differences between research prototypes and software sold commercially. I suspect that many think the differences are that more testing is required to improve quality and comprehensive documentation needs to be written (and that nuisance small feature you have been ignoring needs to be implemented). All of these are certainly true, but being involved in the process revealed that there is far more.

For me, one of the biggest revelations was internationalization. It is obvious that the entire world does not work in English, but the impact on software is easy to ignore in a research environment, where you and all your colleagues and graduate students can work in English. In coding POET, I had written everything from menus and dialog boxes to internal-error messages in English, with no thought of making the software usable in another language.

I felt a great deal of sympathy for developers who had to go through 100,000 lines of code, locating all the error messages and other English text, assigning message codes and creating a message library, and so on. Clearly no one is going to write a research prototype with all of the mechanism to make it multilingual. That would not make sense; the incremental effort required would probably prevent the prototype from ever being created. On the other hand, giving the matter some thought is advisable if there is potential for turning the prototype into a product. As it happened, I had built an internal table structure to define the menu system, putting all the menu text into one place, but can't claim that I did anything else that eased the burden of the IBM developers with respect to internationalization.

5.5 Success at Last

Finally, in 1998, IBM Component Broker was released, including the feature Object-Level Trace (OLT) built on POET [5] (not the best reference, but one that is still accessible on-line). The process, from starting to build POET through commercial release, took about seven years.

In such a large product there was a possibility that OLT would be a feature no one used. Fortunately, that was not the case, as evidenced by bug reports (not too many and not too serious, fortunately) and requests for additional functionality. While most of the bugs were dealt with entirely by IBM staff, I also had the opportunity to work with them in dealing with some of the more obscure bugs.

6. CONCLUSION

Being an entrepreneur is, I am sure, an exciting challenge for those so inclined. In the context of software development, it allows an individual or a small group to work very hard and bring a bright idea to market. And potentially make a great deal of money when the company goes public or is sold to a larger company. Or, to find that all the hard work has resulted only in yet another failed startup.

Working with an existing company to bring a bright idea to market is an alternative. It is likely to be less intense, but comes with its own set of frustrations. It certainly isn't a recipe for getting rich, but also has far less downside risk. If the objective is simply to get technology into the marketplace, it is equally as valuable as creating a startup, but relies on finding a company that is willing to work with a researcher for this kind of purpose. That is unfortunately rare; the creation of CAS represented a bold step by IBM that greatly facilitated its participation in this kind of activity.

7. ACKNOWLEDGMENTS

Very many people played helpful roles in the long journey from a discarded first prototype to a delivered feature in a product. Jacob Slonim got the whole process started by recruiting me for a sabbatical at CAS and then fought the IBM bureaucracy when the usual way of doing things was an impediment. John Botsford and Nick Cooper provided much help in understanding how IBM works and how I could best work with it. Patrick Finnigan found many contacts in development groups that might be interested in our work. Eric Labadie played a crucial role by believing in the potential of POET and working very hard to move it forward. I am exceedingly grateful to them and to many others too numerous to list here.

8. REFERENCES

[1] M. A. Bauer, *et al*. Services supporting management of distributed applications and systems. *IBM Systems Journal,* vol. 36, no. 4, 1997. pp. 508-526.

[2] Brooks, F. P. *The Mythical Man-Month*. Addison-Wesley. 1995.

[3] Cheung, W. H. *Process and event abstraction for debugging distributed programs*. Ph.D. thesis. Department of Computer Science, University of Waterloo, 1989.

[4] Heller, D., Ferguson, P., and Brennan, D. *Motif Programming Manual*, vol. 6A. O'Reilly, 1993.

[5] Gregor, A., Johnson, D., and Wohlers, C. *IBM Component Broker on System/390* (SG24-5127-00). http://www.redbooks.ibm.com/redbooks/pdfs/sg245127.pdf

[6] IBM. Developer Support News. Issue 17, November 15, 1995. ftp://public.dhe.ibm.com/rs6000/developer/library/dsnews/dsn5q.asc

[7] Kunz, T., Black, J. P., Taylor, D. J., and Basten, T. POET: Target-system-independent visualizations of complex distributed-application executions. *The Computer Journal*, vol. 40, no. 8, 1997. pp. 499-512.

[8] Linton, M. A., Calder, P. R., and Vlissides, J. M. Interviews: A C++ Graphical Interface Toolkit. Technical Report CSL-TR-88-358, Computer Systems Laboratory, ECE Department, Stanford University, July 1988. http://i.stanford.edu/pub/cstr/reports/csl/tr/88/358/CSL-TR-88-358.pdf

[9] Taylor. D. Method and apparatus for constructing displays of partially ordered data. Canadian patent 2,129,190, issued August 17, 1999. (Also U. S. Patent 5,640,500, issued June 17, 1997, with title "Computer program product for enabling a computer to construct displays of partially ordered data.")

[10] Tokuda, H. and Manning, E. G. An interprocess communication model for a distributed software testbed. *SIGCOMM '83: Proceedings of the Symposium on Communications Architectures & Protocols,* pp. 205-212. Austin, TX, 1983.

CAS Students and Highly Qualified People

The Four Pillars of CAS

Marin Litoiu
Department of Electrical Engineering and
Computer Science
School of Information Technology
York University, Canada
mlitoiu@yorku.ca

ABSTRACT

Over the last 25 years, the Centre for Advanced Studies (CAS), at IBM Toronto Lab, has established itself as a world model of collaborative research. I am among the few lucky people who experienced CAS from different perspectives: I was a CAS *student*, an IBM CAS *collaborator*, an IBM CAS *research staff member* and now I am a CAS *Visiting Scientist*. This article is about these four CAS roles that I consider the pillars of CAS. Since I spent most of the time as a Research Staff Member, I will focus mostly on that role, highlighting why it is important and the set of skills that makes that role successful. I also point out some major IBM driven IT events that I witnessed while collaborating with CAS.

Keywords

Collaboration, CAS, IBM, Research, autonomic computing, adaptive computing, cloud computing, control theory, software engineering, performance engineering, Research Staff Member, CAS Visiting Scientist, CAS Fellowship Student.

1. INTRODUCTION

Established in 1990, the IBM Centre for Advanced Studies (or IBM CAS) has become a world-renowned collaboration model among industry, universities and the government research and funding institutions. In the world of innovation, there are many models to follow, but few are comparable with CAS in terms of efficiency, low cost, outreach and multitudes of outcomes. CAS, as any organization, owes its successes to many people working in or with that organization. Students and professors from Canadian, US or European countries, IBM software developers, managers or decision makers alike have contributed to IBM CAS's successes over the years. I am among the few lucky people who have experienced several roles in CAS: I was a CAS *student*, a CAS *collaborator*, a CAS *research staff member* and a CAS *Visiting Scientist*.

In this chapter I describe my perspective on CAS and I focus on the four pillars at the core of its success.

2. CAS'S FOUR PILLARS

I will discuss the roles in the chronological order, as I experienced them.

2.1 CAS Fellowship Students

Many PhD or Master students learn about CAS from their supervisors and I was no different. I first heard of CAS while I was a fresh immigrant to Canada and a PhD student at Carleton University. At that time, I was looking for more industrial experience and I considered leaving the PhD program to take a job either with the Centre de Research in Informatique in Montreal, or with the National Research Council, in Ottawa. It was then that my supervisor, sensing the urgency, said: "IBM Toronto Lab has a research centre, Centre for Advanced Studies, where they carry on research with universities. That is a place for you." I did not know anything about this centre but it was in Toronto and it was IBM, so I could not say no to that. And so I became a CAS *student*. Since then, for 20 years this fall, my professional career has been intrinsically linked to IBM Centre for Advanced Studies.

In 1995, the CAS model was based on student internship embedded within development groups. It meant that students were part of a development group, they attended the group status meeting, and, "occasionally," they helped the development team with feature developments and unit testing. The student research project, meant to validate or extend his/her thesis subject, had to be relevant for that development group. IBM and the software group were in the midst of distributed systems middleware and Internet development. The industry and internet were transitioning from Remote Procedure Calls (RPC) to Objected Oriented Middleware (CORBA) to make the development of distributed systems look like that of the monolithic ones - that is classes, objects and messages. My research project focused on tracing and modeling the performance of distributed systems. There was a very tight integration between my research work and the development features plan. Figure 1 shows me, my development manager (Vito Spatafora) and the development leader (Bin Qin) demoing our joint research/development work at CASCON '97. They could explain and demo the project as expertly as me. While there were many technical aspects I learned from developers, the most important lessons were related to teamwork and communication within and outside the group. On the walls of the Lab, situated at 1150 Eglinton East, there were posters of the Beatles and other famous groups, as a reminder of how important the group work was.

The most memorable event from my student time at IBM was not related to my thesis or work but was rather adjacent. That was the time of IBM Deep Blue versus Kasparov chess matches. Growing up as a chess aficionado and mesmerized by the great champions, Spasky, Fisher, and Karpov, I cheered for Kasparov, hoping he would delay the inevitable, the dominance of brute force machine over man. I followed the games, the openings and the moves. My wish was not granted, Kasparov was defeated. Part of the coronation festivities, a less powerful Deep Blue computer was demoed at CASCON 97, and attendees got the chance to play against it. Nobody had a chance, even with that little machine. Since then, I gained interest in IBM but I lost my interest and love of chess.

Figure 1. CASCON 97 Demo. With the ADTC developers, Vito Spatafora and Bin Qin.

2.2 IBM Employees and CAS Collaborators

During my last year of PhD, I was offered a full time position at IBM (one of the outcomes of a CAS project) with the same group I was doing the internship. And so I started my second life in CAS, as an IBMer. The Centre for Advanced Studies was thought as a university-IBM collaboration place, with few full time IBM researchers but with many IBM developers acting as part time researchers. The role of the development group was first and foremost to initiate and champion CAS research projects and be the ultimate beneficiary of the research outcomes. It was understood that no CAS research project could succeed without the support of a development group. There were several reasons for that: the researchers from universities needed to understand the IBM software processes and products so they could ground their projects in the real world. Students needed technical mentorship and research partners while working on IBM Toronto Lab premises. For Lab software developers, a CAS project offered a chance to get familiar with the latest research trends, expand their creativity, get involved with CASCON by either co-authoring a paper, presenting a demo or giving a talk in a workshop. At that time, our development group, Application Development Tools Centre (ADTC), was busy transitioning IBM software development tools from Visual Age to Eclipse. For the amateurs of software engineering history, IBM's first generation of visual tools was called Visual Age and it was developed in Smalltalk (see Figure 2 for a commemorative plaque). It was very hard to maintain and evolve the tools to keep up with the plethora of technologies and development needs arising just before the 2000 IT bubble. A leaner approach and plugging architecture was needed and that led to the open source Eclipse[6] that is known now to any software developer. While Eclipse was an IBM wide effort, IBM ADTC in Toronto was in charge of the distributed middleware support and later on with business development tools. Research questions around those technologies were transferred to CAS and research projects were initiated and carried out by several research groups over many years. Eclipse become known in Universities and used by students due to the CAS as well as University Relations efforts.

2.3 CAS Research Staff Members

I joined CAS as a Research Staff Member in 2000. The title "CAS Research Staff Member" (or RSM) was borrowed from IBM Research Division where the hierarchy is flat and everyone had the same title. However, the job description was different: a CAS RSM spent a big chunk of his/her time doing project and people management. Roughly, I spent 40% of my time doing research, 40% doing project and people management and 20% in committees, conference organization (including CASCON). This varied from month to month and made the life very interesting and rewarding, as I was not locked in one repetitive activity.

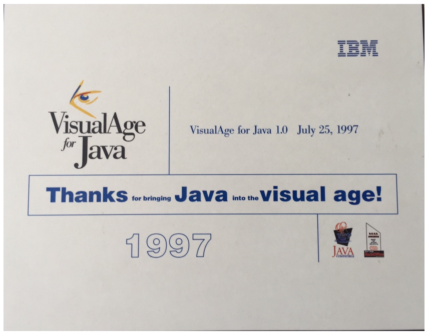

IBM
VisualAge for Java
VisualAge for Java 1.0 July 25, 1997
Thanks for bringing Java into the visual age!
1997
JAVA

Figure 2. Visual Age for Java, first edition, written in Smalltalk. This paved the road to Eclipse.

As an RSM, I basically had freedom to organize my time, choose the university and IBM collaborators and initiate projects. There was a competition among RSMs for budget and projects and, at the time of funding decisions, we inevitably became competitive and our egos got bruised when the projects we supported were rejected. However, those small squabbles were quickly forgotten and there was a real sense of collegiality among all of us.

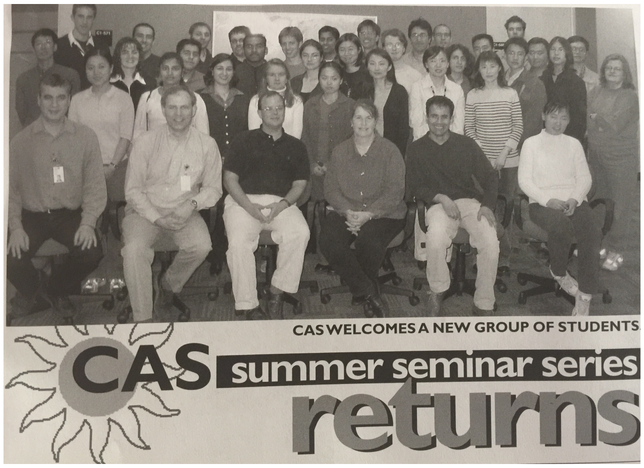

CAS WELCOMES A NEW GROUP OF STUDENTS
CAS summer seminar series
returns

Figure 3. CAS welcomes the CAS fellowship students and the summer seminar series (scanned from IBM Log-on News).

A CAS RSM activity revolves around CAS projects and there were many projects each year. Figure 3 shows some of the students joining CAS in the summer of 2003, an indication of CAS activity size. During my time as CAS RSM, with few exceptions, all students were hosted in one open area. That created a CAS research centric environment (as opposed to development group centric) by allowing more interaction between RSMs and students (Figure 4) and among students (Figure 3 banner mentions the CAS summer seminars, hosted by students).

As an RSM I had to look both inward and outward. The IBM stakeholders (product and project managers, department managers, developers) who invested time and money in CAS wanted to see results: patents, smart features in their products, better skills and knowledge transferred to their groups, as well as new great graduates hired.

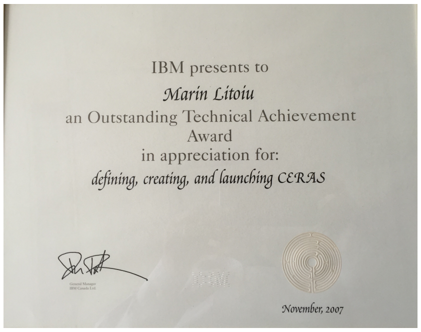
Figure 4. With CAS students in CAS open area: Marin Litoiu, Gerard Tarcisius, Faryaaz Kassam, Jenny Zhou, Maria Koshkina

While CAS had a dedicated budget for projects and CASCON, that was only augmented with direct contributions from the development departments. Consequently, the project selection and funding decisions were done with input from the development groups, as was the continuous project evaluation. There was always a conflict between the academic and the industry timing. While the university researchers measured time in Masters or PhD thesis milestones, the software industry worked with quarters. Progress had to be shown quarterly and be substantial. I learned that this asynchronicity cannot and must not be tried to be solved; at most, it can be mitigated. It should be explained and reminded to both universities and development participants that they should not try to change each other's different worlds. Both parties should have the right expectations and mitigate the risks.

An RSM has to look outwards as well, write papers, present to and organize conferences, sometimes get involved in government policy or research committees. While these activities are not necessary and always recognized in IBM, they are sine-qua-non aspects that give credibility to CAS, IBM, and to RSM(s). I describe two major outward initiatives because CAS should be proud of them both.

2.3.1 *CAS and Autonomic Computing Initiative.*

One of the main outward initiatives I led in CAS was the IBM Autonomic Initiative. Started in 2003, at the corporate level, its goal was to motivate the university professors and students to tackle the complexity of software management by starting projects and courses on Autonomic Software, which is software that manages itself. CAS was an influential actor in this and by partnering with IBM University Relations and IBM Autonomic Computing Division, we started with 2 projects in 2003 and reached 25 projects in 2005[7]. Autonomic computing has become a teaching subject in universities and a research subject since then. An offspring of autonomic computing initiative was the creation of the symposium for Software Engineering for Adaptive and Self-managing Software, SEAMS[3]. We started it as a workshop, the Design and the Evolution of Autonomic Systems(DEAS), co-organized with CAS Visiting Scientists Hausi Muller and John Mylopoulos. The goal was to inject software engineering principles to the Autonomic Computing field. Later, DEAS morphed into SEAMS and is now in its 12th year. Another offspring was the Center for Excellence for Research in Adaptive Systems (CERAS), a research project started by IBM CAS and Ontario Centre of Excellence (Figure 5). This was the first large Canadian project looking into virtualization and cloud research. Other national projects followed, among them the Smart Applications on Virtual Infrastructure (SAVI) that would not have been possible without CAS support.



IBM presents to
Marin Litoiu
an Outstanding Technical Achievement
Award
in appreciation for:
defining, creating, and launching CERAS

General Manager
IBM Canada Ltd.

November, 2007

Figure 5. Marking the Creation of Centre of Excellence for Adaptive Systems (CERAS)

I highlight here the most important CAS research contributions in the autonomic computing field, contributions in which CAS and I played an important role. The contributions had impacted both IBM and academic community:

- Pioneering the use of control theory in tuning performance models. This work was conducted with Murray Woodside (Visiting Scientist), Tao Zheng (CAS student) and with Gabriel Iszlai (IBM developer). The first paper capturing the idea was published at CASCON 2005[4] and a string of other papers followed[8][18][20]. This work has won several awards, culminating with the Most Influential Paper of the Decade Award, at CASCON 2015. The ideas also led to an IBM patent, by the same researchers.
- Formalizing the composition and analysis of autonomic systems using control theory. I have done this work with Dan Ionescu (CAS Visiting Scientist), Bogdan Solomon (CAS student), Gabriel Iszlai and Mircea Mihaescu (IBM Developers) [13][12][15][9].
- Performance modeling of autonomic systems[10] and performance metrics extractions from source code. The latter involved Jim Cordy (CAS Visiting Scientist), Nevon Brake (CAS student) and IBMers Valentina Popescu and Elisabeth Dancy [24].

2.3.2 *CAS and Software Engineering Research in Canada*

CAS can also claim its major influence in another major initiative in Canada that I chaired while I was a CAS RSM. It is the Consortium for Software Engineering Research (CSER), a research consortium comprising companies and universities. During the time I was the chair of CSER, IBM CAS funded most of CSER projects and also provided logistics and in-kind support. CSER involved many other companies and has been a model of precompetitive research. At the same time, both IBM and Canada software industry has benefitted from CSER, which provided a platform for training many generations of software engineers.

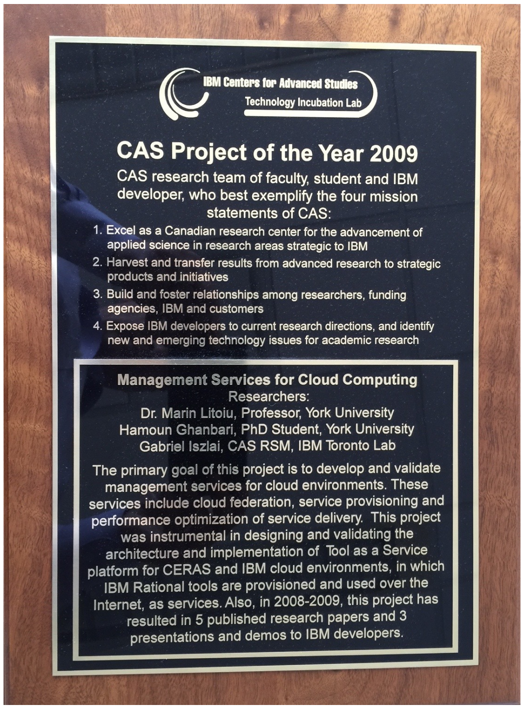
Each year, CSER researchers have been the main contributors to CASCON events. Lately, CSER has evolved in a conference, collocated with CASCON.

Among the most impactful research I carried on with CSER researchers, I would like to mention:

- Business processes visualization and understanding. This work has been the result of a long collaboration with Peggy Storey (CAS Visiting Scientist) and CAS students Ian Bull, Derek Rayside[11][19][22][23].
- Architectures and design patterns for web service evolution, work with Hausi Muller (CAS Visiting Scientist) and Piotr Kaminsky (CAS student)[17][21].

2.4 CAS Visiting Scientists

Since I went back to my first life, of a professor, I had the opportunity to work with CAS as a Visiting Scientist, the forth pillar of CAS model. Seeing CAS from the other side, that of the University researcher gives me another perspective and a 360^0 view of university-industry collaboration. The vast experience CAS has in working with universities is mostly evident in its understanding of university research and in its processes developed around the win-win principle.



IBM Centers for Advanced Studies
Technology Incubation Lab

CAS Project of the Year 2009

CAS research team of faculty, student and IBM developer, who best exemplify the four mission statements of CAS:

1. Excel as a Canadian research center for the advancement of applied science in research areas strategic to IBM
2. Harvest and transfer results from advanced research to strategic products and initiatives
3. Build and foster relationships among researchers, funding agencies, IBM and customers
4. Expose IBM developers to current research directions, and identify new and emerging technology issues for academic research

Management Services for Cloud Computing
Researchers:
Dr. Marin Litoiu, Professor, York University
Hamoun Ghanbari, PhD Student, York University
Gabriel Iszlai, CAS RSM, IBM Toronto Lab

The primary goal of this project is to develop and validate management services for cloud environments. These services include cloud federation, service provisioning and performance optimization of service delivery. This project was instrumental in designing and validating the architecture and implementation of Tool as a Service platform for CERAS and IBM cloud environments, in which IBM Rational tools are provisioned and used over the Internet, as services. Also, in 2008-2009, this project has resulted in 5 published research papers and 3 presentations and demos to IBM developers.

Figure 6. The CAS Project of the Year Award honours the CAS team that best epitomize the CAS mission statement. The recipients are members of the four CAS pillars.

From the project's inception and submission until the final stage of the project, the trajectory is as smooth as it can be. The intellectual property agreements, the first and major hurdle in a university-industry collaboration is already in place and agreed upon with most universities in Canada. This agreement elaborates on the ownership of newly created intellectual property, the freedom of publishing while at the same time protecting IBM confidential information. This contrasts with companies not versed in collaboration where it is often required to spend more than a year defining and signing an agreement. Also, CAS is very efficient in enabling access to IBM people, products and processes. Basically, as a Visiting Scientist you get physical and virtual access to the IBM Canada Lab and to IBM as a whole. While physical access was available from the very beginning of CAS, the virtual access to IBM in its entirety was created gradually to provide continuous project collaboration, easier sharing of software artifacts or access shared repositories. As an example of this transition, in my earlier days as a CAS RSM, if a student or a professor wanted to work on the project off cycle (outside the student four months internship), we had to ship an IBM computer to the university because, according to IBM internal procedures, "all development should take place on IBM owned computers." Besides giving virtual access to IBM, CAS also has a series of events that facilitate the interaction with IBM people. The most notable events are CASCON conference and the CAS University Days. While CASCON is known and available to a larger audience, the IBM CAS university days are meant for closer collaborators, IBM Visiting Scientists and Faculty Fellows and their students. Taking place over several days at IBM Toronto Lab, usually in May, those events allow for deep dives into research topics of interest to both IBM and university researchers, who share confidential information and establish future research priorities.

Figure 7. Receiving the "Faculty of the Year Award" from Bart Vashaw (Program Director, IBM). The award acknowledges the Faculty who had a lasting impact on IBM people, products and processes.

As I work with IBM CAS from the academic side, I notice the same organic collaboration I experienced as IBMer. The research or practical accomplishments of the research projects are shared with CAS researchers and IBM developers. Below are some accomplishments I originated from the academic side but in which IBM CAS has a major role:

- Defining the architecture for hierarchical cloud optimization, work done with CAS researchers and other CAS visiting scientists[2], work for which we received the Best Paper Award.
- Cloud provisioning performance models, work done with Johnny Wong (CAS Visiting Scientist), Ye Hu (CAS student) Gabriel Iszlai (IBM CAS RSM) [16].
- Cloud services for management of Tools as a Service and the first deployment of IBM Rational Tools as a service in Cloud. This accomplishment received the CAS Project of the Year Award (cf. Fig 6).

- Advancement of IBM research agenda on cloud computing for which I received the IBM Faculty of the Year Award (cf. Fig 7).
- Extended Kalman filters and particle estimators for multiclass software and business processes, with my CAS students Hamoun Ghanbari and Andrei Solomon and CAS RSMs Alex Lau and Gabriel Iszlai[14][25].

3. CONNECTING THE DOTS

Students, postdocs and colleagues who know my professional trajectory, often ask me the following questions: "Is it worth being a CAS student?", "Should I take an industry job and later return to university as a professor?" "Should I collaborate with industry?" One fallacy of posterior analysis is that one can connect the dots of the past in a nice story in which all the steps seem well choreographed and put together through a vision. Another fallacy is that, if the steps are being planned and choreographed, a professional trajectory must be repeatable. These fallacies apply to rare events[5] but mutatis-mutandis apply to professional careers narratives as well. Therefore, although I am hesitant to give definitive answers, I always emphasize both the pros and cons on making career choices.

A CAS fellowship and internship is an amazing opportunity for students who love practical things, do not have yet a strong industrial or practical experience and/or need practical case studies to validate their research. Maintaining permanent contact with development groups teaches communication and group dynamics while exposing the students to new technologies; however, it takes time and effort and might even slightly delay the thesis.

Choosing an industry job after getting a PhD when you are interested in University jobs is risk taking. Personally, I made that choice, I interviewed and declined academic positions and I took the IBM CAS RSM job, thinking that, if I wanted, I could apply for a faculty position later on. While it worked in my case and a few others, an industrial job will slow down the publication throughput, if not cut it completely. For some companies, publications are not an incentive; you might get yourself too busy with your release schedules, team dynamics and competition. There is a big chance you will never get back to a university professor job, unless you carefully manage your trajectory. However, if you succeed in your transition, you will have plenty of industrial and academic collaborators to work with and this is not a given in universities.

On collaborating with CAS and industry in general I give a resounding yes, for many reasons. An industrial collaboration keeps you grounded, working on relevant problemns, it gives you an opportunity to validate your assumptions. More importantly, students ought to know that they work on relevant research questions; many of them want contacts in industry to further their careers.

4. CONCLUSIONS

CAS is a proven collaboration model among university researchers and IBM. CAS owes its successes to many students, professors, IBMers and CAS staff members. I highlighted in this chapter the importance of the four pillars and I enumerated some of the CAS accomplishments in which I played a part. Definitely there are many more accomplishments and I hope other CAS members and collaborators will bring them to light. As we celebrate CAS' 25th birthday, I am happy and proud I have contributed to its growth and outreach and I hope the CAS model will live on for many years to come.

ACKNOWLEDGMENTS

My many thanks go to IBMers who helped make CAS a success during my full time with IBM and with CAS: CAS managers Joe Wigglesworth, Gabby Silberman, Kelly Lyons, Joanna Ng gave me the freedom rarely seen in the corporate world; I had tremendous help from other CAS RSMs Julie Waterhouse, Jen Hawkins, Paul Smith, Terry Lau, Calisto Zuzarte, Robert Enenkel, Elena Litani, Stephen Perelgut, Marcellus Mindel (current CAS manager) as well as very strong support from IBM development and management leaders Kevin Stoodley, Bob Blainey, Arthur Ryman, Mike Starkey, Mircea Mihaescu, Tom Doucher, Alan Ganek, Jeff Kephard, Gabriel Iszlai, Andrew Trossman, Harm Sluiman and others. Visiting Scientists and CAS students made my days in CAS rewarded with fun and accomplishments: Johnny Wong, Hausi Muller, Peggy Storey, Murray Woodside, Dan Ionescu, John Mylopolous, Jim Cordy, Tao Zheng, Mike Smit, Bogdan Solomon, Brad Simmons, Ian Bull, Ladan Tahvildari, Jenny Zou, Kit Barton, Alex Duran, etc. In CAS, I also worked with many IBM interns who made my life enjoyable, Tom Plackson, Alexandru Litoiu as well as many staff members, Cheryl Morris, Brenda Chow and Debbie Kilbride. To all, and many others I forgot to mention, a big Thank You!

5. REFERENCES

[1] M. Litoiu, Hamid Khafagy, Bin Qin, Anita Rass Wan, and J. Rolia. 1997. A performance engineering tool and method for distributing applications. In *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research* (CASCON '97), J. Howard Johnson (Ed.). IBM Press

[2] Marin Litoiu, Murray Woodside, Johnny Wong, Joanna Ng, and Gabriel Iszlai. 2010. A business driven cloud optimization architecture. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (SAC '10). **Best Paper Award,** ACM, New York, NY, USA, 380-385.

[3] Mylopoulos, J.; Litoiu, M.; Smit, M., *Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2013 Proceedings on* , vol., no., pp.iii-iv, 20-21 May 2013

[4] Tao Zheng, Jinmei Yang, Murray Woodside, Marin Litoiu, and Gabriel Iszlai. 2005. Tracking time-varying parameters in software systems with extended Kalman filters. In *Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research* (CASCON '05), James R. Cordy, Anatol W. Kark, and Darlene A. Stewart (Eds.). IBM Press 334-34 **(Most Influential Paper Award of the Last Decade, CASCON 2015)**

[5] Nassim Nicholas Taleb, *The Black Swan*, The Random House Publishing Group, 2007

[6] Eclipse open source, http://Eclipse.org, Sept 15, 2015.

[7] Jeff Kephart, Autonomic Computing, the First Decade, Keynote at ICAC 2011, http://www3.cis.fiu.edu/conferences/icac2011/files/Keynote_Kephart.pdf, Sept 2015

[8] T. Zheng, C. M. Woodside and M. Litoiu, "Performance Model Estimation and Tracking Using Optimal Filters," *Software Engineering, IEEE Transactions on,* vol. 34, pp. 391-406, 2008.

[9] B. Solomon, D. Ionescu, M. Litoiu" Web Service Distributed Management Framework for Autonomic Server

Virtualization", in *Comp. and Inf. Scie, vol. 18,* Springer Verlag, 2008, pp 61- 71.

[10] Litoiu M., "A Performance Analysis Method for Autonomic Computing*," ACM Transactions on Adaptive and Autonomous Systems (TAAS)*, Vol 2, No 1, pp. 1-27, 2007.

[11] Rayside D., Litoiu M., Storey M., Best C. Lintern R., "Visualizing Flow Diagrams in WebSphere Studio Using SHriMP Views," *Information Systems Frontiers*, Vol. 5, No 2, pp. 161-174, 2003.

[12] L. Checiu, B. Solomon, D. Ionescu, **M. Litoiu** and G. Iszlai, "Observability and controllability of autonomic computing systems for composed web services," in *Proceedings of the 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI),* 2011, pp. 269-274.

[13] B. Solomon, D. Ionescu, M. Litoiu and G. Iszlai, "Self-organizing autonomic computing systems," in *Proceedings of the 3rd IEEE International Symposium on Logistics and Industrial Informatics (LINDI),* 2011, pp. 99-104.

[14] H. Ghanbari, C. Barna, M. Litoiu, M. Woodside, T. Zheng, J. Wong and G. Iszlai, "Tracking adaptive performance models using dynamic clustering of user classes," in *Proceedings of the 2nd Joint IEEE/ACM WOSP/SIPEW International Conference on Performance Engineering,* 2011, pp. 179-188.

[15] B. Solomon, D. Ionescu, M. Litoiu and G. Iszlai, "Autonomic computing control of composed web services," in *Proceedings of the ACM/IEEE Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS'10),* 2010, pp. 94-103.

[16] Ye H., Wong J., Iszlai G., Litoiu M., "Resource Provisioning for Cloud Computing," *Proceedings of CASCON 2009*, November 2009, pp. 101-111.

[17] Kaminski P., Litoiu M., Muller H., "A Design Technique for Evolving Web Services," *Proceedings of ACM IBM Center for Advanced Studies Conference (CASCON 2006), ACM Digital Library*, pp.211-222, Toronto, 2006.

[18] Zheng T., Woodside M., Litoiu M., "Service System Resource Management Based on a Tracked Layered Performance Model," *Proceedings of IEEE International Conference on Autonomic Computing*, Dublin, Ireland, pp. 175-184, 2006.

[19] Bull I., Storey M. A., Favre J. M., Litoiu M., "An Architecture to Support Model Driven Visualization," *Proceedings of the IEEE International Conference on Program Comprehension,* Athens, June, pp 100-106, 2006.

[20] Woodside M., Zheng T., Litoiu M., "The use of optimal filters to track parameters of performance models," *Proceedings of IEEE International Conference on Quantitative Evaluation of Systems (QEST),* Turin, Italy, September 2005, pp 74-83, 2005.

[21] Jun Ma, Holger M. Kienle, Piotr Kaminski, Anke Weber, Marin Litoiu, "Customizing Lotus Notes to build software engineering tools," *Proceedings of the ACM IBM Conference of the Centre for Advanced Studies (CASCON 2003), ACM Digital Library*, pp.211-222, October 06-09, 2003.

[22] Derek Rayside, Marin Litoiu, Margaret-Anne Storey, and Casey Best. "Integrating SHriMP with the IBM WebSphere Studio Workbench," *Proceedings of the ACM IBM Conference of the Centre for Advanced Studies (CASCON 2001)*, Toronto, Ontario, Canada, ACM Digital Library, pp.79-93, November 5-7, 2001.

[23] Margaret-Anne Storey, Casey Best, Jeff Michaud, Derek Rayside, Marin Litoiu, and Mark Musen, "SHriMP Views: An Interactive Environment for Information Visualization and Navigation," *Proceedings of ACM CHI 2002*, Minneapolis, Minnesota, April 20-25, pp. 520-521, 2002.

[24] Brake N., Cordy J., Dancy E., Litoiu M., Popescu V., "Automating Discovery of Software Tuning Parameters" Proceedings of Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2008); 30th IEEE/ACM Software Engineering Conference on Software Engineering (ICSE 2008), 2008.

[25] Solomon A., Litoiu M., Lau A., "Business Process Adaptation on a Tracked Simulation Model," *ACM IBM Center for Advanced Studies Conference (CASCON 2010)*, November 2010, pp. 184-198.

The Role of IBM CAS on Research Innovation and Student Training

Ying Zou
Department of Electrical and Computer Engineering
Queen's University
Canada
ying.zou@queensu.ca

ABSTRACT

The IBM Center for Advanced Studies (CAS) and CASCON celebrate their 25th anniversary in 2015. CAS has had a great impact on shaping the careers of Canadian faculty members and graduate students as they work to help the Canadian information technology sector grow and innovate. This paper summarizes my experience with CAS and CASCON on the training of highly qualified personnel and on producing world-leading research results through collaborative research.

1. INTRODUCTION

Established in 1990, the IBM Center for Advanced Studies (CAS) is a center that facilitates collaborative research between academic faculty members and IBM personnel. CAS supports such collaborative research through various funding programs, such as the IBM CAS fellowships and the CAS strategic research grants. The CAS research staff link the academic researchers with IBM personnel who have interesting and challenging practical problems which are in need of deep understanding and novel solutions. Such collaborative engagements often result in paper publications in premier international venues, and are commonly transferred into IBM new products or processes.

Hosted by IBM CAS, CASCON is an annual international conference on computer science and software engineering. CASCON offers a wide range of programs, including research papers, workshops, keynotes, and technology showcases. One of the key distinct features of CASCON is that it facilitates vivid discussions and knowledge transfer between academic researchers and industry. Over the past 25 years, CASCON has become Canada's largest industrial academic conference, attracting over 1,500 attendees every year from around the world. It is a terrific meeting place for academics and the software industry in Canada.

Speaking of my own personal experience, I have been working with CAS since 1999 when I was a first year PhD student in the Department of Electrical and Computer Engineering at the University of Waterloo. CASCON was the first conference that I have ever attended. Since then, I have attended CASCON almost every year. CASCON is my favorite conference since it enables me to stay in touch with industry as an academic while also keeping tabs of the work of other academics across Canada. For the past 17 years of my experience, IBM CAS and CASCON have had a tremendous impact on my career and the careers of my students.

As a student, I was fortunate to be awarded an IBM CAS PhD fellowship. The fellowship allowed me to spend four summer terms at IBM CAS in IBM Toronto laboratory. These four summers were my most memorable and productive semesters during my PhD studies. CAS provided modern facilities and a friendly environment for research collaborations. My PhD supervisor, Dr. Kostas Kontogiannis, and I had many brainstorming meetings with IBM personnel to better understand their practical research problems and we received a tremendous amount of constructive feedback to guide my research and to improve its impact on practice.

CAS offered open spaces to facilitate the discussion among the fellowship students working on different projects to exchange ideas. All the CAS fellowship students got together in CAS over the summer. The students were encouraged to give lightening talks on their research on a weekly basis. I got the opportunity to meet many intelligent fellow graduate students around the world to share research ideas and discuss leading edge technologies on various occasions, such as hallways, lunch breaks and buses to the IBM Toronto lab.

CAS is also a wonderful place for the students to have fun besides conducting research. Many group events were organized by CAS research staff, such as an annual picnic, and by the peer students, such as movie nights and party gatherings. All these made my PhD studies enjoyable! Many of the fellowship students keep a life-long friendship that goes the boundaries of one's research.

After I joined Queen's University as a faculty member, I was very grateful to establish a research collaboration with IBM CAS again, and have been working with many IBM personnel who are innovative and passionate about practical research and are always willing to mentor, guide and shape students into future leaders. It was the greatest pleasure to work with Joanna Ng, Tinny Ng, Dianna Lau, Leho Nigul, Alex Lau, Jay W Benayon, Bill O' Farrell, Elena Litani, Jen Hawkins, Bhadri Madapusi, Terry Lau, and Tack Tong. Such collaborative research interactions helped me shape my research directions. With the tremendous support from IBM CAS, I was able to apply for matching funds, such as Natural Sciences and Engineering Research Council (NSERC) Collaborative Research and Development Grants (CRD) projects and Ontario funded projects to support a large number of graduate students and undergraduate students to work on the extended research scope of my CAS fellowship projects. As always, IBM CAS has continued the tradition of welcoming my graduate students to spend their summer semesters at IBM CAS.

In the rest of this paper, I would like to share some experiences on how CAS has impacted the training of graduate students and helped shape their thesis research.

2. A SHORT SURVEY ON TRAINED HQPS

CAS has taken a very active hands-on and constructive role in the training of Highly Qualified Personnel (HQP) (i.e., PhD students, Master's students, undergraduate students and Postdoctoral fellows). To best describe the students' personal experience with IBM CAS and CASCON, I decided to have the students themselves share their thoughts about CAS. I sent a short survey to my eight past and current graduate students who have worked at IBM CAS on their thesis topics. The survey consisted of two questions:

1) What you liked the most about CAS and CASCON?
2) What did CAS do to your career?

Five out of the eight surveyed students responded. Two of the respondents are my current PhD students. The rest of three replies are from two of my former PhD students who were awarded with IBM CAS PhD fellowship, and one Master's student. The former students are currently working in the software industry in Canada.

The key themes emerged from their feedback about IBM CAS and CASCON include:

- Innovative and passionate IBMers
- Practical research problems
- Friendly and constructive support
- Unique access to IBMers and tools
- Knowledge sharing and collaboration
- Preparing for their future careers

In the following subsections, I list the excerpts from their replies.

2.1 What you liked the most about CAS and CASCON?

[Reply 1] *"I value the wonderful people and their enthusiasm towards technology I met during CAS and CASCON. Especially, one can meet a lot of like-minded people in CASCON. Some of them are still in touch. CASCON is one of the most interesting conferences I have been. CASCON provides ample opportunity for knowledge sharing and collaboration. My days in CAS were equally interesting. Looking back, lots of activities (such as CAS picnics, and bowling) we did during CAS and I had a very accomplished experience with CAS."*

[Reply 2] "*I like the most about the support that I got from CAS and IBMers. CAS are very open to students. CAS treat students like IBM employees: we have access to almost all the IBM software products for our research, and most of the IBM internal resources. More importantly, when needed, CAS staff are always ready to help us and try to get other IBMers to join in & help our research projects. I benefited a lot from IBMers' knowledge and insightful feedback. (IBMers are very kind and willing to help CAS students)."*

[Reply 3] "*I love working with IBM CAS researchers. It has been my greatest pleasure to work with IBMers and CAS research staff in the past 3.5 years. They do research with a passion. They offer constructive comments and suggestions to me in every project meeting. They help me refine my thesis topic."*

[Reply 4] "*I like the open atmosphere they provide which promotes the exchange of idea and collaboration between the research students. For example, the open space seating plan CAS arranged for the research students makes it really easy for us to have conversations with other students. Before CAS I've had experience in other companies such as Microsoft Research where each student stay in their own cubical. I like the CAS way much better. In addition, I remember CASCON has a really big Poster section where CAS research students (and other researchers as well) can share their preliminary research ideas. In summary, I think CAS and CASCON has been very successful in providing an open atmosphere and venue for its research students to work and collaborate*."

[Reply 5] "*The research in CAS aims to tackle the real world problems. I am honored to work in CAS to think, explore the practical world. CAS values every idea and does not restrict your mind. I have the freedom to think about every possible solution. And people in CAS gave me valuable feedback.*

CASCON gives me the opportunity to widen my horizon. I can look into other research thoughts in other areas. We have similarity and difference. Every idea give me inspiration."

2.2 What did CAS do to your career?

[**Reply 1**] "*CAS helped me to grow both personally and professionally. In addition to laying ground for new ground breaking research, it helped me grow interpersonal skills through constant discussion and presentation.*"

[**Reply 2**] "*My research experience at CAS was a mix of academia and industry. On one hand, we conducted academic research at CAS. On the other hand, because the research projects were connected with IBM products to better serve IBM's products or resolve challenges from the practice. Our research had a very solid foundation in industry. As a result, I got good background to seek both industry and academia positions.*

For my own career, I got the opportunity to start my career as a software engineer at IBM. I believe my experience at CAS played a very important roles to get the position and to adapt to the new position very fast."

[**Reply 3**] "*CAS helps broaden my research horizons. It helps me understand the problems in industry. Doing research to solve real problems makes my research competitive*."

[**Reply 4**] "*My CAS experience definitely has developed my skill-set and helped improve my ability to tackle a complex research questions. CAS has provided ample opportunities for research students to receive feedback and comments from senior industrial partners about the ongoing research project by having frequent meetings. This experience has provided an invaluable new perspective to me which is difficult to obtain in a pure academic setting. I am very happy that I had the CAS experience and it definitely helped me excel in my current industry job."*

[**Reply 5**] "*IBM CAS research staff provide great mentorship to shape the research direction of my PhD thesis. They help me grow my creativity and knowledge to overcome the obstacles and find the light during the course of my PhD study."*

3. RESEARCH RESULTS

Over the past twelve years, my graduate students and I have collaborated with IBM CAS on three areas: smart SOA and services, mobile and smart interaction, and business process driven software development. One of the major objectives for the collaboration is to work on real life industrial problems and provide practical solutions. Our CAS collaborators, such as Joanna Ng, Tinny Ng, Dianna Lau, Alex Lau and Leho Nigul, are domain experts on the research problems and provide frequent and constructive feedback to enhance our research work. The CAS research staff along with the IBM developers gave us the unique research problems to solve, access to their code base and documentation in order to study the real world software systems and the deployment environment to adopt and test our research results in a real world setting.

Through such collaborations, we produced fruitful results which have been published in eight international journal papers, three book chapters and thirty-three international conference papers. These papers are published in the top tier international journals and conferences with competitive acceptance rates (e.g., 12%-35%), such as the IEEE Transactions on Software Engineering (TSE), the IEEE Transactions on Services Computing (TSC), the International Conference on Web Services (ICWS), the International Conference on Service Oriented Computing (ICSOC), and the International Conference on Software Maintenance and Evolution (ICSME). In recognition of the strategic value of some of our research ideas and results, IBM Canada filed two patents to protect such ideas.

In the following subsections, I briefly introduce my graduate students' theses that are direct results of our close and productive collaborations.

3.1 Smart SOA and Services

Background: Nowadays, end-users increasingly rely on on-line services to conduct various web tasks, such as on-line shopping and trip planning. Potentially, end-users compose different web services to create web tasks in order to fulfill their personal goals. Despite the increasing use of the web tasks to support human activities, most web tasks are performed manually by end-users. In particular, there are a large number of services available in the internet. The previously used web services are not tracked for the end-user. Therefore, an end-user has to manually sift through the sheer volume of web services to identify desired services and build web tasks. Moreover, the end-user needs to repetitively configure the web tasks to suit their own needs when they want to invoke the previously used web tasks again. The end-users suffer from cognitive overloading due to the lack of automated tool support for service composition and web tasks configuration.

Innovation: To shelter end-users from the complexity of service composition, we provide techniques that assist non-IT professional end-users in service composition by dynamically composing and recommending services to meet their requirements. To acquire the desired process knowledge that specifies the needed web tasks, we propose various approaches to automatically extract process knowledge from different sources, such as the existing commercial applications on the Web, the ontology definitions, and on-line how-to instructions.

Instead of requiring end-users to specify detailed steps for service composition, the end-users only need to describe their goals using a few keywords. Our approach expands the meaning of an end-user's goal using process knowledge then derives a group of web tasks to help the end-user fulfill the goal [8, 9]. To ease the discovery of desired services, we index web services based on the semantic concepts available in the service description documents and help users to formulate search queries for web services [2]. Online reviews are analyzed to help users select web services from a set of functionally similar web services [4]. Our approach automatically identifies the data flow between web services and generates a user interface to execute a composite service. To avoid users entering redundant information to execution services, we provide techniques to automatically fill the parameters for the users to invoke services and chain the services that need to execute in sequence [6].

Moreover, a set of services can be frequently used together among different applications to deliver a unique functionality. Such set of services are called a service composition pattern. The reuse of the service composition patterns can facilitate the composition of new applications, improve existing applications and optimize maintenance processes of services. To facilitate the identification and reuse of service composition patterns, we provide an approach that mines the service composition patterns from execution logs produced by service-oriented applications during runtime [3]. Since the execution logs can be produced by heterogeneous web resources, we propose a unified description schema to describe various web resources in order to identify functionally similar services of different types. This helps reveal complete service composition patterns. Then we identify frequently associated services using Apriori algorithm and heuristics.

3.2 Mobile and Smart Interaction

Background: The number of mobile applications has increased drastically in the past few years. However, some applications are more popular than other applications. For mobile applications, the user perceived quality can be quantified by the number of downloads and ratings. Earlier studies suggested that user interface (UI) design is one of the major reasons that can affect the user-perceived quality of mobile applications.

Innovation: To facilitate the developers to build mobile apps with high user perceived quality, we have examined the effect of UI design on user-perceived quality by focusing on two different aspects, namely UI complexity and UI reuse [1]. We carry out our empirical studies on 1,292 Android applications from the Android market (i.e., Google Play). Our studies confirm that UI complexity quantified by the number of inputs and outputs is associated with the user perceived quality. We provide guidelines for the proper amount of UI complexity that helps an application achieve high user-perceived quality. We also observe that UIs of mobile applications are widely reused among and across different domains. Frequently used UI elements with certain characteristics can provide high user-perceived quality. Finally, we extract a catalog of UI templates with high user-perceived quality from existing mobile applications for developers to design UIs with high quality. Developers and quality assurance personnel can use our guidelines to improve the quality of mobile applications.

3.3 Business Process Driven Software Development

Background: In today's ever-changing environment, business processes are constantly being customized to reflect the up-to-date organizational structure and business objectives. It is a challenging task to design and maintain a business application, which needs continuous improvement to adapt to the changes in business. To achieve the growing requirements from the business world, business applications have been gradually evolved and provide sophisticated functional features in user interfaces and business logics. However, such rich features are often not obvious for users to navigate in the user interfaces (UIs). As a result, business users, especially novice business users, may have difficulties in using these business applications to fulfill business duties. Moreover, the existing development approaches rely on software developers' craftsmanship to design and implement business applications. Such a development paradigm is inefficient and leads to inconsistency between business processes and business applications. These problems result in increase in operational cost and decrease in work efficiency.

Innovation: To support evolving business processes in software development, we design and develop a code generation framework that automatically analyzes business process specifications and generates user interfaces, software architecture and code skeletons of business applications. To improve the usability of business applications, we leverage contextual information in business process specifications to generate the sequence of the tasks presented in the user interface and guide business users to accomplish their work more efficiently and effectively [11]. To help software practitioners generate a preview of software architecture, we identify architectural components from business process specifications by analyzing dependencies among tasks [7]. To verify the achievement of quality requirements, we extend a set of existing code product metrics to automatically evaluate the quality of the generated software architecture designs. Eventually, we apply refactoring strategies, such as software architectural styles or design patterns, to optimize the generated software architecture designs and resolve identified quality problems. The effectiveness of our proposed approach is illustrated through case studies.

To maintain business processes and the underlying software applications in sync, we provide techniques that automatically extract business processes from the source code of e-commerce systems [10]. To generate a high level abstract representation of business processes, we analyze the user-system interaction and users' navigation flows to identify tasks and the coordination between the tasks. To provide detailed descriptions of business processes, we capture business policies and data flows in the source code to infer the business logic. To refine the results of the extracted business processes, we incorporate the documents specified by business analysts. We compare the documented business processes with the extracted business processes and calculate the similarity between two types of business processes based on their structures. From the comparison results, we establish the linkages between the source code and the documented business processes. Finally, we develop a prototype tool as an eclipse plugin.

4. CONCLUSION

In this paper, I have summarized the personal experience of my students and myself with IBM CAS and CASCON in the aspects of training highly quality personnel (HQPs) and conducting collaborative research. IBM CAS is terrific institute that provides invaluable support to help both the faculty members and the graduate students successfully advance our careers and benefits Canada to grow and innovate in the information technology sector. CASCON is always my favorite conference which distinguishes itself as a place for meeting of minds between industry and academics. I wish the success story of IBM CAS and CASCON to continue.

5. ACKNOWLEDGMENTS

I would like thank Dr. Hua Xiao, Dr. Bipin Upadhyaya, Shaohua Wang, Yu Zhao and Ran Tang for their participation in the survey. I would also like to acknowledge the contributions of the following students' theses to the research results section: Dr. Hua Xiao, Dr. Xulin Zhao, Dr. Bipin Upahyaya, Shaohua Wang, Rang Tang, Syed Ehsan Salamati Taba, and Mao-Keng Huang.

6. REFERENCES

[1] S. E. S. Taba, I. Keivanloo, Y. Zou, J. W. Ng, T. Ng, "An Exploratory Study on the Relation between User Interface Complexity and the Perceived Quality," 14th International Conference on Web Engineering (ICWE), LNCS 8541, pp. 370-379, Jul 1 2014, Toulouse, France, Springer International Publishing.

[2] B. Upadhyaya, F. Khomh, Y. Zou, A. Lau, J. Ng, "A Concept Analysis Approach for Guiding Users in Service Discovery," IEEE 5th International Conference on Service-Oriented Computing and Applications (SOCA), pp. 1-8, Dec 17 2012, Taipei City, Taiwan.

[3] B. Upadhyaya, R. Tang, Y. Zou, "An approach for mining service composition patterns from execution logs. Journal of Software: Evolution and Process Special issue: selected papers from IEEE 12th International Symposium on Web Systems Evolution (WSE) 25(8), pp. 841-870, Aug 2013.

[4] B. Upadhyaya, Y. Zou, SH Wang, J. Ng, "Automatically Composing Services by Mining Process Knowledge from the Web," 11th International Conference on Service Oriented Computing (ICSOC), LNCS 8274, pp. 267-282, Dec 2 2013, Berlin, Germany, Springer Berlin Heidelberg.

[5] B. Upadhyaya, Y. Zou, I. Keivanloo, and J. Ng, "Quality of Experience: User's Perception about Web Services," IEEE Transactions on Services Computing, 8(3), pp. 410 – 421, May-June 2015.

[6] S. Wang, Y. Zou, I. Keivanloo, B. Upadhyaya, J. Ng, and T. Ng, "Automatic Reuse of User Inputs to Services among End-users in Service Composition," IEEE Transactions on Services Computing, 8(3), pp. 343 – 35, May-June 2015.

[7] X. Zhao, Y. Zou, "A Business Process Driven Approach for Generating Software Architecture," Journal of Software Practice and Experience (SPE), 41(10), pp. 1049-1071, Wiley, Sep 2011.

[8] H. Xiao, B. Upadhyaya, F. Khomh, Y. Zou, J. Ng, A. Lau, "An Automatic Approach for Extracting Process Knowledge from the Web," IEEE International Conference on Web Services (ICWS), New Application and Experience Track, pp. 315-322, Jul 4 2011, Washington, DC

[9] H. Xiao, Y. Zou, R. Tang, J. Ng, L. Nigul, "Ontology-driven Service Composition for End-users," Journal of Service-Oriented Computing and Applications (SOCA), 5(3): 159-181, Springer London, Mar 2011.

[10] Y. Zou, J. Guo, K Foo, M. Hung, "Recovering Business Processes from Business Applications," Journal of Software Maintenance and Evolution: Research and Practice (JSME), 21(5), pp. 315-348, Wiley, Sep/Oct 2009.

[11] Y. Zou, Q. Zhang, X. Zhao, "Improving the Usability of e-Commerce Applications using Business Processes," IEEE Transactions on Software Engineering, 33(12), pp. 837-855, IEEE Computer Society, Dec 2007

The Value of IBM CAS Canada for PhD Students

Norha M. Villegas
Department of Information and
Telecommunications Technologies
Faculty of Engineering
Universidad Icesi, Colombia
nvillega@icesi.edu.co

ABSTRACT

The collaboration and research model established by the IBM Centre for Advanced Studies (CAS) of Canada provides many benefits for PhD students to succeed in their studies and future careers. Pursuing a PhD as a CAS student is a luxury that very few graduate students have. This paper revisits my experience as a PhD CAS student of IBM Canada between 2009 and 2013 to illustrate the value that IBM CAS Canada provides for PhD students.

Keywords

IBM CAS Canada, PhD students, student mentoring, context-awareness, CASCON conference, CSER, CAS University Days, CAS projects, CAS students, University of Victoria.

1. INTRODUCTION

I conducted my PhD at University of Victoria (Canada) under the supervision of Professor Hausi A. Müller. The problem addressed in my dissertation is the dynamic management of context information, with the goal of improving the relevance of situation-aware smart software (SASS) systems' context-aware capabilities with respect to changes in their requirements and execution environment [7]. To this end, my thesis focused on the investigation of dynamic context management and self-adaptivity to: (i) improve context-awareness and exploit context information to enhance quality of user experience in SASS systems, and (ii) improve the dynamic capabilities of self-adaptivity in SASS systems. The experience of being a CAS student between 2010 and 2013 was key for the success of my PhD studies. This paper documents my experience as a PhD CAS student of IBM Canada to illustrate the value that CAS has for PhD students, particularly for finding their research problems, validating early ideas and contributions, improving research, presentation and English communication skills, and consolidating a strong research and professional network.

2. LANDING

I started my PhD in September 2009, after arriving in August from Colombia to spend one month improving my English communication skills at University of Victoria (UVic). I had the fortune of being accepted as a PhD student at this university by Professor Hausi Müller, thanks to my mentor and great friend Grace Lewis from Carnegie Mellon Software Engineering Institute, who introduced me to Hausi. Doing my PhD under the supervision of Hausi was the best thing that could happen to my research career. I am infinitely indebted to Hausi for giving me the opportunity of conducting a PhD program full of exciting adventures and challenges. I could not have made a better decision; Professor Hausi Müller is the best supervisor I could ever have.

One of the first big adventures after starting my PhD was to attend CASCON 2009. It was very exciting to attend so many interesting talks. I remember myself as one of the most engaged attendees, paying attention to all the talks and taking notes carefully. Of course, I was somehow afraid to participate in the middle of an audience full of famous professors and senior graduate students. CASCON 2009 played a major role in successfully starting my journey as a PhD student. This CASCON conference style absolutely helped me understand how the research arena works at international conferences. Furthermore, the engaging research environment of CASCON challenged me to participate as a presenter a year later. I wanted to have a more active role.

My goal at CASCON 2009 was to find my research topic. This was a very ambitious objective for a foreign student pursuing the first quarter of her PhD program. Fortunately, I was not so far from this goal. Fortuitously, I was invited with my supervisor to SITCON: The *CAS/NSERC Strategic Workshop on Smart Internet Technologies*,[1] where I found stimulating ideas for my research.

3. FINDING MY RESEARCH PROBLEM

During my graduate studies I had the fortune of being actively engaged in top international software engineering research communities such as IBM CAS Canada, CSER,[2] SEAMS,[3] ICSME,[4] and MESOCA.[5] This section discusses how IBM CAS Canada helped me identify research challenges for my PhD thesis.

After CASCON 2009, I went back to Victoria full of interesting questions and challenges in my head around context awareness. Context information in the Smart Internet realm was a protagonist in the 2009 conference. Consequently, I started a review of the state-of-the-art context-awareness and its role in realizing the vision of the Smart Internet. Having started the development of a systematic literature review on this topic, my supervisor received an invitation to contribute a chapter for a Springer-Verlag book entitled Smart Internet [1].

This invitation represented an important milestone in identifying the challenges to be addressed in my research. We submitted a

[1] http://research.cs.queensu.ca/~cordy/SITCON/
[2] https://sites.google.com/site/cser2015fall/
[3] https://www.hpi.uni-potsdam.de/giese/public/selfadapt/seams/
[4] http://conferences.computer.org/icsm/
[5] http://mesoca.etil.ca/

chapter entitled “Managing Dynamic Context to Optimize Smart Interactions and Services” [2] that not only became the first publication of my PhD studies, but also played a crucial role in developing my research and English skills, as well as in finding the research problems to be addressed in my thesis. That chapter surveys context modeling and management approaches intended for the optimization of smart interactions and services, discusses the main challenges and requirements for supporting context-awareness as needed by the smart Internet, and provides a feature-based framework useful to guide the evaluation and implementation of context modeling and management mechanisms in light of these requirements. Having a comprehensive and systematic literature review published as a Springer-Verlag book chapter within the first year of a PhD program is a milestone that only students engaged in a research community such as CAS IBM Canada can achieve.

A second milestone in determining the research problems for my thesis was the CAS proposal written in 2009, with which I became a CAS student in 2010. I still remember how excited I was when Hausi asked me to write a CAS proposal. Today, six years later, while reading the proposal again to write this chapter, I realize that the early objectives stated in that first CAS proposal were actually part of my thesis. This is evidence of the added value provided by CAS to grad students. Most PhD students take one year or more to find the research problems to be addressed in their dissertations.

Being a CAS student implies to visit IBM as an intern every year. Internships at IBM were always useful to play with and experience IBM technologies, as well as interact with highly recognized IBMers. During my first internship I investigated the application of dynamic context management techniques to Smarten-Up SOA systems based on WebSphere platforms.[6] That research concentrated on two challenges. The first one was the integration of dynamic context management into the WebSphere Fabric Business Assembler.[7] The second one was the implementation of feedback loops to control dynamic binding and service discovery in Fabric, according to policies and context observations. I fondly remember the productive discussion sessions I had during my 2010 internship at IBM with Joanna Ng (CAS Director), Alex Lau (CAS RSM) and Chris Brealey (STSM/Architect at IBM Canada).

CASCON 2010 was a milestone in validating the early contributions of my research work. As I dreamed and envisioned in 2009, I played an active role in the new version of the conference. I presented a poster, together with Ishita Jain and Sowmya Balasubramanian (UVic Master students), about the findings of my first CAS internship at the CASCON showcase (cf. Figure 1). Moreover, I had the opportunity to co-chair with Hausi Müller, Alex Lau and Stephen Jou a CASCON 2010 workshop entitled “Dynamic Context-Aware Applications: Approaches and Challenges” [3]. In this workshop I presented the findings of the survey published in the Smart Internet book and the first proposal of a context management framework that was the genesis of one of the main contributions of my thesis. This presentation was a truly interactive session, the room was completely full. All attendees participated in a discussion based on research questions I was reviewing by then. The feedback was absolutely positive, in particular from senior researchers. I never forget the words of Mark Chignell, University of Toronto: “I look forward to seeing your progress in realizing all the interesting ideas presented today in CASCON 2011.” Many people congratulated me for the quality of the proposal and my presentation skills. I still remember the adrenaline rush I felt in that workshop. I am really indebted to IBM CAS Canada for having given me valuable opportunities to grow as a researcher.

Figure 1. Ishita Jain, Norha Villegas and Sowmya Balasubramanian at CASCON Showcase

4. SHOWCASING FOR THE FIRST TIME AT CASCON

CASCON 2011 arrived after one year of intensive and productive work. My CAS project for 2011 focused on the application of dynamic context management techniques to the improvement of user experience in the Personal Web. This work was done in collaboration with graduate students from my home university: Universidad Icesi, Cali, Colombia. This time I had a paper accepted for the conference entitled “A dynamic context management infrastructure for supporting user-driven web integration in the personal web” co-authored with Hausi Müller (my supervisor), Juan Muñoz from Universidad Icesi, and Alex Lau, Johana Ng and Chris Brealey from IBM [4]. Furthermore, during several months we prepared a great demo on the application of our dynamic context management framework to improve shopping experiences in Smarter Commerce scenarios. This line of research led to several highly cited papers [5, 6, 7, 10].

The days of CASCON 2011 were very intense and full of adrenaline. Besides presenting the full paper, we showcased our demo during all days of CASCON. Our demo was full of visitors all the time. Hausi made sure to invite every single important person to see our work. One of the most exciting interactions was the visit by Tim Francis, B2B & Commerce Solutions CTO at IBM. We spent more than 30 minutes presenting our demo and discussing our ideas with Tim. For me and our team, this was the

[6] http://www.ibm.com/software/websphere

[7] http://www.ibm.com/developerworks/library/ws-WSBFoverviewpart1/

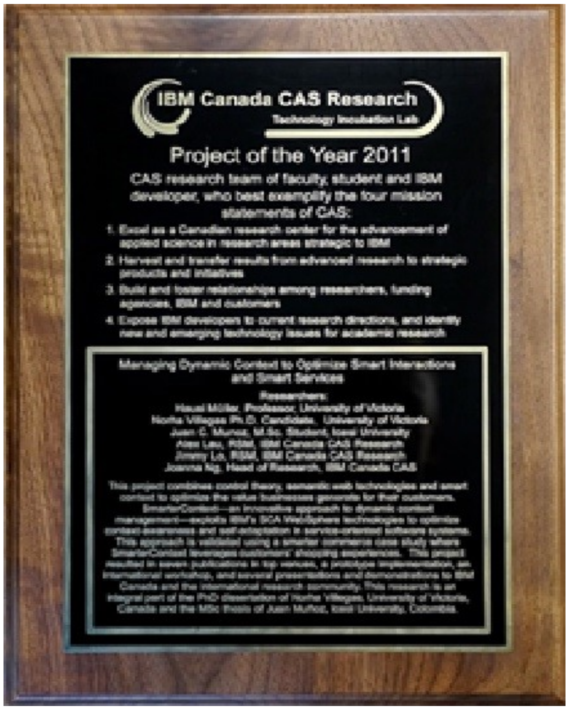
ultimate experience presenting our industrial research results to the head of Commerce Solutions. I remember this event as one of the most challenging and valuable moments of my days as a CAS student.

5. THE FIRST AWARDS

5.1 2011 CAS Project of the Year Award

Our demo and all the work conducted during 2011 were a big success. Our research received the IBM Canada CAS Research Project of the Year 2011 award (cf. Figure 2). I did not expect that, despite all the hard work. This was a very exciting moment. This award was possible thanks to all the hard work done by Juan Muñoz, who was conducting his master thesis at Universidad Icesi in Colombia; and Catalina Sierra and Miguel González, who were undergraduate students at Universidad Icesi under my supervision at that time. Juan Muñoz is now conducting his PhD on requirements engineering for self-adaptive software at University of Paris I, Catalina Sierra is a Product Manager at Quandoo GmbH (Berlin) and Miguel González is a Software Testing Engineer at Microsoft (United States).

This award was a huge achievement in my PhD studies. I was very honored for having our project selected among several high quality projects conducted by many researchers who I very much admired. It is great for PhD students to be part of a research community that recognizes the quality of research projects. This is very important for the future research and academic careers of PhD students. Furthermore, I was very happy to see how proud my supervisor was seeing my work being recognized.



IBM Canada CAS Research
Technology Incubation Lab

Project of the Year 2011

CAS research team of faculty, student and IBM developer, who best exemplify the four mission statements of CAS:

1. Excel as a Canadian research center for the advancement of applied science in research areas strategic to IBM
2. Harvest and transfer results from advanced research to strategic products and initiatives
3. Build and foster relationships among researchers, funding agencies, IBM and customers
4. Expose IBM developers to current research directions, and identify new and emerging technology issues for academic research

Managing Dynamic Context to Optimize Smart Interactions and Smart Services

Researchers:
Hausi Müller, Professor, University of Victoria
Norha Villegas Ph.D. Candidate, University of Victoria
Juan C. Munoz, M.Sc. Student, Icesi University
Alex Lau, RSM, IBM Canada CAS Research
Jimmy Lo, RSM, IBM Canada CAS Research
Joanna Ng, Head of Research, IBM Canada CAS

This project combines control theory, semantic web technologies and smart context to optimize the value businesses generate for their customers. SmarterContext—an innovative approach to dynamic context management—exploits IBM's SCA WebSphere technologies to optimize context-awareness and self-adaptation in service-oriented software systems. This approach is validated using a smarter commerce case study where SmarterContext leverages customers' shopping experiences. This project resulted in seven publications in top venues, a prototype implementation, an international workshop, and several presentations and demonstrations to IBM Canada and the international research community. This research is an integral part of the PhD dissertation of Norha Villegas, University of Victoria, Canada and the MSc thesis of Juan Muñoz, Icesi University, Colombia.

Figure 2. IBM Canada CAS Research Project of the Year 2011

5.2 2012 CASCON Best Paper Award

In 2012, I worked with Sahar Ebrahimi, a master student of UVic supervised by Professors Hausi Müller and Alex Thomo, applying my context management engine to a recommender system [5]. The goal was to improve the effectiveness of recommendations by incorporating context information gathered, managed and provisioned by our SmarterContext engine [6]. This work received the Best Paper Award in CASCON 2012.

Figure 3. Norha Villegas and Hausi Müller receiving the CASCON 2012 Best Paper Award presented by Program Chairs Jenny Zou and Hans-Arno Jacobson

6. EXPANDING MY RESEARCH NETWORK

Another important benefit of being a CAS student is networking. The Centre for Advanced Studies of IBM Canada brings together a prestigious network of Canadian and international researchers working on software engineering, computer science, and information technology topics. This is a very special community, not only for the qualities of its members, but also for its commitment to mentoring graduate students and young researchers. CASCON is a prime example of this. Presenters are usually graduate students who receive valuable and constructive feedback from all professors in the audience. In contrast to other international venues, CASCON is recognized for being a "conference of friends" rather than a "conference of competitors". This provides PhD students with an environment that invites them to present their work and collaborate with other students and professors.

CAS University Days and CSER meetings are also a luxury for graduate students. Sessions at University Days are very useful for students to identify research problems that can be incorporated into their theses. CSER meetings provide students with the best venue to present their early ideas and results and improve their communication skills.

The networks built in these communities are crucial for the professional development of researchers—including PhD students.

7. CONCLUSIONS

The Centre for Advanced Studies of IBM is a highly successful research model. I am so grateful to have had the chance to be an integral part of this unique collaborative R&D environment. This experience influenced my research career profoundly. As a professor in Colombia I envy the Canadian professors who work on research projects with CAS. A great challenge for me is help establish CAS-like research organization in Colombia.

8. ACKNOWLEDGMENTS

I want to thank Professor Hausi A. Müller for being an excellent mentor, for all his patience while I was learning how to conduct research through our CAS projects. I would like to thank all professors who are part of the CAS community for all their feedback and advice I received during my PhD studies. I am especially grateful to the Center for Advance Studies (CAS) of IBM Canada for the opportunity of being a CAS student. My three-year experience at CAS allowed me to exchange valuable knowledge with practitioners, and inspired me with ideas for the application of my research to industry. I would like to thank CAS Director Joanna Ng, as well as CAS Research Staff members Alex Lau, Diana Lau, Tinny Ng, and Jimmy Lo for their support in all aspects of my CAS projects. Last, but not least, I want to thank all graduate students who were part of the CAS community during my PhD studies for their feedback, support, and all the moments we shared at CASCON conferences, CSER meetings, and University Days.

REFERENCES

[1] M. Chignell, J. Cordy, J. Ng, and Y. Yesha (Eds.) *The Smart Internet: Current Research and Future Applications.* Springer (Jan. 2010)

[2] Villegas, N.M. and Müller, H.A. 2010. Managing dynamic context to optimize smart interactions and services. *The Smart Internet,* pp. 289-318, Springer (Jan. 2010)

[3] Müller, H.A., Villegas, N.M., Lau, A. and Jou, S. Dynamic context-aware applications: approaches and challenges. *Proceedings 20th Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2010),* pp. 404-406, ACM (Nov. 2010)

[4] Villegas, N.M., Müller, H.A., Munoz, J.C., Lau, A., Ng J. and Brealey, C. A dynamic context management infrastructure for supporting user-driven web integration in the personal web. *Proceedings 21st Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2011),* pp. 200-214, ACM (Nov. 2011)

[5] Ebrahimi, S., Villegas, N.M., Müller, H.A. and Thomo, A. SmarterDeals: A Context-aware Deal Recommendation System based on the SmarterContext Engine. *Proceedings 22nd Conf. of the Center for Adv. Studies on Collaborative Research (CASCON 2012),* pp. 113-127, ACM (Nov. 2012)

[6] Muñoz, J.C., Tamura, G., Villegas, N.M. and Müller, H.A. Surprise: User-controlled Granular Privacy and Security for Personal Data in SmarterContext. *Proceedings 22nd Conf. of the Center for Advanced Studies on Collaborative Research (CASCON 2012),* pp. 128-142, ACM (Nov. 2012)

[7] Villegas, N.M. *Context Management and Self-Adaptivity for Situation-aware Smart Software Systems.* PhD Thesis, Department of Computer Science, University of Victoria (Feb. 2013) http://hdl.handle.net/1828/4476

[8] Villegas, N.M., Tamura, G., Müller, H.A., Duchien, L. and Casallas, R. DYNAMICO: A reference model for governing control objectives and context relevance in self-adaptive systems. *Software Engineering for Self-Adaptive Systems II,* Springer, pp. 265-293 (Jan. 2013)

[9] Tamura, G., Villegas, N.M., Müller, H.A., Duchien, L. and Seinturier, L. Improving context-awareness using the DYNAMICO reference model. *Proceedings 8th ACM/IEEE International Symposium on Software Engineering. for Self-Adaptive and Self-Managing Systems (SEAMS 2013),* pp. 153-162 (May 2013)

[10] Villegas, N.M. and Müller, H.A. The SmarterContext ontology and its application to the Smart Internet: A Smarter Commerce case study. *The Personal Web: A Research Agenda.* M. Chignell; J. Cordy, R. Kealey, J. Ng and Y. Yesha (Eds.). Springer, pp. 151-184 (Jul. 2013)

CAS Projects: Distributed Systems

Building Bridges

Experiences and Lessons from CORDS: A CAS Research Project Spanning Multiple Academic and Industrial Research Groups [*]

Michael Bauer
Department of Computer Science
The University of Western Ontario
London, Ontario
bauer@uwo.ca

ABSTRACT

The Consortium on Research on Distributed Systems (CORDS) was one of the first, and perhaps largest, projects undertaken within in the Center for Advanced Studies at the IBM Toronto Laboratory. The project began shortly after the founding of the Centre and engaged academics and graduate students from a number of institutions and a number of IBM staff and researchers from the IBM Toronto Laboratory, the Center for Advanced Studies and T.J. Watson Research. This paper provides a brief background on CORDS and some personal reflections on the impact of the project and collaborations on individuals.

Categories and Subject Descriptors

K.2 [**History of Computing**]: People; K.7.2 [**The Computing Profession**]: Organizations

General Terms

Management

Keywords

The Centre for Advanced Studies (CAS), Consortium on Research on Distributed Systems (CORDS)

1. THE BEGINNING

It was 1990 and the Centre for Advanced Studies (CAS) at the IBM Toronto Laboratory had just come into existence. With the vision of Larry Achtemichuk. Vice-President of IBM and Director of the IBM Toronto Laboratory, and under the leadership of Dr. Jacob Slonim, CAS was finally a reality. CAS was established at the IBM Toronto Laboratory as a means to bring academic and industry researchers and developers together to explore solutions to existing problems and to look forward to anticipate new directions and challenges emerging in the world of information technology. It was a conduit between these two groups – IBM tools and expertise could be made accessible to academic researchers and IBM researchers and developers would have access to emerging ideas, technology and talent. Jacob Slonim was busy promoting CAS and engaging IBM staff and was busy identifying potential collaborators among the academic community in Canada and the United States.

Jacob had, of course, already been busy reaching out to the academic community after joining IBM and had already found a number of academic collaborators. During these early days of CAS, projects were just beginning to be formed, but Jacob wanted to bootstrap the interactions and projects through CAS, and recognized an opportunity to go beyond individual projects to something more far-reaching. He wanted to enable a large scale project that would address significant problems in an emerging area, encompass a number of researchers, cross research boundaries and be noticeable.

At this time, networks were well established and there was a great deal of interest in networks of systems and the emergence of distributed applications. The means of building reliable and scalable distributed applications were primitive, typically requiring custom code and one-of software development. The development environments were immature, the tools limited and the underlying services needed were still research areas themselves. There were many opportunities for some innovative research – just the kind of domain Jacob was seeking.

After consultation with colleagues and potential collaborators, a small group was pulled together to focus on distributed systems, underlying services and the development, deployment and management of distributed applications. The project, CORDS – Consortium on Research on Distributed Systems, was conceived. A proposal was written and submitted to NSERC, the Natural Sciences and Engineering Research Council of Canada, for funding. CORDS was ambitious with a number of interacting and interdependent subprojects, made use of a variety of existing tools, and looked to develop new tools and interfaces. It involved a large number of researchers, graduate students, IBM staff and resulted in a large number of papers, many of which were presented at CASCON (CAS's annual conference) and published in its proceedings. It also produced multiple prototypes for demonstration of ideas at CASCON. One aspect of CORDS, perhaps unappreciated at the time but perhaps with the greatest long-term impact, was the mixing, interaction and working together of different individuals and research groups. CORDS created an opportunity for individuals to establish linkages, work through problems, cross boundaries, and really work together – it created an environment for building bridges.

2. RESEARCH SCOPE

To understand and appreciate the environment for research created by the CORDS project, we must first take a quick look at the scope of the research that was to be addressed. The project was based on the realization that the next decade would see a dramatic growth in distributed systems and a concomitant growth in distributed applications. The broad aim of the project was to explore directions and technologies to a) support the development and management of distributed applications, in particular: languages and programming tools, b) programming environments that provide an integrated set of facilities for visualization, modelling and managing the applications and data.

The proposal brought academic and IBM researchers together, built on their expertise and prototype software, and focused on an environment for developing and managing distributed applications. The environment was to enable the development of distributed applications using programming constructs based on the process model of programming [1, 6], provide application support services and enable the modeling, monitoring and management of the distributed applications. The research would proceed in four complementary areas:

- Languages and tools supporting programming in the process model. This would use and explore the use of Hermes and Concert/C, extensions to include SQL, and distributed debugging (***Lang***).
- Modeling and monitoring applications and services, including visualization, integration of simulation and modeling, monitoring and network management, and distributed management information bases (***MandM***).
- Applications support services, which primarily addressed database services, naming and directory services (***AppSS***).
- Runtime services, which focused on optimistic recovery, controlled mapping of language constructs to runtime services, multiprocessors and threads (***Run***).

There was also a strong push to build demonstration distributed applications (***DistrApps***) using the tools, frameworks and services developed through the project.

An initial project and proposal writing team was assembled and principal investigators identified. Table 1 summarizes the initial group of principal investigators and their affiliation.

The project was funded, though it had already begun through some initial funding provided through the CAS. The project itself in its scope and institutional engagement was even new to NSERC with eventual positive impacts. The consortium would also quickly expand, bringing in new groups with like expertise.

From the outset, at the proposal stage, CORDS was designed to facilitate interactions. The initial research areas and projects would involve multiple parties, with overlapping subprojects, including software dependencies. The interdependencies among research areas and institutions are illustrated in Table 2.

Of course, planned interactions laid out in a proposal can be quite different than actual interactions.

3. CONSTRUCTS FOR BUILDING

Yes, there were planned interactions and dependencies. But CORDS, the project, driven by the researchers and supported through CAS, created an environment in which interactions were enthusiastically embraced. Much of this can be attributed to the project itself and those involved. But there were a number of mechanisms that also fostered interactions. First, and foremost, the project had an overarching vision, albeit general, where there were exciting and interesting problems to address. The researchers saw that research questions beyond their individual groups could be addressed and that there was the potential for significant impact. Eventually, most of the graduate students involved in the project also saw this. Many of those involved could see that the whole, the capabilities, contributions, insights, would greatly exceed the sum of the individual parts.

Second, there was a concerted effort to exchange and distribute existing software tools and components, to incorporate the work of others into one's own prototypes, to provide feedback and to even help extend prototypes developed elsewhere. There was an acceptance of the work of others. This fostered discussions, interactions and furthered communications. This easily led to publications with a number of authors, a new experience for many of us involved (see, for example, papers on CAS and CORDS Architecture [2, 3, 5]).

Through CAS, graduate students were brought together. Faculty involved in the project were able to visit CAS and had the opportunity to work with and exchange ideas with other faculty and with students from other institutions – CAS became an exchange point for sharing ideas and working together. The students benefited from the collocation at CAS. They got to know one another, to build friendships and recognize one another's expertise and skills.

As the project progressed, there were opportunities to create prototypes based on existing software and software developed in the context of the project. Prototypes were often undertaken by the students at CAS, where they were able to work together, bringing pieces of software together to realize some novel system or application. The demonstration of

Table 1: CORDS Initial Principal Investigators

Principal Investigator	Affiliation
Michael Bauer	The University of Western Ontario (UWO)
Paul Larson	The University of Waterloo (Wat)
Alberto Mendelzon	The University of Toronto (Tor)
C. V. Ravishankar	The University of Michigan at Ann Arbor (UM)
Jacob Slonim	Centre for Advanced Studies, IBM Toronto Laboratory (CAS)
Toby Teorey	The University of Michigan at Ann Arbor (UM)
Shaula Yemini	IBM T.J. Watson Research Center (IBM-TJW)
Yechiam Yemini	Columbia University (Col)

Table 2: CORDS Research Areas and Institutional Participation

	UWO	Wat	Tor	CAS	UM	IBM-TJW	Col
[Lang]	√	√		√		√	√
[MandM]	√		√	√	√	√	√
[AppSS]	√	√			√	√	√
[Run]	√	√				√	√
[DistrApps]	√	√	√	√	√	√	√

prototypes at CASCON became a focal point. Students, and often faculty, spent the day or evening (and sometimes late into the night) assembling prototypes and building working demonstrations, usually complaining about how well "it worked in the laboratory". While occasionally stressful, they were unique educational opportunities and, looking back, actually fun.

4. BRIDGES

For many of us CORDS was an unprecedented opportunity to be deeply involved in a substantive, extended research project. It was an experience in which the value of large-scale research with multiple collaborators was evident. It was a project that helped build and hone skills of both researchers and students in working in large teams.

CORDS was a project that ran for three years and the researchers and students involved produced papers, software, and demonstrated prototypes. Based on the 1993 CASCON proceedings [4] which showcased a wide range of CORDS output, there were 21 papers with a member of the CORDS project team as a co-author of the 87 published in the proceedings. The proceedings itself lists the participants in the CORDS project: 28 faculty, 15 from IBM, 55 students. CORDS had continued to grow from its original concept.

While the project was, from a number of perspectives very successful, for me, personally, it was what I learned, my experiences and the bridges that I built that are the most valuable and most long lasting. I learned a great deal about participating in and managing large scale distributed research projects and the value of an overarching vision. I learned about what could be accomplished with a strong team. I also learned the value of building a team of individuals that are both strong in research and strong in a willingness to collaborate and cooperate. With such a team of individuals, I learned to trust them to work toward the overarching goals of the project which meant frequently letting them "do their thing" re research. The experiences and ties built during CORDS led to a follow-on project – MANDAS (MANagement of Distributed ApplicationS). This project involved some of the researchers involved in CORDS and others. Other projects, some with CAS participation and some without, followed as well.

Besides my personal growth, the most valuable benefits of participating in CORDS were those personal connections with others in the project. CORDS enabled the connections and many of those "bridges" are still in place:

- Ongoing interactions with Jacob Slonim while he was at IBM, after he moved to Dalhousie and after that.
- Ongoing collaboration with Pat Martin and Wendy Powley of Queens University on subsequent research projects.
- Use of CORDS participants as external examiners for PhD Theses, including Pat Martin, Jerry Rolia, Murray Woodside, which also provided opportunities to catch up on research.
- Continued interaction with Kelly Lyons and Andrew Marshall, who were students working on the CORDS project and James Hong who was a research assistant on the project. Kelly became Director of CAS and has since moved on as a faculty member at the University of Toronto. Andrew became a security expert at TD Bank Financial Group and is now a security consultant. James became a faculty member at Postech University in Korea.
- A resource pool of expertise, including those mentioned above, but also Toby Teorey (University of Michigan), Yechiam Yemini (Columbia University), Paul Larson, David Taylor and Jay Black (University of Waterloo).
- Dan Sinai, who was the NSERC program officer that had to deal with the CORDS investigators and sites and was an advocate for the project. Dan continued at NSERC for a number of years, and reports that CORDS, even with all of its complexity, had a

> significant influence on subsequent NSERC industry-academia programs. Dan eventually left NSERC and joined Western as part of a National Center of Excellence and became Associate Vice-President Research where we worked closely and productively for many years (as we did within the scope of the CORDS project). Dan recently left Western to join IBM.

While I have not personally canvassed others involved in the CORDS project, I am pretty sure that many of them would also identify their own, long-lasting "bridges".

5. CONCLUSION

CORDS was an innovative, complex, challenging, collaborative project ... that was FUN (though I am sure that the characterization of the project as "fun" might not have occurred to many of us involved at the time we were in the throes of deploying a demonstration system!). It taught me a great many things about projects, about teams and about collaborators. Most importantly, it was the means to build a number of long-lasting relationships and personal connections.

6. REFERENCES

[1] D. Bacon and R. Strom. Implementing the hermes process model. Technical Report RC 14518, IBM T.J. Watson Research Center, Yorktown Heights, NY., 1989.

[2] M. Bauer, N. Coburn, D. Erickson, P. Finnigan, J. Hong, P. Larson, J. Pachl, J. Slonim, D. Taylor, and T. Teorey. A distributed system architecture for a distributed application environment. *IBM Systems Journal*, 33(3):399–425, 1994.

[3] M. Bauer, P. Finnigan, J. Hong, J. Rolia, T. Teorey, and G. Winters. Reference architecture for distributed systems management. *IBM Systems Journal*, 33(3):426–444, 1994.

[4] A. Gawman, W. M. Gentleman, E. Kidd, P.-A. Larson, and J. Slonim, editors. *Proceedings of CASCON'93*. IBM Canada Ltd. Laboratory, Centre for Advanced Studies and the National Research Council of Canada, 1993.

[5] J. Slonim, M. Bauer, P. Larson, J. Schwarz, C. Butler, E. Buss, , and D. Sabbah. The centre for advanced studies: A model for applied research and development. *IBM Systems Journal*, 33(3):382–398, 1994.

[6] R. Strom, D. Bacon, A. Goldberg, A. Lowry, D. Yellin, and S. Yemini. *Hermes: A Language for Distributed Computing*. Prentice-Hall, 1991.

Reflections on Industry-Academic Collaborative Research

Hanan Lutfiyya
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada
hanan@csd.uwo.ca

ABSTRACT

The IBM Centre for Advanced Studies was established in 1990 at the IBM Toronto Software Laboratory. In the last 25 years, this lab has played a significant role in promoting and participating in industry-academic collaborations in Canada and worldwide. The CAS model established in the 1990's resulted in an exciting and successful model to support collaborative research. This paper provides a brief background on my experiences and reflections on this model of collaborative research.

1. INTRODUCTION

Early in my academic career, Mike Bauer, invited me to join a project with the title "MANagement of Distributed Application and Systems" or MANDAS. This project was a follow up to a project titled "Consortium for Research on Distributed Systems" (CORDS). Both projects were among the first projects between the IBM Centre for Advanced Studies and academic researchers. I was a junior professor who had done primarily theoretical work in a systems area and had very little experience with industrial collaborations and was concerned that it would be a distraction from research needed for tenure.

My concerns were unfounded. The research was challenging resulting in a number of publications and there was an excellent team of people to work with including Mike Bauer, Jay Black, Rick Bunt, Pat Finnigan, Thomas Kunz, Andrew Marshall, Pat Martin, Wendy Powley, Jerry Rolia, David Taylor, Murray Woodside and numerous graduate students.

It was a wonderful experience where I established long term friendships, considerably enhanced my research horizons and had fun.

Since then I have had other CAS projects and other team-based projects that I have enjoyed. I have also had many wonderful interactions with other CAS researchers who were not part of any CAS project I was on.

2. A MODEL OF RESEARCH COLLABORATIONS

I have enjoyed all team based projects that I have worked on but the MANDAS project stands out. IBM provided seed funding for three years to support PhD students, which was matched with NSERC CRD grants, significant employee time and space at the IBM Toronto Software Laboratory to facilitate interactions.

The MANDAS research was divided into a set of projects to be worked on by smaller groups. It is often easy to take the money and work on your own prototypes and give a "wave your hands" explanation that the project prototypes are sufficient validation for the overall architecture. This did not happen.

The MANDAS project took the approach that the team would develop a prototype that incorporated work from each group. The prototype would be a demo at the IBM Centers for Advanced Studies conference commonly referred to as CASCON. The PhD students funded by CAS worked together during the summer at IBM on their prototypes and then the integration of their prototypes for the demo which allowed for rapid refinements and validation. Faculty members visited IBM to interact with students, other faculty and IBM employees. This synergy was very beneficial to the research.

Other groups were using a similar format and so CASCON rapidly became Canada's most important forum for the presentation of exciting work taking place throughout Canada.

The results can be seen with the high number of graduate students who are trained to consider both academic and industrial perspectives, a high number of publications and patents, and innovations to products as well as new products.

3. REFLECTIONS

In recent years there has been a considerable change in the funding landscape with an increased emphasis on transforming scientific research into competitive advantage that leads to successful technological innovation and economic growth. The CAS model described earlier has all the elements needed: teamwork, multiple year projects, seed funding, access to industrial expertise and common goals derived from reflections on long term strategies as opposed to short goals associated with the next version of products.

No model is perfect. One possible enhancement could include having industry employees spend time in academic labs to work with large interdisciplinary teams that will be needed in the development of new healthcare systems, power grids, etc.

There are significant challenges in establishing this type of model, e.g. management buy-in. However, CAS has shown the benefits to universities and industry. It is time to reflect on existing and new models of collaboration between academic and industry in moving forward in the next 25 years.

CORDS, MANDAS, GLAD and Beyond: Our CAS Experience

Patrick Martin
School of Computing
Queen's University
Kingston ON K7L 2N8
613 533-6063
martin@cs.queensu.ca

Wendy Powley
School of Computing
Queen's University
Kingston ON K7L 2N8
613 533-6734
wendy@cs.queensu.ca

ABSTRACT

This paper highlights the authors' experiences working with IBM Centers for Advanced Studies (CAS) from 1992 to 2015.

1. The Early Projects

In 1992, Queen's University became a CAS partner university. Pat Martin joined the Consortium for Research on Distributed Systems (CORDS) project as a PI and David Rappaport and Kelly Lyons (a CAS PHD student) were working on a project called Graph Layout Algorithm Display (GLAD). Pat and David hired Research Associate Wendy Powley to assist on both projects. What was to be a short-term 2 year partnership with CAS has turned into a 24 year collaboration between CAS and Queen's and has shaped the careers and futures of all involved.

GLAD was associated with Kelly Lyons' PHD thesis. In this project, Kelly was developing graph layout algorithms which would be used to visualize data sets. Wendy's task was to write a "state of the art" visualization tool which would display the data points and allow the user to rearrange the points of the graph on the screen. This was done using Motif under the X Window System and the results, shown in Figure 1, were, no doubt, the precursor to current day visualization tools.

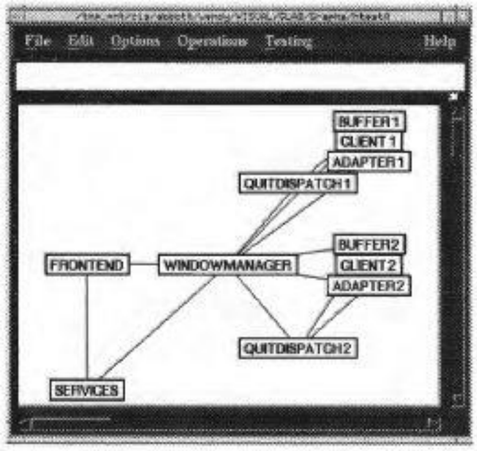


Figure 1: GLAD Graph Representation

The GLAD project ended in 1994 when Kelly graduated as "Dr. Lyons" and returned to IBM. The project may have ended, but the lifelong friendship and the "GLAD" jokes between Wendy and Kelly continue to this day.

The first project that Pat and Wendy were involved in was CORDS, a project which focused on the development, operation and management of distributed systems. The CORDS project was led by Jacob Slonim and Pat Finnigan at CAS and involved more than 100 students and researchers at IBM and 13 universities (seven Canadian and six in the US). In an early paper by Jacob Slonim, Michael Bauer and Paul Larson [1], many broad objectives were stated for the CORDS project including advancing the state of the art, understanding the complexities and issues in distributed systems, knowledge transfer, and adding value to the products developed by the IBM Toronto Laboratory. Although no doubt all of this (and more) was accomplished, this project more importantly laid the foundation for several life-time working relationships and lasting friendships.

A group of CORDS researchers with a common interest in distributed systems management (a more detailed focus than CORDS) continued working with CAS members and spawned a new project called "MANDAS" – MANagement of Distributed Application and Systems. This project involved many of the CORDS researchers and brought many newcomers to CAS from across North America.

The CORDS and MANDAS projects themselves were distributed systems with researchers scattered across North America (and beyond). The distributed nature of our team led to the ability to develop truly distributed applications on which we could study and evaluate our solutions to management.

Being the Database group at Queen's, the focus of our work was primarily on the data management aspects in each of the projects. In order to manage a system, data must be collected, stored, queried and analyzed. Little did we know that we were looking at "Big Data". For the CORDS project, we developed a "multidatabase" system (MDBS) allowing integrated access to heterogeneous data sources such as directories, different types of databases, and file systems. This required delving into 3 binders filled with ODBC documentation which was hot off the press in the early 90's. We also explored the use of the X.500 directory service to store the catalog information for the MDBS. We exploited the Telos system [2] in MANDAS to build and query a meta-database of information used for distributed systems management.

2. CASCON

Many papers were published by the researchers of CORDS and MANDAS outlining the tools and technologies developed by the group. These papers now reside in their space on the World Wide Web, a place that barely existed when we first started working together. No permanent record, however, was made of the numerous CASCON demos that were done to showcase the work on these projects. It was the teamwork required for the demos that brought us together and solidified our working relationships. The demos forced the groups to communicate and interact. Building a working prototype ensured that we worked out every detail and solved problems that we hadn't yet considered.

The CASCON demo was a yearly goal of the CORDS and MANDAS teams. By October, the groups' individual pieces were expected to fit together to showcase our work. With today's networked infrastructure and sophisticated communication, this probably would have been trivial. However, in the early 90's, we were faced with many barriers. Standard communication was unpredictable. Connecting one module sitting at Queen's with a module at Waterloo was certainly possible, but the infrastructure didn't always function the way we expected, and the details were difficult to figure out. Stack Overflow was not yet available to solve our problems. Instead, we had to resort to documentation (which was often in the form of books which were difficult to obtain), trial and error, or brute force debugging. Communication between researchers was mainly via email and phone (with expensive long distance charges). Remote meetings among group members were virtually impossible. Skype would have been a great distributed application for us to experiment with as well as to use for meetings had it existed.

The solution to our problems was to a) impose very stringent coding standards on the programmers developing the code to ensure compatibility and ease of debugging and b) invite all the technical people to meet at David Taylor's lab in Waterloo and put the entire demo on machines in a central location. This greatly diminished our problems, however, this solution was ironic considering the topic of our research. Even being centrally located, with all the technical people present, it generally took at least two days to get everything set up and communicating flawlessly.

CASCON demos were the highlight of the year for many of us, and the pre-conference set-up at CASCON was an event in itself. Hardware was transported to Toronto and assembled in the demo hall. In the early years, we took one of the Queen's systems specialists with us to help set up our machines and to configure our network. If we required a connection to the outside world, it was necessary to make a special request and it was risky to count on it working. Contrast this with today's expectation that each of our many individual devices will connect instantly (and wirelessly) and that we will have unlimited internet access for the duration of the conference.

Once the hardware was set up, team members would gather together to ensure that the demo worked. Despite our best efforts at Waterloo to achieve a working demo ahead of time, there were always many hours of coding, testing, debugging and hacking to make things work once we arrived at CASCON. It was expected that the night prior to the technology showcase opening, at least a few of us would be pulling an "all-nighter" to make the demo work by morning. Others would often just stay for moral support – and to make (or take) frantic phone calls to (from) the PIs. Although no one was seriously harmed during CASCON preparation, there were at least a couple of hospitalizations. We took our jobs very seriously!

In the early 90's under Jacob Slonim's direction, CASCON was a formal affair. There were strict requirements for how the demo hall looked. Signs were (literally) perfectly aligned. No chairs were allowed anywhere in the demo hall. Those at the booth were to be dressed formally (suits for all) and standing to greet visitors. Demos began shortly after 7AM with VIP tours and continued late into the evening. Booths were staffed continuously during the day and there was never a shortage of interested parties wandering around the hall. Demos were not to be demolished even one minute ahead of closing time despite the fact that we were all exhausted.

Given the size of the CORDS and MANDAS projects, our group always had a huge presence at CASCON. We joked about the "wall of MANDAS" - a dedicated section for posters and demos related to the MANDAS project. Being grouped together in the demo hall only added to our comradery and we shared many good times.

As the years went by, the Technology Showcase (and CASCON in general) became a little less formal and the rules relaxed somewhat. Pregnant women were allowed chairs. The "required" demo hours were changed to coffee breaks, lunchtime and other (short) designated times. Dress became more casual. Although the rules have relaxed, the atmosphere that Jacob created of a truly professional event has persisted to this day.

The Queen's University Database lab has been very visible at CASCON throughout the years. In addition to the numerous demos and posters that our group has presented, we have organized several workshops and have had many of our papers accepted for presentation. We have won two best paper awards and two best student paper awards.

3. The Later Years

Following the MANDAS project, Pat and Wendy continued working with CAS on a number of projects with the first being "no knobs operation" for database tuning. The goal of this project was to make DB2 "self-tuning" or, in other words, "self-managing". Once we discovered how many tuning parameters there actually were, we began with automatic tuning of multiple buffer pools.

IBM coined the term "Autonomic Computing" in 2001. An autonomic system is one that is self-aware and self-managing which described our "no-knobs operation" research perfectly. For the following years we explored autonomic computing principles in the areas of database systems, web service environments, and eventually cloud computing. An area of particular interest was the application of autonomic computing principles to database workload management, a topic that we have explored in depth with the help of several IBM RSMs including Paul Bird, Berni Schiefer, Keri Romanufa, Haider Rizvi, Sam Lightstone and Randy Horman. Pat's current CAS project has moved into the big data realm and is on the topic of workflow orchestration for Analytics-as-a-Service. The Database lab, along with Dan Rope, Craig Statchuk and Mike McRoberts of IBM, are developing a framework that will allow users to perform new and complex data analysis by composing multiple analytic products into an executable workflow.

4. CAS Impact

Our association with CAS has had tremendous influence on our careers. For Wendy, it has secured her job as a Research Associate with Pat for 24 years, something that is virtually unheard of in the academic world today. Our successful working relationship with CAS has, without a doubt, been instrumental in Pat's success to secure funding from outside sources. We have been fortunate to attract top-rank graduate students as they are enticed by the opportunity of spending summers at IBM and the lure of industry contacts. Over 40 graduate students have worked on CAS-sponsored projects over the years and 18 of these students have gone on to positions with IBM.

The early relationships formed at CAS were lasting and have resulted in many partnerships beyond CAS and IBM. We have worked with Gabby Silberman, former Head of CAS, on projects at CA Technologies and Dell Research. We are currently involved in a five year, multi-university project with Mike Bauer and Hanan Lutfiyya, two of our partners on the CORDS project. The majority of the eleven PIs on this project have been involved with CAS at some point in their career and many of our Advisory and Scientific Committee members are familiar faces from CAS including Stephen Perelgut, Vic DiCiccio, Calisto Zuzarte, and Hausi Müller. We have organized a workshop at CASCON annually since the beginning of this project in order to share our work with the larger CAS community.

The connections established through CAS have provided Pat with other exciting opportunities. He is a member of the Scientific Advisory Committee for the Southern Ontario Smart Computing Innovation Platform (SOSCIP) and ran one of the first research projects using the SOSCIP cloud resources. He is also currently teaching a new course at Queen’s University in Cognitive Computing using IBM’s Watson.

Pat and Wendy were early adopters of DB2 for course use and Queen’s was one of the first university members of the Academic Initiative program. DB2 has been in use at Queen’s for teaching database courses for nearly 20 years. In 2006, in her role as CAS Head, Kelly involved Wendy in an initiative called the Shared Software Infrastructure Hub. The goal of this project was to provide infrastructure to universities to allow them to use leading edge software for their courses. Wendy met and worked with researchers and IBM staff members across North America and contributed a database course based on DB2.

The Ontario Celebration of Women in Computing (ONCWIC 2010 – 2015) conference, now the Canadian Celebration of Women in Computing (CAN-CWiC), a nation-wide event, resulted from conversations and collaboration between Wendy, Kelly Lyons and Gabby Silberman. Three of the five chairs of ONCWIC have been CAS affiliates. IBM has sponsored the conference since its inception and is a Platinum Sponsor for CAN-CWiC 2016. Stephen Perlegut and Judy Huber have been instrumental in these efforts. Stephen, in his new role as IBM Canada Ecosystem Business Development Manager is planning a Canada-wide programming competition using IBM's Bluemix with the finals being held at CAN-CWiC 2016.

The people that we have met through CAS have become life-long friends which we look forward to meeting and reminiscing with each year at CASCON and other international conferences. They have become partners in many varied endeavors. These friends are those who come to mind when forming a committee or a research team, when searching for an external thesis examiner, when we need someone to give a talk (or a roast speech), when we are in need of a reference, or if we are seeking advice. Because of CAS, we have contacts from all across Canada, at most universities and in many companies. CAS has formed our work family and we will carry with us fond memories of all the times that we have shared throughout the years. Many thanks to IBM and all the CAS Heads - Jacob, Gabby, Kelly, Joe and Joanna – for making this such a wonderful journey.

5. REFERENCES

[1] Slonim, J., Bauer, M., Larson, P. CORDS: Status and Directions, Proceedings of the 1992 Conference of the Centre for Advanced Studies on Collaborative Research, Volume 2, Pages 1-21.

[2] Mylopoulos, J., Borgida, A., Jarke, Matthias, J. and Koubarakis, Telos: Representing Knowledge about Information Systems. ACM Transactions on Information Systems (TOIS), Volume 8(4), October 1990, pages 325-362.

CAS Projects: Software Evolution

Impact and Synergy in the CAS R&D Ecosystem

Hausi A. Müller
Department of Computer Science
Faculty of Engineering
University of Victoria
hausi@cs.uvic.ca

ABSTRACT

The impact and synergy generated over the past 25 years by the IBM Centre for Advanced Studies (CAS) within IBM Corporation, the Canadian computer science and software engineering research community, Canada's IT sector, and the rest of the world have been truly spectacular. I am honored and privileged to have been deeply involved in this unique collaborative R&D ecosystem and had a chance to observe its evolution and significance up close and personal. In 1992/93, fortuitously, I spent my first sabbatical in CAS at IBM Toronto Lab. This experience influenced my research career profoundly. I participated in every CASCON conference because I realized early on how lucky we are that IBM started CAS and CASCON in Canada and not somewhere else in this world. I experienced the enthusiasm generated by CAS projects, the synergy between IBM developers, CAS Research Staff Members, professors and students as well as how CAS PhD students grow and excel in this stimulating environment.

Keywords

Impact, synergy, significance, software engineering, computer science, collaboration, research, R&D, Center for Advanced Studies, CAS, CASCON conference, IBM, Canada, collaborative research, CAS University Days, pre-competitive technology development, research projects, CAS fellowships, CSER, HQP, PhD students, NSERC, CRD projects.

1. THE OVERTURE

In the fall of 1991 Jacob Slonim toured Canadian universities from Bonavista to Vancouver Island to recruit faculty, students, and projects for the newly founded IBM Centre for Advanced Studies (CAS). He did reach the famous island in the middle of the Pacific and visited the Department of Computer Science at University of Victoria. He gave a talk on research models, research partnerships, pre-competitive technology development, and how to connect IBM developers to university researchers in the CAS model [1, 2].

We proudly showed him our Rigi program understanding tool. At the end of the visit I casually mentioned that I was planning for a sabbatical at the University of California, Irvine the following year. Jacob, in his most diplomatic way, said: "No, you are not—you are coming to CAS in Toronto next year." I was a little surprised by his candid overture. However, it worked—I went to Toronto and have been working with IBM CAS ever since. The rest is history.

2. MY FIRST SABBATICAL AT IBM CAS

In August 1992 I leisurely drove my family in three weeks through the Rockies, across the Prairies, around Lake Superior to the big city of Toronto. We were very lucky to have found an apartment right on the Boardwalk in the beautiful Beaches. While we orchestrated a soccer game somewhere in North York with Jacob refereeing, the Blue Jays won their first World Series downtown. I even had the opportunity to sit in the IBM box in the SkyDome for a Blue Jays game. Oh, it was an exciting year in Toronto.

2.1 CAS Program Understanding Project

In 1991 IBM CAS was located on the top floor of a high rise at 895 Don Mills Road, North York. Kelly Lyons, who was a Queen's University CAS PhD student at the time and John Botsford, who was the CAS Research Principal Investigator, did a fabulous job looking after me at Big Blue and in the big city. John Schwarz was the Toronto Lab Director and really believed in CAS.

I worked closely with CAS Research Staff Members Eric Buss and John Henshaw (cf. Figure 1) on the CAS program understanding project [3]. They analyzed massive amounts of code written in PL/IX using the Software Refinery from Kestrel Institute running on machines with incredible amounts of memory. They worked closely with development groups and provided valuable but sometimes overwhelming information to them.

Figure 1. CAS Program Understanding Project with Research Staff Members Eric Buss and John Henshaw—CASCON 1992

In our first IBM NSERC Collaborative Research and Development (CRD) project I had the wonderful opportunity to work with John Mylopoulos, University of Toronto and Renato de Mori, McGill University (cf. Figure 4). Over the next decade several research groups from McGill University as well as Universities of Michigan, Toronto, Victoria and Waterloo worked cooperatively on complementary program comprehension and reverse engineering approaches [3] (cf. Figure 11).

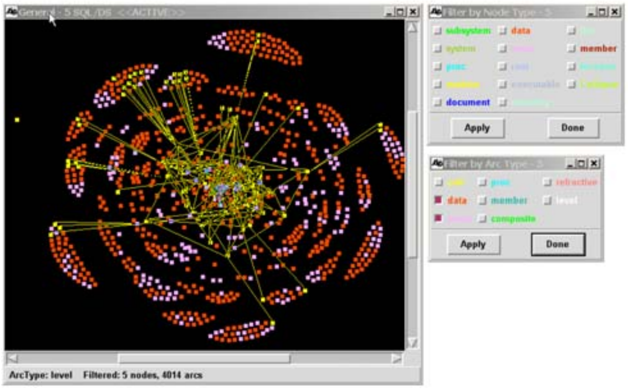
All groups used the source code of SQL/DS (a multi-million-line database system) as the reference legacy system. One of the objectives was to integrate the various toolsets under a single reverse engineering environment [11]. One of the results of this intense collaboration was the highly cited Software Bookshelf IBM Systems Journal paper [14]. Ric Holt's PBS (Portable Bookshelf) and our Rigi environment were very successful realizations of this vision. By 1999 Canadian research groups dominated international program understanding and reverse engineering conferences such as *IEEE Working Conference on Reverse Engineering (WCRE)* and *IEEE International Conference on Program Comprehension (ICPC).* In 2000 we were invited to contribute a roadmap paper in the *Future of Software Engineering* volume commemorating the new millennium [18]. In 2004 Ahmed Hassan and Ric Holt, University of Waterloo, created the now highly recognized and acclaimed *ACM/IEEE International Workshop on Mining Software Repositories (MSR).*

2.2 Rigi

Our Rigi tool was designed to explore the architectural aspects of a system under analysis [5, 6, 7]. We developed a method to identify, build, and document layered subsystem hierarchies [6]. Critical to its usability was the tool's ability to store and retrieve reverse engineered views. Rigi also supported human- and script-guided structural pattern recognition [8, 9, 10].

Figure 2. CASCON 1993 — Rigi Demo: Brent Sauder, ASI Hausi Müller, Kenny Wong, Mike Whitney, Scott Tilley, UVic

While on study leave at CAS I analyzed SQL/DS using Rigi many times over (cf. Figures 2, 3 and 5). Producing visualizations that were meaningful and palatable to developers was a huge challenge as well as tedious and time consuming [7, 10]. I realized that support for automation is critical even for visual program understanding tasks. As a result, we extended Rigi with a scripting layer using the Tcl/Tk scripting language, the Rigi command language (RCL), and Rigi standard format (RSF) [8, 9, 12, 13]. The scripting layer made Rigi extensible, domain retargetable, and provided access to external tools—allowing Rigi to function as the cornerstone of a comprehensive reverse engineering environment [11, 14].



ArcType: level Filtered: 5 nodes, 4014 arcs

Figure 3. Rigi SQL/DS visualization

Many different students and research associates affiliated with my research group worked on Rigi [17, 23, 24, 25]. It evolved into a powerful and versatile environment for software reverse engineering, understanding, exploration, visualization, and re-documentation [10, 24]. Subsequently, several research groups from around the world built tools on top of Rigi including Dali (SEI), famous SHriMP (Storey), Bauhaus Rigi (Koschke), Shimba (Systa), Nokia Toolkit (Riva) and Klocwork suite (Klocwork Solutions)—now that is true impact.

Figure 4. CASCON 1994 — Software Program Understanding Research with researchers from IBM, NRC, Alberta, McGill, Michigan, Toronto, Victoria, Waterloo, and others

3. CAS as the Center of the World

On Vancouver Island Toronto is—for good reason—frequently referred to as the center of the world. From a Canadian informatics perspective, CAS is actually the center of the world. CAS, CASCON, and CAS University Days are fantastic places to meet and greet prospective graduate students, CAS research staff members with challenging problems and case studies, developers with industrial-strength data, aspiring researchers and faculty candidates, potential research collaborators and grant proposal writers, famous computer science and software engineering professors, and computing pioneers—in 2005 Kelly Lyons, Stephen Perelgut, Cheryl Morris and I orchestrated a highly acclaimed CASCON event to honor over 90 Canadian Pioneers of Computing (cf. Figure 1 in Kelly Lyons' article).

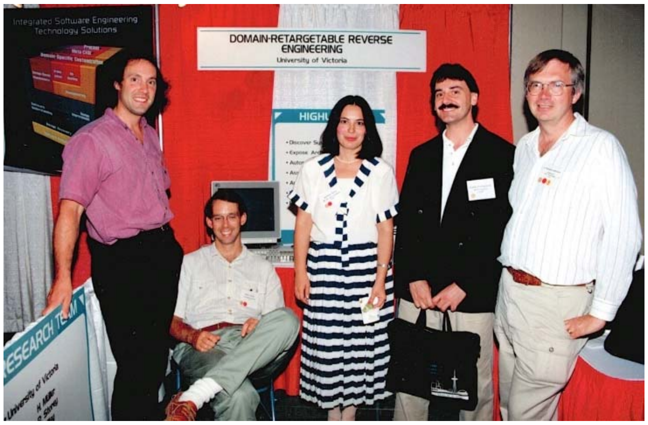

Integrated Software Engineering Technology Solutions
DOMAIN-RETARGETABLE REVERSE ENGINEERING
University of Victoria
RESEARCH TEAM

Figure 5. CASCON 1995 — Famous Tools Developers: Mike Whitney, Brian Corrie, Marijana Tomic, Kostas Kontogiannis and Howard Johnson

During my sabbatical at CAS I met many important people from academia and industry, who became great friends and opened up many doors for me. First and foremost, John Mylopoulos, a great mentor and friend (cf. Figure 6). In the early nineties he took me under his wings and we have collaborated ever since. In particular, we executed several highly successful IBM NSERC CRD projects. Other co-investigators and collaborators in these projects included principal investigators John Mylopoulos, Kostas Kontogiannis Kenny Wong and Martin Stanley, CAS Research Staff Members John Botsford, Polly McPherson, John Maillard, Nick Cooper, Joel Troster, Bill O'Farrell, Graham Ewart, Pat Finnigan, Jan Pachl, Ivan Kalas, Kelly Lyons, Ian Mcintosh, Bill Hay, Stephen Perelgut, Cheryl Morris, Marin Litoiu, Emilia Tung, Leho Nigul, Chris Brealey and Tim Francis. Since 1993 I have been a CAS Visiting Scientist and CAS Fellow. In 2006 I received the CAS Faculty Fellow of the Year Award.

Figure 6. John Mylopoulos, University of Toronto—Principal Investigator, Visionary, Leader, and Wonderful Mentor

After working with IBM CAS on program understanding and reengineering for over a decade, we worked on service oriented computing, SOA, smarter commerce, smart Internet, context management, and personal web tasking with Joanna Ng, Eleni Stroulia, Jim Cordy, Pat Martin, Wendy Powley, Jenny Zou, Jimmy Lo, Alex Lau, Diana Lau, Tinny Ng.

I am very happy to report that many of us are still working together. For example, John Mylopoulos, Marin Litoiu and I are organizing a workshop on cyber physical systems at CASCON 2015.

After lobbying for many years we finally in 2001 managed to attract the *ACM/IEEE International Conference on Software Engineering (ICSE)* to Canada. As General Chair I selected Toronto as the venue for ICSE 2001 because I knew I could count on the support of IBM and CAS. CAS Director Gabby Silberman and Anatol Kark, leader of the NRC Software Engineering Group, provided generous sponsorship support. Many other folks from the CAS ecosystem including the CASCON meeting organizer from Carlson Wagonlit helped greatly in organizing ICSE in Toronto at the Westin Harbour Castle. ICSE 2001 was a great technical and financial success with 1174 attendees.

In 2006 CAS and 12 universities received an NSERC Leo Derikx Synergy Award at a ceremony in Winnipeg in the middle of CASCON (cf. Figure 7). The NSERC Synergy Awards for Innovation were launched in 1995 by NSERC to recognize partnerships in natural sciences and engineering research and development (R&D) between universities and Canadian industry. There is no better partnership model than CAS for this prestigious NSERC award.

Figure 7. Kelly Lyons and Jacob Slonim accepting proudly the 2006 NSERC Leo Derikx Synergy Award on behalf of CAS

4. CASCON

The progress of CAS research projects is showcased and celebrated annually at the CASCON conference [41, 42, 44]. While CASCON is an international conference with much participation from the United States, it is also a true Canadian conference where Canadian computer scientists and software engineers gather in November attracting over 1300 practitioners and researchers. The CASCON venues over the years included North York (Toronto Lab, Four Seasons Inn on the Park, Radisson and Ontario Science Center), Mississauga (Airport International Conference Center), and Markham (Sheraton Parkway and Hilton Suites).

Figure 8. CASCON 2001 — Kenny Wong University of Alberta

This "Meeting of Minds" has evolved into a comprehensive research and academic forum to collaboratively formulate problem statements, shape research agenda and address issues of the latest emerging technologies that are of great academic significance for the research community and of strategic business value to the industry at large.

In 25 years, CASCON has evolved into an international academic and research conference where innovations from the academic, public and industrial sectors converge and inspire, where important research collaborative contacts are made and valuable research and industrial alliances are formed, making it a truly premier event in Canada. Hayat and Lyons recently studied the research networking impact of CASCON. They analyzed a specific network of researchers by looking at the CASCON conference paper co-authorship structure [44].

I always enjoy contributing to CASCON and taking in its collaborative and stimulating atmosphere. I participated in every single CASCON—because I realized early on how lucky we are that IBM started CAS and CASCON in Canada and not somewhere else in this world. I experienced the synergy generated by CAS projects and how CAS PhD students grow and excel in this environment. Every year I encourage my students to attend CASCON.

I was CASCON program co-chair in 2003 and 2010 with Anatol Kark and Arthur Ryman. To celebrate the 20th anniversary of CAS and CASCON, we created a special CASCON Anniversary Volume entitled CASCON First Decade High Impact Papers [27] and composed an anniversary song (cf. Figure 9).

4.1 CASCON Workshops

CASCON workshops are collaboration and incubation engines for new research topics, research alliances and partnerships, grant proposals and project meetings, as well as paper and book projects. In many ways the interactive workshops are the heart and soul of CASCON.

I always enjoy participating, interacting and running CASCON workshops. It is a unique atmosphere where practitioners have a chance to challenge academic researchers. Selected CASCON workshops I helped orchestrate cover a wide range of topics including the following: program understanding, software evolution, technology transfer, legacy software systems, adoption-centric software engineering, engineering autonomic systems, SOA governance and migration, root cause analysis and diagnosis, smart internet, dynamic context-aware applications, smarter commerce, self-adaptive systems, personal web tasking, and engineering cyber physical systems.

4.2 CASCON Demos

Being part of the CASCON exhibits has always been an important goal for my research group. In software engineering research you are more credible if you actually implement proposed methods and tools and validate the approaches on industrial data. Most of my students built applications and tools to validate their research and in turn demonstrated them at CASCON—including the first CASCON in 1991.

In the early days, it was a real challenge to log the big monitors and machines to the conference in Toronto. We spent many CASCON nights polishing demo scripts and dry runs before facing a tough industrial audience. Many students wore a tie for the first time showcasing their tools and results. My task was usually to run around and drum up customers for our demos and posters.

Figure 9. CASCON 2010 — Jim Cordy and Hausi Müller leading the sing-along in the 20th CASCON Anniversary Song

The stress level of the students—and CAS directors—always skyrocketed during the IBM executive walk-throughs. For a couple of years, we had the opportunity to demonstrate our software visualization tools in the middle of the exhibits space on giant monitors from Ontario Hydro. Once a Boeing developer, Ted Kitzmiller, challenged us to visualize the parts hierarchy of a Boeing 777. We boldly claimed that we can do it right here at CASCON. We downloaded a parts file from Boeing and told him to come back in a couple of hours. We wrote a few RCL scripts and produced the interactive visualizations in the nick of time. Ted was literally blown away and, as a result, we were invited to demo Rigi the following week at Boeing.

5. MY CAS PHD STUDENTS

A large part of the success of CAS can be attributed to the CAS PhD students. Over the past 25 years I met some truly amazing students at CAS and CASCON. The CAS experience provides students with an ideal foundation to become faculty members who can collaborate with industry effectively.

Scott Tilley and Kenny Wong (cf. Figures 2, 8 and 10) were my first CAS students working on program understanding projects, domain retargetable reverse engineering [8, 13], programmable reverse engineering [9], re-documentation [10], and the reverse engineering notebook [17]. Kenny also wrote the famous Rigi manual [12]. Scott and Kenny are now professors at Florida Institute of Technology and University of Alberta, respectively. Peggy Storey was technically not a CAS student, but worked for my CAS projects even as an undergraduate student. She not only developed tools, but also studied how tools affect programmer [15, 16]. She is now a professor and Canada Research Chair at University of Victoria.

Figure 10. John Schwarz, IBM Lab Director, awarding CAS Fellowships to Scott Tilley CAS students (CASCON 1993)

In the late nineties we shifted our attention from program understanding and reverse engineering to software reengineering and migration including Y2K migration. Dennis Smith, Scott Tilley and Grace Lewis, Carnegie Mellon Software Engineering Institute (SEI), who were working on Y2K, brought the SEI reengineering team to CASCON and added many new challenging dimensions to our projects. Converting the Toronto compiler back-end from PL/IX to C++ was real challenge and ultimately a success [20] (cf. Figure 11). In his thesis Johannes Martin distilled our programming language migration research (i.e., PL/IX to C++ and C to Java) including his Ephedra language migration platform [19]. Johannes is now a software engineer with ITC Consult GmbH in Germany.

Holger Kienle, a gifted writer, was highly effective in documenting and summarizing our tool building experience [23, 24, 25]. Holger is now a freelance computer scientist in Berlin. By 2005 our CAS projects were in the areas of service oriented architecture and autonomic computing. Piotr Kaminski was on a great trajectory for his PhD [20, 21], when Google enticed him to the excitement of Silicon Valley.

Ron Desmarais and Przemek Lach implemented the Yakkit platform to realize context-aware applications [29]. Ron integrated autonomic and cloud computing in developing adaptive control solutions for resource provisioning [37]. Ron and Przemek now work for Xperiel, a start-up company in Sunnyvale, California.

Figure 11. PL/IX to C++ Migration Project (CASCON 1998)

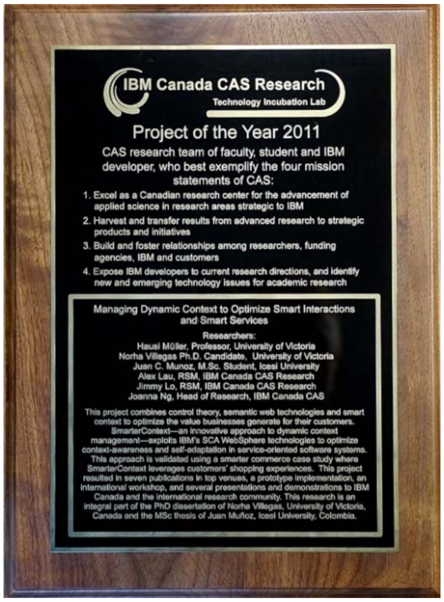
By 2008 my research group concentrated on engineering self-adaptive software systems. Our society is increasingly demanding situation-aware smarter software systems. Norha Villegas was one of my most productive CAS PhD students ever (cf. Norha Villegas paper in this volume). She investigated context management technologies—the SmarterContext ontology and the Personal Context Sphere (PCS)—and self-adaptive strategies for smart, situation-aware systems including the highly cited DYNAMICO reference model [26, 28, 30, 31, 32, 33, 34, 35, 36]. Because of her research we won the 2011 IBM Canada CAS Research Project of the Year Award [28] (cf. Figures 12 and 13) and the 2012 CASCON Best Paper Award [30]. Her dissertation was nominated for an ACM Distinguished Dissertation Award [32]. Norha is now a professor at Universidad Icesi, Colombia.

Figure 12. CASCON 2012 — Elena Voyloshnikova, Lorena Castañeda, Gabriel Tamura, Norha Villegas, Hausi Müller, Sudhakar Ganti, Przemek Lach, Ron Desmarais, Pratik Jain, Juan Muñoz

Lorena Castañeda is my most recent CAS student. She investigates models at runtime (MART) and self-adaptive web tasking strategies to support elderly [27, 28]. A MART is a causally connected self-representation of the associated system that emphasizes its structure, behaviour, or goals from a problem space perspective. Cyber physical systems require rethinking the software life cycle for which the distinction between development and execution time stages is no longer apparent.

Many other graduate students and research associates in my research group were directly involved and have contributed extensively to our IBM CAS projects, including Ernest Aaron, Priyanka Agrawal, Sowmya Balasubramanian, Andreas Bergen, Scott Brousseau, Brian Corrie, Marcus Csaky, Sahar Ebrahimi, Stephan Heinemann, Ishita Jain, Pratik Jain, Priyanka Gupta, Przemek Lach, Jun Ma, Jim McDaniel, Juan Muñoz, Sangeeta Neti, Mehmet Orgun, Atousa Pahleven, Jochen Stier, Nina Taheri, Marijana Tomic, Jim Uhl, Anke Weber, Mike Whitney, Qian Yang, and Qin Zhu. CSER

While I was on sabbatical at CAS in 1992/93 there were many calls from industry (IBM Toronto Lab, Bell Canada, Sun Microsystems, SMEs and others), government (NRC), and universities to strengthen software engineering education in Canada. There were few software engineering courses taught at universities.



IBM Canada CAS Research
Technology Incubation Lab

Project of the Year 2011

CAS research team of faculty, student and IBM developer, who best exemplify the four mission statements of CAS:

1. Excel as a Canadian research center for the advancement of applied science in research areas strategic to IBM
2. Harvest and transfer results from advanced research to strategic products and initiatives
3. Build and foster relationships among researchers, funding agencies, IBM and customers
4. Expose IBM developers to current research directions, and identify new and emerging technology issues for academic research

Managing Dynamic Context to Optimize Smart Interactions and Smart Services

Researchers:
Hausi Müller, Professor, University of Victoria
Norha Villegas Ph.D. Candidate, University of Victoria
Juan C. Munoz, M.Sc. Student, Icesi University
Alex Lau, RSM, IBM Canada CAS Research
Jimmy Lo, RSM, IBM Canada CAS Research
Joanna Ng, Head of Research, IBM Canada CAS

This project combines control theory, semantic web technologies and smart context to optimize the value businesses generate for their customers. SmarterContext—an innovative approach to dynamic context management—exploits IBM's SCA WebSphere technologies to optimize context-awareness and self-adaptation in service-oriented software systems. This approach is validated using a smarter commerce case study where SmarterContext leverages customers' shopping experiences. This project resulted in seven publications in top venues, a prototype implementation, an international workshop, and several presentations and demonstrations to IBM Canada and the international research community. This research is an integral part of the PhD dissertation of Norha Villegas, University of Victoria, Canada and the MSc thesis of Juan Muñoz, Icesi University, Colombia.

Figure 13. IBM Canada CAS Research Project of the Year 2011

To generate ideas and enlist support, Jacob Slonim and I orchestrated a *National Workshop on Software Engineering Education* in May 1993 with an impressive response from the Canadian computer science and software engineering community [3]. Key recommendations included establishing continuing software engineering education Masters programs and building a national software engineering consortium. Ontario universities created ConGESE (Consortium for Graduate Education in Software Engineering; in Alberta and British Columbia Paul Sorenson spearheaded WestMOST (Western Canada Masters of Software Technology).

Over the next three years Jacob Slonim, Morven Gentleman, Ric Holt, John Mylopoulos, François Coallier, and others worked hard extracting commitments from government (NSERC and NRC), companies, and universities to create the Consortium for Software Engineering Research (CSER). By 1996 CSER was up and running and duly incorporated as a not-for-profit organization. Over the past 19 years the biannual CSER meetings attracted 60-90 researchers and practitioners. For many years Anatol Kark, National Research Council, was deeply involved in CASCON and CSER. He was CSER Research Director for over a decade and program co-chair of five CASCONs (cf. Figures 14 and 15).

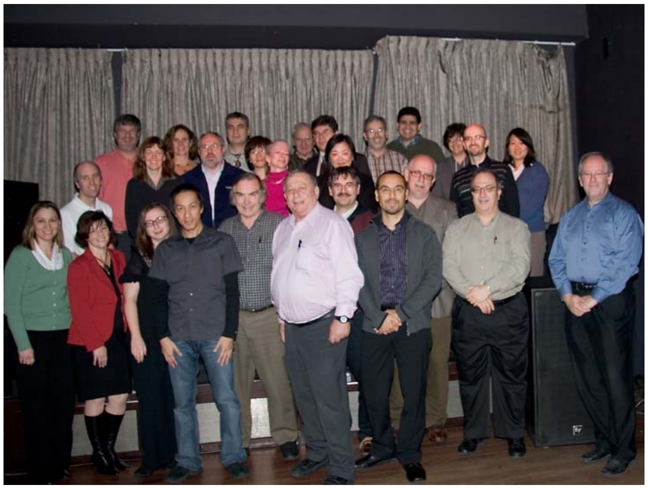


Figure 14. CASCON 2009 — Grace Lewis, Michael McAllister, Peggy Storey, Mike Godfrey, Wendy Powley, Eleni Stroulia, Kelly Lyons, Kenny Wong, Anatol Kark, Marin Litoiu, Morven Gentleman, Emilia Tung, Chess Kark, Jacob Slonim, Dennis Smith, Hausi Müller, Joanna Ng, Kostas Kontogiannis, Jim Cordy, Hakan Erdogmus, Ahmed Hassan, Brian Down, Hanan Erdogmus, Alexei Lapouchnian, Stephen Perelgut, Jenny Zou, and Pat Martin

In 2001 CSER won one of the prestigious NSERC Leo Derikx Synergy Awards. Joe Wigglesworth (CAS Director) and I had the privilege to accept this award at the Westin Harbour Castle in downtown Toronto.

Figure 15. CSER 2009 Anatol Kark retires from National Research Council (NRC), Ottawa

CSER is no longer incorporated but rather a loose organization providing a forum for Canadian software engineering researchers. Many Canadian students and faculty members have grown up with CSER—seeing their confidence and expertise grow while transitioning from Masters student to PhD student and eventually to faculty member is a real privilege. Unlike many other organizations, CSER has managed to recruit and attract new generations of researchers. This is a great tribute to the Canadian software engineering community.

6. CONCLUSIONS

CAS is both an IBM and a Canadian success story. In fact, it was so successful that IBM replicated it around the world (cf. article by Gabby Silberman and Cheryl Morris). Other companies also tried to emulate the collaborative CAS model. There are many people in the CAS ecosystem who worked really hard to sustain its impact and synergy over 25 years. I very much benefited from their vision and determination to make it a great success.

My sabbatical at CAS in 1992/93 influenced my research career profoundly. The collaborative CAS projects and NSERC CRD grants added tremendous value to my research program and greatly benefited the training of my students. NSERC CRD grants are intended as mutually beneficial collaborations to give companies access to the unique resources available at universities. Equally important particularly for the training of students is access to real-world problems in industry.

Figure 16. Panel at 20th Anniversary of CASCON with CAS Directors 1991-2015 — Gabby Silberman, Joanna Ng, Jacob Slonim, Kelly Lyons and Joe Wigglesworth

I worked with all CAS Directors Jacob Slonim, Gabby Silberman, Joe Wigglesworth, Kelly Lyons, and Joanna Ng (cf. Figure 16). With every new director the CAS ecosystem and collaboration changed. They all had different personalities, perspectives, experiences, strengths, and visions [1, 2, 43, 44, 53, 46]. They all had amazing energy, vision and determination to make CAS a great success, which was absolutely crucial to sustain this impressive 25-year journey of CAS and CASCON.

I hope that the precious gems of CAS and CASCON will generate impact and synergy for the next 25 years. I wish the new Director, Marcellus Mindel, great success for his term at the helm of CAS. To sustain these wonderful success stories and collaboration engines, people have to exhibit leadership, determination, and staying power—it is very worthwhile and highly rewarding.

Finally, as we just celebrated Canadian Thanksgiving, I am deeply grateful for all the wonderful colleagues and friends I had the pleasure to meet through CAS and CASCON over the past 25 years.

ACKNOWLEDGMENTS

I am deeply indebted to Jacob Slonim, who convinced me to spend my first sabbatical at the IBM Toronto Lab with IBM CAS at 895 Don Mills Road, North York. I especially would like to thank Kelly Lyons for showing me the ropes during that year and being a wonderful friend ever since. I would like to thank all the CAS Directors I worked with over the past 25 years for their leadership and support. I also would like to thank the Directors of the IBM Canada Software Laboratory for their vision and support for CAS and CASCON over the years. I also would like to thank all the collaborators on all our CAS projects, CASCON papers, exhibits and workshops, and of course the sing-alongs (cf. Figure 9). Finally, I would like to thank all my students who travelled with me to Toronto every year to experience the wonderful environment of IBM CASCON. I also would like to recognize and thank all the CAS staff including Cheryl Morris, Debbie Kilbride, Brenda Chow, Emilia Tung, Jimmy Lo, and many others for orchestrating and running CAS and CASCON so well behind the scenes.

REFERENCES

[1] Slonim, J., Bauer, A., Larson, P.-A., Schwarz, J., Butler, C., Buss, E.B. and Sabbah, D. The Centre for Advanced Studies: A model for applied research and development. *IBM Systems Journal* 33(3):382-398 (Aug. 1994)

[2] Perelgut, S., Silberman, G., Lyons, K.A. and Bennett, K. Overview: The Centre for Advanced Studies. *IBM Systems Journal* 36(1):474-488 (Jan. 1997)

[3] Buss, E., De Mori, R., Gentleman, M., Henshaw, J. Johnson, H., Kontogiannis, K. Merlo, E. Müller, H., Mylopoulos, J., Paul, S., Prakash, A. Stanley, M., Tilley, S., Troster, J. and Wong, K. Investigating reverse engineering technologies for the CAS program understanding project. *IBM Systems Journal* 33(3):477-500 (Aug. 1994)

[4] Müller, H.A. and Slonim, J. (Eds.). Proceedings of *National Workshop on Software Engineering Education (NWSSE)*, Toronto, Canada, May 31, 1993, 204 pages. Printed by National Research Council of Canada. (May 1993)

[5] Müller, H.A., Tilley, S.R., Orgun, M.A., Corrie, B.D. and Madhavji, N.H. A reverse engineering environment based on spatial and visual interconnection models. *ACM SIGSOFT Software Engineering Notes* 17(5):88-98 (Nov. 1992)

[6] Müller, H.A., Orgun, M.A., Tilley, S.R. and Uhl, J.S. A reverse-engineering approach to subsystem structure identification. *Journal of Software Maintenance: Research and Practice* 5(4):181-204 (Jan. 1993)

[7] Müller, H.A., Tilley, S.R. and Wong, K. Understanding software systems using reverse engineering technology: Perspectives from the Rigi project. *Proceedings Centre for Advanced Studies Conference on Collaborative Research (CASCON 1993).* pp. 217-226 (Nov. 1993)

[8] Tilley, S.R., Müller, H.A., Whitney, M.J., and Wong, K. Domain-retargetable reverse engineering. *IEEE International Conference on Software Maintenance (ICSM 1993).* pp. 142-151 (Sep. 1993)

[9] Tilley, S.R., Wong, K., Storey, M.-A. and Müller, H.A. Programmable reverse engineering. *Int. Journal of Software Engineering and Knowledge Eng.* 4(4):501-520 (Mar. 1994)

[10] Wong, K., Tilley, S.R., Müller, H.A., and Storey, M.-A. Structural redocumentation: A case study. *IEEE Software* 12(1):46-54 (Jan. 1995)

[11] Whitney, M., Kontogiannis, K., Johnson, H., Bernstein, B., De Mori, R., Corrie, B., Merlo, E., McDaniel, J., Müller, H.A., Mylopoulos, J., Stanley, M., Tilley, S.R. and Wong, K. Using an integrated toolset for program understanding. *Proceedings Centre for Advanced Studies Conference on*

Collaborative Research (CASCON 1995). pp. 19-31 (Nov. 1995)

[12] Wong, K. Rigi User's Manual Vers. 5.4.4 (Jun. 1998) http://www.rigi.cs.uvic.ca/downloads/rigi/doc/rigi-5.4.4-manual.pdf

[13] Tilley, S.R. *Domain-Retargetable Reverse Engineering.* PhD Thesis, Department of Computer Science, University of Victoria (Jan. 1995)

[14] Finnigan, P., Holt, R., Kalas, I., Kerr, S., Kontogiannis, K., Müller, H.A., Mylopoulos, J., Perelgut, S., Stanley, M. and Wong, K. The software bookshelf. *IBM Systems Journal* 36(4):564-593 (Nov. 1997)

[15] Storey, M.-A., Wong., K. and Müller, H.A. How do program understanding tools affect how programmers understand programs? *Proceedings Forth Working Conference on Reverse Engineering (WCRE 1997).* pp. 12-21 (Nov. 1997)

[16] Storey, M.-A. *A Cognitive Framework for Describing and Evaluating Software Tools.* School of Computing Science, Simon Fraser University (Dec. 1998) http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.2367&rep=rep1&type=pdf

[17] Wong, K. *The Reverse Engineering Notebook.* PhD Thesis, Department of Computer Science, University of Victoria (Apr. 1999) http://dspace.library.uvic.ca/bitstream/handle/1828/278/WongPhD.pdf

[18] Müller, H.A., Jahnke, J.H., Tilley, S.R. Smith, D.N., Storey, M.-A. and Wong, K. Reverse engineering: A roadmap. *Proceedings Conference on the Future of Software Engineering.* ACM, pp. 47-60 (May 2000)

[19] Martin, J. *Ephedra: A C to Java Migration Environment.* PhD Thesis, Department of Computer Science, University of Victoria (Jun. 2002) http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.7.3422&rep=rep1&type=pdf

[20] Kontogiannis, K., Martin, J.; Wong, K., Gregory, R., Müller, H.A. and Mylopoulos, J. Code migration through transformations: An experience report. *CASCON First Decade High Impact Papers—CASCON 20th Anniversary Volume*. H.A. Müller, A. Ryman, and A.W. Kark (eds.), pp. 201-213 (Nov. 2010). Reprinted from CASCON 1998.

[21] Kaminski, P., Müller, H.A. and Litoiu, M. A design for adaptive web service evolution, *Proceedings 2nd ACM/IEEE Int. Workshop on Software Eng. for Adaptive and Self-Managing Systems (SEAMS 2006),* pp. 86-92, (May 2006)

[22] Kaminski, P., Müller, H.A. and Litoiu, M. A design for evolving web services, *Proceedings 16th Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2006),* pp. 303-317 (Oct. 2006)

[23] Kienle, H.M. *Building Reverse Engineering Tools with Software Components.* PhD Thesis, Department of Computer Science, University of Victoria (Nov. 2006) http://hdl.handle.net/1828/115

[24] Kienle, H.M. and Müller, H.A. Rigi—An environment for software reverse engineering, exploration, visualization, and redocumentation. *Science of Computer Programming* 75(4):247-263 (Apr. 2010)

[25] Kienle, H.M. and Müller, H.A. The tools perspective on software reengineering: Requirements, construction, and evaluation. *Advances in Computers* 79:189-290, Elsevier (Dec. 2010)

[26] Villegas, N.M. and Müller, H.A. 2010. Managing dynamic context to optimize smart interactions and services. *The Smart Internet,* pp. 289-318, Springer (Jan. 2010)

[27] Müller, H.A.; A. Ryman; A.W. Kark (eds.): CASCON First Decade High Impact Papers—CASCON 20th Anniversary Volume. 274 pages, ACM (Nov. 2010)

[28] Villegas, N.M., Müller, H.A., Munoz, J.C., Lau, A., Ng J. and Brealey, C. A dynamic context management infrastructure for supporting user-driven web integration in the personal web, *Proceedings 21st Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2011),* pp. 200-214, ACM (Nov. 2011)

[29] Desmarais, R., Müller, H.A., and Lach, P. YaKit: A locality based messaging system using iCon overlay, *Proceedings 21st Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2011),* pp. 148-159, ACM (Nov. 2011)

[30] Ebrahimi, S., Villegas, N.M., Müller, H.A. and Thomo, A. SmarterDeals: A Context-aware Deal Recommendation System based on the SmarterContext Engine, *Proceedings 22nd Conf. of the Center for Adv. Studies on Collaborative Research (CASCON 2012),* pp. 113-127, ACM (Nov. 2012)

[31] Muñoz, J.C., Tamura, G., Villegas, N.M. and Müller, H.A. Surprise: User-controlled Granular Privacy and Security for Personal Data in SmarterContext, *Proceedings 22nd Conf. of the Center for Advanced Studies on Collaborative Research (CASCON 2012),* pp. 128-142, ACM (Nov. 2012)

[32] Villegas, N.M. *Context Management and Self-Adaptivity for Situation-aware Smart Software Systems.* PhD Thesis, Department of Computer Science, University of Victoria (Feb. 2013) http://hdl.handle.net/1828/4476

[33] Villegas, N.M., Tamura, G., Müller, H.A., Duchien, L. and Casallas, R. DYNAMICO: A reference model for governing control objectives and context relevance in self-adaptive systems. *Software Engineering for Self-Adaptive Systems II,* Springer, pp. 265-293 (Jan. 2013)

[34] Tamura, G., Villegas, N.M., Müller, H.A., Duchien, L. and Seinturier, L. Improving context-awareness using the DYNAMICO reference model. *Proceedings 8th ACM/IEEE International Symposium on Software Engineering for Self-Adaptive and Self-Managing Systems (SEAMS 2013),* pp. 153-162 (May 2013)

[35] Villegas, N.M. and Müller, H.A. The SmarterContext ontology and its application to the Smart Internet: A Smarter Commerce case study. *The Personal Web: A Research Agenda.* M. Chignell; J. Cordy, R. Kealey, J. Ng and Y. Yesha (Eds.). Springer, pp. 151-184 (Jul. 2013)

[36] Müller, H.M. and Villegas, N.M. Runtime evolution of highly dynamic software, *Evolving Software Systems.* T. Mens, A. Serebrenik and A. Cleve (Eds.), Springer, pp. 229-264 (Jan. 2014)

[37] Desmarais, R. *Adaptive Solutions to Resource Provisioning and Task Allocation Problems for Cloud Computing.* PhD Thesis, Department of Computer Science, University of Victoria (Dec. 2013) http://hdl.handle.net/1828/5111

[38] Bergen, A., Taherimakhsousi, N., Jain, P., Castañeda, L. and Müller, H.A. Dynamic context extraction in personal

communication applications. *Proceedings 23rd Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2013),* pp. 261-273, ACM (Nov. 2013)

[39] Castañeda, L., Villegas, N.M. and Müller, H.A. Self-adaptive applications: On the development of personalized web-tasking systems, *Proceedings 9th ACM/IEEE International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2014*), pp. 49-54 (Jun. 2014)

[40] Castañeda, L., Villegas, N.M. and Müller, H.A. Exploiting social context in personalized web-tasking applications, *Proceedings 24th Annual International Conference on Computer Science and Software Engineering (CASCON 2014),* pp. 134-147, ACM (Nov. 2014)

[41] CASCON Archives https://www-927.ibm.com/ibm/cas/cascon/archives.jsp

[42] CASCON Archives 1998-2007 https://www-927.ibm.com/ibm/cas/cascon/archives/cascon2007-1998.pdf

[43] Wigglesworth, J. Bringing academic research directly to development: IBM's Centers for Advanced Studies. *Proceedings IEEE International Engineering Management Conference (IEMC 2002).* Vol. 2, pp. 886-870 (2002)

[44] Hayat, Z. and Lyons, K.A. The evolution of the CASCON community: A social network analysis. *Proceedings 20th Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2010),* pp. 128-142, ACM Press (Nov. 2010)

[45] Ng, J. 2013. Extending CAS for Today's Relevance. IBM CAS White Paper. (Jun. 2009)

[46] Ng, J. 2015. The Four Languages of Innovation. IBM CAS White Paper. (Apr. 2010)

IBM’s Centre for Advanced Studies: A Personal Retrospective

John Mylopoulos
University of Trento
jm@cs.toronto.edu

I have benefited from IBM's Centre for Advanced Studies (hereafter CAS) since it was founded in 1991. In that year I was organizing an international workshop that was to be held near Toronto on “Intelligent Information Systems”, and was looking for some funding to cover local expenses. It just took two minutes to convince Jacob Slonim, first CAS director, that the project had merit and I got the funding on the spot. For a career academic used to application forms, delays and bureaucracy, this was a breath of fresh air. In fact, Jacob participated in the workshop and made valuable contributions to the discussion, encouraging participants to work on some of the many open problems faced by the Software Industry.

In 1993, I became a member of CAS and a participant in a CAS-supported project on software reverse engineering. The project included Hausi Mueller (University of Victoria) and Renato De Mori (McGill University). The project’s objective was to develop tools that reverse engineering legacy code, working with a team from the IBM Toronto Lab. It was an excellent learning experience for me and all participants, especially our students.

This project was followed by four others during the period 1996-2009, all funded jointly by CAS and NSERC. In addition to working with Hausi, these projects gave me the opportunity to work with Ric Holt (first at the University of Toronto, then Waterloo), Jim Cordy (Queen’s University), Renee Miller (University of Toronto) and others. In all cases we were profitably collaborating with different IBM Toronto Lab developer groups mostly on reengineering tasks. During these projects, I had the opportunity to work with more than a dozen promising graduate students who, upon graduation, went on to become respected academic colleagues. Notably, Kostas Kontogiannis joined the faculty at the University of Waterloo, Igor Jurisica joined Ontario Cancer Institute and the University of Toronto, Peggy Storey the University of Victoria, Kenny Wong the University of Alberta, and Ladan Tahvildari the University of Waterloo. I also saw more than 10 of my own students (mostly MSc) join the Toronto Lab where they have been enjoying an excellent working environment.

Through this long-standing involvement in CAS activities, I have come to recognize and appreciate the unique model of university-industry partnership that CAS established, ac- claimed world-wide as role model for R&D projects. Over the years, I have been in contact with industrial research programmes in Canada, the US, the EU, UK, Italy, Germany, Greece, the Netherlands, Switzerland, Japan, Hong Kong and more. Nowhere have I found a research programme such as the one operated by CAS. This programme is lightweight and flexible in that it requires minimal overhead and delays in applying for and reporting on a CAS project. It is inclusive, giving research opportunities to young faculty and students from around the world, rather than only the very best of the crop. It is innovation-oriented in that problems are defined directly straight out of industrial practice, and in many cases solutions are immediately adopted on-site. Last, but-not-least, the programme is grass roots-oriented, tailored to the interests of individual researchers and developers, rather than some preconceived master plan of topics, priorities and research directions.

I should add that CAS has benefitted tremendously from a long string of insightful directors and staff responsible for match-making researchers and developer groups and supervising CAS projects, notably Jacob Slonim, Gabby Silberman, Marin Litoiu, and Kelly Lyons.

CAS has operated since 1990 the CASCON conference. This conference draws hundreds of researchers and practitioners from around the world with a programme featuring keynote talks, a large assortment of demos put together by researchers and developers, workshops, and technical papers on topics of relevance to the mandate of the IBM Toronto Lab. The conference is very effective in creating bridges be- tween researchers and practitioners in a highly interactive and stimulating environment.

But the benefits of CAS to the community go beyond its projects and the CASCON conference. About 10 years ago, I served as general chair of the 30th Very Large Databases (VLDB) conference, held in Toronto. This is the oldest and most prestigious international conference on data management. CAS provided the organizing committee tremendous support in a number of areas ranging from organizational skills, sponsorship funding, advertising, graphic design and event planning. The conference was a resounding success boasting the second highest attendance ever for a VLDB conference (more than 750 people attending), a strong technical programme and high quality co-located events. For many participants, this was one of the best VLDB conferences ever. I am sure that we could not have accomplished this without constant encouragement and support from CAS.

Overall, CAS is sponsoring a unique industry-based research programme that other companies have tried to adopt. It has had great impact on IT-related research in Canada and the world, and is most deserving of recognition, and continuing support.

Conducting Research with IBM CAS
A Researcher's Perspective

Kostas Kontogiannis
Department of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
kkontog@softlab.ntua.gr

ABSTRACT

The Center for Advanced Studies at the IBM Toronto Laboratory and the IBM CASCON conference are celebrating 25 years of continuous support and collaboration with the University Community in order to conduct leading edge research in Computer Science and Computing applications. This collaboration has already benefited two generations of researchers in Canada, United States, and internationally. In addition to collaborating and supporting Doctoral students through prestigious IBM CAS Fellowships, CAS has provided the opportunity for Faculty members to have access to novel and innovative IBM products and infrastructure, and to join forces with outstanding IBM engineers and computer scientists both at the Toronto Lab, and worldwide. This short paper is an account of my experiences throughout a wonderful journey collaborating with CAS as a Doctoral student, and later on as Faculty.

1. A PERSONAL ACCOUNT COLLABORATING WITH CAS

My journey has started pretty much the same way as with many other fellow Doctoral students in early 1993, after my supervisor at McGill Prof. Renato DeMori and my co-supervisor Ettore Merlo had suggested to me the possibility of embarking on a CAS project in the area of Program Understanding and Reverse Engineering. The CAS Head at that time was Jacob Slonim. The process and the requirements of being awarded a CAS Doctoral Fellowship were straightforward. I should have completed my comprehensive examination, be active in my research, and pass a technical interview with senior IBM technical staff members. In my case my interviewers were Erich Buss, John Henshaw, and John Botsford.

And in this way, a wonderful journey of exploration, research, friendships, and professional growth has begun. In this journey, I had the opportunity to collaborate with many outstanding IBMers, from whom I have learnt a lot.

As a Doctoral student, I had to spend the summer terms away from McGill, working on my research at CAS premises in Toronto. These summer terms were not only productive research-wise, but also provided the opportunity to meet and build long lasting professional relationships with many other students, who are now Professors in renowned Universities, or successful professionals and entrepreneurs. But all good things come to an end, and in 1996 I got awarded my Ph.D. In 1997, after spending a year as a research associate at the Department of Computer Science, University of Toronto, I joined the Department of Electrical and Computer Engineering at the University of Waterloo as a faculty member, where I stayed for more than a decade.

As a faculty member at the University of Waterloo and later at National Technical University of Athens (NTUA), Greece, I had the opportunity to supervise an outstanding group of Doctoral and Masters students, most of whom have conducted their research within the framework of research projects funded by IBM CAS Fellowships. I am not only proud of these students and their career path, but also very appreciative to IBM CAS for providing the necessary infrastructure for these students to be exposed to challenging projects, innovative technologies, and a professional environment in which to work.

Looking back, I consider that the most important impact CAS has in the Canadian Computer Science and Engineering Community, is its ability to nurture young scientists and engineers and provide to them a healthy, fertile, and productive research environment that above all aims to build a collaborative research culture.

Figure 1. Eleni Stroulia and Kostas Kontogiannis at CASCON Technology Showcase

2. A CAS PROJECT PATH

A CAS project starts with a highly motivated student and a project application that has been approved by IBM. In my opinion, a key to a successful CAS project is the availability and engagement of an IBM team to act as a receptor of the technology and innovation to be developed. Students have the opportunity to work with highly skilled senior IBM technical staff, and are expected to spend their summers at the IBM Toronto Lab. In my case, I had the honor, privilege, and opportunity to work with some outstanding people like Joe Wigglesworth, Bill O'Farrell, Terry Lau, Tack Tong, Joanna Ng, and Chris Brealey. A CAS project has a planned duration of three years, and this is often ample time for a Doctoral student who has passed his or her comprehensive examination, and completed his or her coursework, to successfully finish their Ph.D. degree.

Figure 2. Joanna Ng and Kostas Kontogiannis at CASCON 20th Anniversary Panel

Throughout the course of a typical CAS project it is expected that the team has regular bi-weekly or monthly meetings with the receptor IBM team. This provides the team the appropriate traction to conduct the project in a way that is both innovative and of practical importance to the computing industry in general. It is also common for the CAS funds be matched with funds from Federal, Provincial or University sources. This is a very important aspect as it provides the opportunity for several related CAS projects to identify more general research problems and apply for Collaborative Research and Development projects, Strategic grants, or Innovation Funds. My experience is that CAS projects can utilize the opportunities given by agencies such as NSERC, ORF, CFI to leverage CAS funds to spawn new innovative projects in collaboration with other Universities, IBM, and possibly other corporations or government agencies. This model has been applied very successfully in the Canadian research community and has fostered strong research bonds among groups across the country.

3. IBM CAS AND ITS ROLE ON HIGHLY QUALIFIED PERSONNEL

The Center for Advanced Studies and CASCON in particular brings together researchers not only from all over Canada but also, from many other countries. This alone creates a positive spin for researchers and practitioners to exchange ideas and opinions on innovative products and solutions, and in this way incubate the next generation of CAS projects. However, this interaction is only a small fraction of what CAS and CASCON is all about.

First, IBM CAS and its conference CASCON provide the students with the unique opportunity to work and interact with scientists and engineers of one of the most acclaimed computing companies in the world. However, in my opinion, the biggest advantage from this interaction is not the exchange of technical ideas between the students and IBMers, but the exposure of students to an environment of high work ethics, professional attitude, punctuality, and responsibility. These are virtues that go beyond technical abilities, and follow a person throughout his or her life.

Second, IBM CAS and CASCON over the past 25 years have contributed the most to providing a fertile ground for young scientists and engineers to pursue their Doctoral or Masters studies. Currently there are two generations of researchers trained at CAS and CASCON, and we are in the middle of a third generation. It is needless to say that all CAS affiliated graduates have excelled in their professional lives, and have already made an impact in the research community globally. To this extent, the Canadian computing community has benefited greatly from IBM CAS, a fact that is recognized by numerous awards by the Federal and Provincial governments. My Doctoral students (Ying Zou, Ladan Tahvildari, and Kamran Sartipi) who were affiliated with CAS are now prominent Faculty members in top Canadian Universities, while others (Ali Razavi) make an impact in the computing industry working as senior researchers and engineers.

Finally, I would like to amplify the fact that IBM CAS and CASCON are always open to the community, and seek new innovative ways to attract new collaborations and train new talent. This provides the necessary infrastructure to foster new communities, the means and the environment for researchers to participate in new initiatives such as Centers of Excellence as well as, Federal and Provincial Research Centers, and help propel the Canadian research community to be among the best worldwide.

Figure 3. Enjoying the CASCON Paper Sessions with Tim Lethbridge

Acknowledgment

I would like first and foremost to thank all my graduate students who have given me a wonderful time working with them, the University of Waterloo and NTUA for giving me an outstanding environment to conduct research, and IBM CAS for supporting our work throughout the years.

Academia, Big Business, Lunch and other CAS Reflections

Chris Brealey
IBM Senior Technical Staff Member and professional nerd
DevOps, Cloud, Mobile, REST, SOA, Web services, etc.
IBM Canada Lab, Markham, Ontario, Canada
cbrealey@ca.ibm.com

1. INTRODUCTION

There's a nasty rumor about town that the prevailing allure of the annual fall conference of the Centres for Advanced Studies, or CASCON, thus reason to attend said conference, is lunch. I can neither confirm nor deny this rumor. For myself, I can only state that

1. I have attended CASCON religiously for at least 12 years or roughly half of CAS's esteemed 25-year life for the opportunity to listen to, present to, learn from, and otherwise interact with colleagues from academia and from IBM, and
2. lunch at CASCON is tasty.

The balance of this chapter constitutes as much a personal trip down CAS memory lane as anything else. It bears none of the quantitative earmarks of a typical CAS paper. This is neither a scientific nor, in all likelihood, particularly un-biased write-up. The only guarantee I make is that I will not recount what I ate for lunch.

2. A BREAK FROM ACADEMIA

Fourth year at the University of Victoria held for me some of the most fun and engaging classes of my then academic existence—software engineering, graph theory, artificial intelligence, and computer graphics to name a few. That last class was taught by a relatively new face in the classrooms of UVic—the ever patient and personable Dr. Hausi Müller.

Little did I know at the time that my graduation from the University of Victoria in 1989 would not mark the end of my acquaintance with academia in general, or with Dr. Müller in particular.

Notwithstanding a couple of co-operative education work terms, life as a student of computer science gave way to life as a software developer at IBM Canada Ltd. at the end of July, 1989. Fast forward about 10 years to two events that would nudge me in the direction of CAS. First, I began working on a project to create tools for developers to use to build Web services based on the then emerging SOAP and WSDL specifications.

Second, I met and began working with Dr. Arthur Ryman, chief architect on the aforementioned tools, and co-founder of the Toronto Centre for Advanced Studies. Between the rapidly growing interest in SOA and Web services amongst enterprises, and Dr. Ryman's affinity to CAS, it is not surprising to me that within the next couple of years our Web services tools development team and I began making appearances at CASCON.

3. TECHNOLOGY SHOWCASES

My involvement with CASCON those first few years took the shape of staffing a poster session, otherwise known as pedestal or simply "ped" at the conference's Technology Showcase.

I only had a couple of IBM business conferences under my belt at this point in my career, but it was enough to recognize that CASCON presented a very unique flavor to the intellectual palette compared to those conferences. Most folks that ambled up to our pedestal were not software project managers, or product architects, or CIOs, or analysts.

Sure, we got all kinds of visitors including a smattering in those roles. For the most part though, they tended to be university students, professors, and IBM developers, all keen to learn more (often between bites of lunch perched on plates balanced expertly on one hand, leaving the other hand free to gesticulate wildly) about what these new-fangled Web services things were, why the information technology industry was becoming so entranced by them, and what opportunities there might be to get involved and advance the art of Service Oriented Architecture (SOA).

Here was an audience that enjoyed the simple pleasure of chatting about the origins and theory of a programming model – of debating the technical and philosophical pros and cons of a protocol—of exploring the strengths and weaknesses of an approach—of igniting ideas and opportunities to advance Web services and industry's use of them to the next level. Incidentally, it was also here that I reconnected with my old (but not too old!) fourth year computer graphics professor, Dr. Müller.

After over a decade I was dipping my toes back into academic waters. I was hooked!

4. WORKSHOPS

October 20, 2005 marked my first involvement in and presentation at a workshop: *Web Service Interoperability Challenges*. In retrospect, I don't think I went into that session with a true understanding of the mutually beneficial relationship between IBM and academia that is CAS. Had it been a typical IBM customer facing conference, the session might have focused on what Web services are, their value to the enterprise, best practices for implementing and deploying them, and tools and run-times (from you know who!) to help.

But this wasn't a typical IBM conference. This was CASCON. The audience was different. The purpose was different. The conversational dynamic throughout the session was different. Not worse or better than the average business conference—just different. As shocking as this may seem, we IBMers were not at

CASCON to sell. We were there to jam with our academic partners – to learn – to share – to explore – to theorize – to innovate.

One indicator of these differences was in the word “challenges” in the workshop title. Web services had already come a long way since their inception in the late 90’s and offered numerous benefits to enterprises; however, they also had a few issues. So, in the workshop, presenters and audience members alike strolled through a few of the juicier gaps and complexities of Web services—from general interoperability woes resulting from subtly different interpretations of the specs, to how to deliver services with certain qualities of service like reliable delivery or security, to how to embed massive amounts of binary goo in a SOAP message and whether or not it’s even a good idea.

Thinking back on it, whatever babbling monologue I may have hurled in the general direction of the audience that day was most definitely not the exciting part. The exciting part was the discussion—the sensation of intriguing new lines of thought and novel ideas flowing from and amongst the participants.

As the years progressed I was invited back to attend, co-present, or co-chair workshops. I began to notice a couple things. I don’t think I noticed that I was noticing them at the time. In hindsight then:

1. The level of discussion around SOA and Web services was evolving. A mixture of old and new faces from one workshop to the next assured the presence of both continuity from past discussions and injection of fresh perspectives, ideas, research and results into the CAS / IBM consciousness.
2. My circle of friends, colleagues and acquaintances was growing! I was working with professors and students from universities around the planet—Carnegie Mellon University—the University of Toronto—The University of Athens—the University of Victoria (naturally!) I was also meeting & greeting colleagues from within IBM that I had previously only worked with remotely, if at all.

Interesting, if not downright valuable, effects resulted. Brainwaves generated in these workshops influenced, often in subtle ways, the evolution of open source projects and full-fledged products that we were working on at IBM. Observations and feedback about the look, feel, and function of IBM’s run-times and tools—say, in the Web services space, but certainly not limited to that—helped seed work items in the product backlogs. The workshops were neither planned nor conducted as end user feedback sessions; nevertheless, they served—at least subliminally—in that capacity.

In return, the predominantly academic participants in these sessions gained a measure of insight into the pragmatic, customer and revenue-oriented imperatives facing IBM’s development projects. This so-called “reality check” served not to limit or constrain any of the innovation or out-of-the-box thinking in the room. Far from it, I found that it helped focus and link research agendas to real problems with which real customers were grappling.

5. FELLOWSHIP PROJECTS

I’m not sure exactly how this happened—after all, CAS’s 25th year with IBM happens to be my 26th year with IBM as a regular employee, and my memory just is not what it used to be—but after four or five years of technology showcases and workshops I found myself slurped into something mystical called a “CAS fellowship project.”

Let’s see if I’m close… IBM and academia work together to identify areas of research of mutual benefit to both sectors. IBM secures funding for professors and students to execute a research agenda. The professors and students and their circles reap the benefits of published papers, advancement of theory, education, invention, theses, and degrees. IBM and its customers benefit from the injection into products, standards, and open source projects of new ideas, approaches, features and designs. Sounds like a win-win to me!

Returning to that first project, it was ignited from a research proposal that Dr. Kostas Kontogiannis of the National Technical University of Athens drafted following some chitchat and discussion the two of us had. This particular project might not have happened had I not met Dr. Kontogiannis at a prior CASCON workshop about the good, the bad, and the ugly of Web services. Our kindred interest in SOA and model driven software engineering primed not only that proposal and project, but also a long-standing collaboration on numerous workshops and research work to follow.

What was the project all about? Well, one of the technical challenges with developing Web services back in the day was this: Every time somebody tweaked a WSDL document or the XML Schema that it used, a complete regeneration and replacement of all language specific client and server-side artifacts corresponding to the WSDL and Schema was required. This collateral replacement would often ripple through the rest of the system, and the ripple effect was manual and fraught with error. The aim of the project was to devise a theory and practical mechanism for modeling the relationships between a Web service description and its related language artifacts, and for incrementally percolating changes to that Web service description to those artifacts with greater efficiency and less disruption.

The project culminated in the CASCON 2009 workshop *Incremental model synchronization in model driven development environments*—Kostas Kontogiannis, Ali Razavi, Leho Nigul, Chris Brealey.

Figure 1. 2013 CAS Research Project of the Year Tinny Ng, Diana Lau, Kostas Kontogiannis, Michael Athanasopoulos, Joanna Ng, Chris Brealey and Ahmed Hassan

The project was still cooling down when Dr. Kontogiannis and I once again huddled around a cauldron with student Michael

Athanasopoulos, tossed in a witch's brew of mutually chemically unstable ingredients including "Web services" and "REST", and cooked up a second proposal that would ultimately yield the 2013 CAS Research Project of the Year, a *Framework for the Deployment and Use of Legacy Enterprise Services Utilizing the REST Protocol*—Kostas Kontogiannis, Michael Athanasopoulos, Chris Brealey, Diana Lau, Joanna Ng, and Tinny Ng (cf. Figure 1).

If there was ever a project whose workshops inspired the kind of passionate debate central to the vibrancy of CAS, it was this one. After all, two polarizing technologies—Web services and REST—were in the mix. Ironically, the purpose of the project was not to throw the two approaches into the ring and have them go ten rounds until a winner was declared with the loser left broken and bloody on the floor. The purpose of the project was to devise a theory and mechanism that would help the two approaches co-exist, in recognition that the typical enterprise was overflowing with perfectly valid, procedure oriented services seeking re-expression as REST services for loose, simple integration with internet-friendly clients like mobile apps and web applications. To this day I break out into a wide grin when I see Michael's prototype Service Component Architecture REST binding come alive with zero human intervention in front of a pre-existing Web service. Sweet!

I truly cannot pick a favorite fellowship project, but that one might have been the most fun for the discussion it inspired.

6. FAMILY

Should we define CAS by its mission, or by how it is organized, or by what it does, or by its conference agenda, or by the fellowship projects it sponsors? Perhaps; but when I muse upon that question, I invariably find my mind wandering to the people that make CAS what it is – the professors, the students, and the fellow IBMers, all of whom I am in constant awe for their commitment to collaboration and research to help build a smarter planet.

My most humble thanks go out to the CAS team behind this book for inviting me, a very small cog in the CAS machine, to pen a few words.

See you at CASCON—quite possibly near the lunch buffet.

CAS Projects: Software Engineering

Growing Up with the IBM CAS at Toronto: A Personal Reflection

Nazim H. Madhavji
Department of Computer Science
University of Western Ontario
London, ON
N6A 5B7

madhavji@gmail.com

ABSTRACT

This paper describes a personal reflection of my collaborative relationship with the Centre for Advanced Studies (CAS) at Toronto and the IBM Toronto Lab. In particular, it describes: some of the collaborative projects and the transfer of the results to the Lab; impact of the collaboration on my research, student training, and pedagogy. Finally, I describe some important lessons learnt during the past twenty-five years of collaboration with CAS and the IBM Toronto Lab.

Categories and Subject Descriptors

A.m [Miscellaneous]

K.2 [History of Computing]

K.7.2 [The Computing Profession]: Organizations.

Keywords

Academia-Industry Research collaboration, Technology Transfer, Lessons learnt, IBM Canada, 25th Anniversary of CAS.

1. AN UNEXPECTED BLESSING

Ring ring; ring ring .....

“Hello?”

“Ehh ... Nazim?”

“Yes, who’s calling please?”

“Ehh ... this is Jacob; Jacob Slonim. I am now at IBM Toronto Lab ... remember me from GEAC?”

“Yes, of course! Hello Jacob?”

“Ehh ... well, I am in town [Montreal] and_ehh ... would like to meet you this afternoon”

Jacob and I met that afternoon and through the evening, where he tells me, “I would like you to come to the Lab and talk to the product people. I am setting up a Centre for Advanced Studies and would like you and your students to work with us, visit us, spend summers with us ....”.

As a relatively young researcher at the time, I had never heard of such a thing, where an organisation was reaching out to academics, and what has since turned out to be a close collaboration over many years. This was indeed God sent; an unexpected blessing.

As a faculty member, this sort of connection solves, to throw in a number, 70% of our problems – finding like-minded people in industry who would like to collaborate with university researchers to create a win-win value. The remaining 30% may be a grind but without a dependable and tolerant industrial partner, practically-significant scientific progress is going to be difficult, especially for empiricists like me who look for “real” problems to investigate, “real” data to analyse, and ”real” environment to try out the resultant findings and solutions.

What started out with a simple call to an academic used to sitting behind the desk and experimenting with conjectured ideas on incremental development of Modula-2 programs, led to a long journey over many years, in terms of:

- broadening my horizon and gaining knowledge about large-scale software problems;
- creating new knowledge and solutions through collaborative research projects;
- building tools and integrate them into production;
- graduating numerous students who are making a difference today;
- writing numerous scientific publications and obtaining a patent;
- fostering relationships with IBMers;
- raising research funds used for training highly qualified personnel (HQP);

Unquestionably, IBM Canada through its Centre for Advanced Studies (CAS) has been the most significant player in influencing my thoughts and shaping my career through the past 25 years. In fact, it is many-fold more remarkable to think that CAS has galvanised collaboration with numerous Canadian and US academics. This journey has not ended thankfully -- a continued blessing. As I write, new explorations are being discussed with IBM personnel, whereby the role of software engineering is being analysed in the hurricane of Big Data and Data Analytics.

Ours is a field where technological shifts are to be expected of course: waterfall processes, iterative processes, RUP, TQM, process improvement, metrics, UML, OO technologies, development environments, CASE, frameworks, e-commerce, services and SOA, Open source, agile methods, crowdsourcing, mobile devices and computing, social media, wearable devices, continuous engineering, DevOps, Cloud computing and architectures, Big Data, IOT, and more. IBM and CAS have been at the forefront of these and other topics. In turn, for those academics, like me, who have been intimately involved with CAS, these technological shifts at IBM have prodded us to keep abreast of such changes. It is quite plausible to think that without the catalytic changes at IBM, many of us would have become stagnant by now.

2. COLLABORATIVE PROJECTS AND TECHNOLOGY TRANSFER

CAS, in conjunction with the CASCON conference, has been instrumental in motivating and supporting my research work for many years. Early work (pre-2002) centred around tools for software process modelling and analysis, congruence between software process and its environment, evolutionary policy checking and analysis, and tool insertion. Tools such as X-elicit, V-elicit, Congruence Evaluation System (CES) and the TIM method are examples of this research.

The process tools, which were demoed at various CASCON conferences, were used to model and assess the quality of the Lab's requirements engineering processes and the AS/400 Plan and Dependency Management (PMDM) process. These processes underwent changes directly resultant from the quality assessment. Also, the TIM approach was validated by inserting the Requirements Management Tool (RMT) in a production project and was subsequently adopted for internal use by the Lab's process improvement team.

Later work examined a large, legacy system, specifically, architectural degeneration and multiple component defects (MCDs), fault localisation, and system testing. We analysed MCDs defects over several releases of a large legacy system (of size approximately 20 million source lines of code and age over 20 years). This investigation showed how MCDs can depict the system's *defect architecture* at any given point in time, and how data over time can help understand the *degeneration* of the system from defect perspective. This is particularly relevant because the study also determined that MCDs require three times more code changes to fix properly (compared to non-MCDs) and are more persistent across development phases (80% for MCDs vs. 12% for non-MCDs) and releases (13% for MCDs vs. 3% for non-MCDs). This work identified components of the legacy system that required more urgent re-engineering attention.

In the area of fault localisation, we developed a prototype tool to automatically locate defective software components by analysing program's execution traces. Between 70-99% of the time, the tool can locate defective components and prioritise them as top contenders, saving 30-50% of a maintainer's time in defect fixing. Our approach was tested on historical data and the results were in agreement with the practitioners' impressions of the system's defect quality.

In the area of system testing, we developed a novel technique to compare program-execution traces and identify similar traces (between test scenarios and field usages) where end-users were experiencing usage problems. Based on this, the tester or quality engineers can improve test cases and scenarios. A paper from this work was published at CASCON where it was ranked as the best student paper. The compression technique was used in the DB2 environment to identify improvements to be made to the test buckets, saving many hours of manual labour.

In parallel, a number of specific tools and techniques were also developed and transferred to the DB2 SVT and QA environment. For example, code coverage analysis using full db2trc was integrated into the DB2 SVT test cycle for v91 and was used to evaluate previously difficult to assess code coverage within DB2's large and complex code. Further interest from the DB2 SVT team led to the development of a source code static analysis tool that analysed DB2 source code to identify complex recursive loops involving chains of multiple function calls across several layers of the system architecture. This tool was used by DB2 LUW Continuing Engineering and RAS/PD development groups during system maintenance work to support system comprehension, analysis and change process. Also, we developed a tool for defect arrival trend analysis that was subsequently used by DB2 QA managers as an additional metric showing code quality. This work was used as an in-process tool to add to the analysis of defect trends for the Faster Redistribute solution of DB2 LUW v95.

3. IMPACT ON RESEARCH, STUDENT TRAINING, AND PEDAGOGY

It would be extremely naive to believe that we, researchers, are usually armed with novel solutions which we can readily be applied to industrial-scale problems! In fact, time and again, I have been astounded by the degree of practical insight product staff have shown on both technical and business matters that have not emerged from university corridors and labs. The faculty invariably learn, as do the research students, from interacting and collaborating with product groups. In turn, the knowledge gained adds to the "fodder for fire" in the creation of research solutions.

From experience, collaborating with CAS and the product groups at IBM over two decades has completely transformed my outlook as a researcher, for the better I believe. In turn, this has changed how I supervise my research students and what I teach to my course students in software engineering. My students are more confident about their research. They believe in the problems they are investigating. They see their solutions and ideas having a positive impact in production environments. On the international stage, they are articulate about their research and can defend their ideas with reason in the scientific community.

Directly or indirectly through matching funds, IBM has funded research of five doctoral and thirteen masters students of mine over the years. Some of them have joined IBM Canada and some, other organisations; some have become faculty members, in Canada and abroad, and have their own research students who have joined industry or academia. In other words, they are all productive members of the society worldwide. From this work, collectively, we have published over 20 papers in scientific journals and conferences. Such is the impact of CAS.

4. LESSONS LEARNT ON RESEARCH

Many lessons have been learnt about research over the years. Here, I describe some examples:

1) When writing a research proposal, involve the practitioners. <u>*Moral of this point*</u>: You need their vote of confidence in the proposal for them to be motivated in your research.

2) Research involving industry is best done in short cycles of: listen (and obtain feedback), plan and do-research, and show-and-tell or try out in the actual or mock-up environment.
   Moral of this point: By operating in short cycles, learning goes on continuously in both camps – researcher's and practitioners' – and there is shared understanding of the emerging solution or results of a study.

3) Not to lose site of the fact that the primary results of research are technical solutions and actionable findings of studies, not research publications.
   Moral of this point: While publications are clearly important in academia, practitioners want actionable results they can use in their projects or organisations. Remember, they are the prime stakeholders, for without them there would be no results in the first place!

4) Due to decentralised authority of permissions and ownership of projects' artefacts in a large corporation, it should be factored in the research plan that what may seem as "little things" like meeting the right product people or obtaining access to certain artefacts may take time, long time. In particular, signing a non-disclosure agreement with CAS does not guarantee immediate access to the production environment. Trust needs to be built up with the product groups, which cannot happen overnight. Also, this is as much an issue at the start time of a research project as during the conduct of the project.
   Moral of this point: Faculty and graduate students better beware of these risks as, perhaps, not much can be done to force change in such circumstances. Perhaps, better to highlight this issue up-front so as to establish realistic expectations on both sides.

5. LESSONS LEARNT ON TECHNOLOGY TRANSFER

Below, I describe some lessons learnt on technology transfer. It is in the form of a dialogue between a product staff and the researcher:

1) So, you did a demo of your research prototype, which may have been interesting. However, why did you not come to me before building your prototype to ask what my requirements were?
   Moral of this point: know the characteristics of the nails before building a hammer for them! In other words, know the requirements of your users before building a solution for them. In hind sight, this is clearly common sense but, even today, much research fails to show the validity and generality of the problem it claims to solve.

2) Am I the first guinea pig? Sorry, I would like to hear from my colleague(s), not from you, that your system is useful.
   Moral of this point: researchers need to understand that transfer of technology is a complicated process. One shouldn't expect that a prototype tool "hot off the oven" (out of the lab) will be embraced by a commercial project that is working under real-world constraints. There is an intermediate stage (or stages) whereby the prototype tool is put to trial use in low- or no-risk projects in industry to test the feasibility of the tool. No one in the right minds will think of flying a prototype plane as a commercial airplane flying real cargo or passengers! In the software world, we have yet to embrace rigour in technology and knowledge transfer from the labs to commercial projects.

3) You say your system is easy to use. However, do you have any learning curve data on how long it would take a practitioner to become proficient in using your system in production projects? I have over fifteen people in my project and so letting them use your system without knowing the learning curve is risky.
   Moral of this point: without trustworthy data on usability, it is akin to driving through fog without fog-lamps; you don't know where you are going. Researchers need to understand that while completion of a prototype solution is encouraging, it is far from satisfactory from the point of view of the end-users of the system in a production project.

4) Is this university software? Nop, sorry, what happens when your student (or you) leave the university? Who will maintain or enhance the system for us?
   Moral of this point: commercial projects might not be able to depend on prototypes. There is thus a need to handover the research prototypes to commercial developers or to spin-offs, that will guarantee maintenance and evolutionary support.

6. CLOSING STATEMENT

Perhaps the ultimate impact of CAS is summed up by the following: whenever I walk into the IBM building at 8200 Warden Ave., Markham, it feels like I have returned home. I suspect I am not the only one with such sentiments about CAS.

7. ACKNOWLEDGMENTS

Numerous people at CAS and IBM, in various roles, have contributed to our success over the years. We thank them all sincerely for their generous time, effort, expertise, patience, and financial support. Here, we list the key enablers and frequent collaborators in alphabetical order, and our sincere apologies for forgetting others: Robert Begg, John Botsford, Enzo Cialini, Michelle Gittens, Dave Godwin, Mark Hecht, Kelly Lyons, Andriy Miranskyy, Joanna Ng, Gabby Silberman, Jacob Slonim, Craig Statchuk, Colin Taylor, Joel Troster, Joe Wigglesworth, Mark Wilding, and Calisto Zuzarte. Thank you CAS! May you live for ever more!

Usable Software Tools: Winding Paths of Involvement in Cascon and CAS

Timothy C. Lethbridge
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa ON K1N 6N5 Canada
+1 613 562-5800 x6685
tcl@eecs.uottawa.ca

ABSTRACT

My entire career as a faculty member, and that of dozens of my undergraduate and graduate students, has been guided along particular paths due to Cascon, the IBM Centers for Advanced Studies (CAS), and the related Consortium on Software Engineering Research (CSER). In this paper I describe these interconnected paths of research and collaboration. The intertwined paths include research and development of highly qualified personnel in reverse engineering, usability, software tools, modeling, clustering, work practices and several other topics relating to usable software tools. The large group of professors, students and practitioners that has met each year at Cascon and the related CSER meeting have molded the research, which would undoubtedly have taken utterly different paths without Cascon, CAS and CSER involvement.

Keywords

Cascon, IBM, CAS, CSER, software engineering tools, usability, clustering, TkSee, Umple

1. INTRODUCTION

I started my involvement with Cascon in 1996 when IBM helped form CSER, the Consortium for Software Engineering Research. That connected me with a dynamic group of researchers from across Canada, as well as some from overseas. Interaction with many of these people has continued to this day.

Several of my students and I have also benefited from fellowships and a 2006 sabbatical period offered by the IBM Centers for Advanced Studies branch in Ottawa.

Over 100 of my published papers, 21 of my graduate students and about 40 undergraduates have directly or indirectly benefited from this interaction. Numerous other collegial relationships and friendships have also resulted.

In this paper I describe highlights of my experiences with Cascon and CAS.

2. THE BACK-STORY (1987-1995)

I will start with a quick look at some background that gave me the expertise in software engineering, modeling, reverse engineering and usability, all of which proved so useful for my involvement with CAS and Cascon.

After working on my Masters (1985-87 at UNB) in computer animation, I was hired at Bell-Northern Research (historically part of now-defunct Nortel) with the expectation I would work on animating CAD tools. But when I arrived at BNR I was asked to work on database and user-experience issues, topics that were later of particular interest at Cascon and CAS.

During my PhD research in usable Knowledge Management tools (1989-94 at the University of Ottawa), I taught user interface design and introductory software engineering to undergraduates; central to the latter was the modeling language OMT, one of the predecessors of UML. Experiencing difficulties students faced gave me great insights for future research.

I also spent time with what was called the Telos team at BNR. Telos was a real-time modeling tool, and I helped them use our knowledge management system (CODE4 [28]) to clarify design concepts. The Telos project was cancelled one tumultuous day, but several employees, including Bran Selic, span it off to form ObjecTime, later bought by Rational, later bought by IBM. Concepts I worked on helped find their way into UML, and my involvement with Bran Selic, who was for several years an IBM Distinguished Engineer and leader of the UML project, has continued to this day.

3. THE CSER AND TKSEE ERA (1996-2002)

In 1996 when I had been a replacement assistant professor for a little over a year, the University of Ottawa bought a new PBX from Mitel Corporation. Part of the deal was that Mitel would help fund UOttawa researchers. The key problem Mitel faced was dealing with a huge volume of legacy code. They were interested in managing knowledge about their code using CODE4.

Within months, IBM, Mitel and the National Research Council NRC) founded the Consortium for Software Engineering Research (CSER) in 1996 and my project became part of this.

CSER was, in its early days focused on the problem of legacy code, and reverse-engineering in particular. Through CSER, my research with Mitel was granted a large amount of matching NSERC funds, so expanded dramatically. I also started working with researchers at the NRC, notably Janice Singer, and we regularly reported our results at the annual Cascon conference as well as the twice-yearly CSER meetings, one of which was and still is held at Cascon. Numerous other conference papers also resulted (many discussed below).

No sooner had we started the research with Mitel and performed studies of their work practices, than we realized the Mitel PBX group's most pressing problem was lack of tool support for intelligently browsing and searching code. Although we did publish one paper on the original theme [1], we completely switched gears and focused on reverse engineering and usable code browsing tools (this was in the early days of the web and before Eclipse).

I hired two postdocs (Nicholas Anquetil and Stéphane Somé) and a host of graduate students. Our work rapidly spread to cover topics such as clustering [2] [12], analysis of code identifiers [2] [4], and work patterns [3] [7] [13] [15]. Paper [3] was awarded in 2010 a Cascon First Decade High Impact Paper award for one of the 10 best papers out of the 425 published in the first decade of Cascon.

Central to our work in this era was TkSee, a 'software exploration environment' that was put into real industrial use at Mitel. The TkSee tool was regularly demonstrated at the Cascon Technology Showcase as it developed, and we published various papers [8] [9], including one in Cascon [5].

A parallel research path of mine involved mining of data from the Mitel software, and applying machine learning to find elements of software that may be in need of maintenance [6] [10].

One of the most exciting periods of this era was the reverse engineering competitions undertaken at Cascon. Susan Sim, then at the University of Toronto, was instrumental. TkSee was pitted against tools from other researchers such as Rigi [29] to extract data from software.

How should that data be represented? What should be the schema? This was one of the key discussions I was involved at several Cascons. The XML Schema GXL (Graph Exchange language [30]) was promoted by many; Ric Holt's TA (Tuple Attribute [31]) format was another contender. In TkSee we adopted TA, but it was the debates themselves that probably had the most value, bringing everybody together to think, to build tools, and to collaborate.

Arising from the Cascon/CSER inspired research nexus, Canadian researchers focusing on reverse engineering connected with researchers in other countries, particularly Germany, and several Dagstuhl workshops were held [32]. One outcome was the Dagstuhl Middle Metamodel for reverse engineering [11], essentially a set of class diagrams describing the entities and relationships possible in reverse engineered software. To this day Canadian and German researchers collaborate extensively in areas of software engineering research with roots in that era.

One of the beauties of CSER meetings was, and still is, that everybody got to explain their work without peer review, so we can talk about half-baked ideas and have help 'baking' them through discussions. The polished research is then often published after rigorous peer review in Cascon.

Numerous graduate students who made their contacts and presented their research in CSER and CASCON would move on to become professors, including some of my own. Today we see third and fourth generation students at CSER and CASCON.

4. THE INTERMEDIATE ERA (2003-2005)

I was on Sabbatical in South Africa in 2002, deep in the mountains working on writing papers at a B&B, when I received a fateful call: Mitel was pulling out of all University research.

However, my research didn't stop with the end of Mitel funding; I was able to obtain CAS Ottawa fellowship funding from IBM and also funding from QNX. We continued work on work patterns in software engineering [17], empirical methods [19], and also focused tools for analysis of traces [14] [16]. Results were presented at CSER and Cascon.

One interesting piece of research [21] involved developing a user interface method to slide back and forth through model history. IBM filed for a patent on this CAS work, although it did not issue since there others had had similar ideas.

I was asked to write about my experiences working with the companies such as IBM CAS and Mitel; the result was a book chapter [18].

5. THE UMPLE ERA (2006-PRESENT)

For many years I had been interested in modeling. My PhD research into knowledge management was one form of modeling, and the OMT notation I started teaching in 1990, later morphing into UML, was another.

I had also long been interested in metaprogramming and code generation, doing a bit of this while working at BNR, in CODE4 and in TkSee as a way to solve ordinary software engineering problems. However the widely-available tools for code generation from models just were not very good, especially in the classroom environment – they were either expensive or too complex, or generated code that was not inspectable, or simply stubs of code as opposed to complete systems. I wanted to rectify this situation.

By 2001 I had amassed a library of UML modeling problems and solutions developed through my teaching experiences, and used these as the nucleus of a textbook I wrote with Robert Laganière [20]. This knowledge imparted in the book owes a lot to interactions with people at Cascon, IBM and CSER. Bran Selic wrote the Preface.

During my teaching and book writing I discovered students could learn to draw diagrams, but without code generation they had a hard time getting them right.

So with IBM CAS funding starting in 2006 we developed the open-source tool Umple [33], which stands for Simple, Ample, UML Programming Language. It permits textual modeling of UML constructs and generates top quality code; moreover it is written in itself.

Much of the early work on Umple was done by students at IBM CAS Ottawa, and early versions of Umple were embedded in Rational Software Architect.

The work on Umple has continued to this day, and we have regularly presented Umple advances at Cascon [23] [24] [25], including at the 2015 conference [26].

One element of our work was creating a taxonomy [22] of software projects so we could ensure empirical research, including work on Umple, covered a full spectrum of applications.

Umple has been worked on by over 60 students and industrial open-source practitioners in the years since it was started, and it is actively used in several universities and companies. It would never have been started, and all these students would not have gained the experience they have without the keen interest of IBM and its Centre for Advanced Studies.

6. CONCLUDING THOUGHTS

My involvement with CAS and Cascon (and the related CSER) has been multi-faceted.

I was on the CASCON program committee for many years, and was very impressed by the quality of peer review. In fact, one of my papers rejected at Cascon was later accepted at ICSE [34], the top software engineering conference.

Overall, my team and I presented 10 Cascon papers, several Cascon workshops and posters at the Technology Showcase almost every year for many years. My Cascon papers, according to Google Scholar have received over 600 citations, and almost all of my 120 publications (cited over 4000 times) have come directly or indirectly from research stemming from the CAS and CSER work described in this paper.

CAS has also affected software engineering education internationally. I became involved in establishing our University of Ottawa Software Engineering program in 1997, and discussions with people at Cascon had a lot of influence. Between 2002 and 2004 I became involved in the IEEE/ACM SE2004 project [27] to develop SE education standards, and again many people from the CAS community helped provide input and feedback.

Finally, the time my students and I spent at CAS Ottawa will always be remembered for both the intellectual challenges and the warm receptions we received from the IBM developers and managers we worked with.

7. REFERENCES

[1] Sayyad-Shirabad, J., Lethbridge, T.C. and Lyon, S, (1997), "A Little Knowledge Can Go a Long Way Towards Program Understanding", 5th International Workshop on Program Comprehension, Dearborn, MI, pp. 111-117.

[2] Anquetil, N. and Lethbridge, T.C. (1997) "File Clustering Using Naming Conventions for Legacy Systems", CASCON '97, Toronto, October, pp. 184-195.

[3] Singer, J., Lethbridge, T.C., Vinson, N, and Anquetil, N (1997) "An Examination of Software Engineering Work Practices", CASCON '97, Toronto, October, pp. 209-223 (republished in 2010)

[4] Anquetil, N., and Lethbridge, T.C. (1998), "Assessing the Relevance of Identifier Names in a Legacy Software System", CASCON 1998, 213-222.

[5] Zayour, I. and Lethbridge, T.C. (2000), “A Cognitive and User Centric Based Approach For Reverse Engineering Tool Design”, CASCON 2000, Toronto, November, pp. 16-30.

[6] Sayyad Shirabad, J., Lethbridge, T. and Matwin, S . (2000), “Supporting Maintenance of Legacy Software with Data Mining Techniques”, CASCON 2000, Toronto, November, pp. 137-151.

[7] Lethbridge, T.C. and Singer, J. (2001). “Experiences Conducting Studies of the Work Practices of Software Engineers”, in Erdogmus, H. and Tanir, O. Eds, Advances in Software Engineering: Comprehension, Evaluation and Evolution. Springer-Verlag, ISBN 0-387-95109-1, pp. 53-76.

[8] Lethbridge, T.C. and Herrera, F. (2001). “Towards Assessing the Usefulness of the TkSee Software Exploration Tool: A Case Study”, in Erdogmus, H. and Tanir, O. Eds, Advances in Software Engineering: Comprehension, Evaluation and Evolution. Springer-Verlag, ISBN 0-387-95109-1, pp. 77-98.

[9] Liu, H. and Lethbridge, T.C. (2001) “Intelligent Search Techniques for Large Software Systems”, CASCON 2001, pp 40-54.

[10] Sayyad Shirabad, J., Lethbridge, T.C. Matwin, S. (2003) “Applying Data Mining to Software Maintenance Records”, proc CASCON 2003, Toronto, October, IBM, in ACM Digital Library, pp. 136-148.

[11] Lethbridge, T.C., Sander Tichelaar, and Erhard Ploedereder (2003) “The Dagstuhl Middle Metamodel: A Schema for Reverse Engineering”, ATEM 2003, First International Workshop on Meta-Models and Schemas for Reverse Engineering, WCRE, Victoria, B.C., in Electronic Notes in Theoretical Computer Science, Elsevier, Vol. 94, pp 7-18.

[12] Anquetil, N., and Lethbridge, T.C. (2003), “A Comparative Study of Clustering Algorithms and Abstract Representations for Software Remodularization”, IEE Proceedings - Software, pp. 185-201. Winner of the Mather Premium award.

[13] Lethbridge, T.C., Singer, J and Forward, A., (2003) “How software engineers use documentation: the state of the practice”, IEEE Software special issue: The State of the Practice of Software Engineering, Nov/Dec 2003, pp 35-39.

[14] Hamou-Lhadj, A., and Lethbridge, T.C., (2004) “A Survey of Trace Exploration Tools and Techniques”, CASCON 2004, Toronto, October, IBM, in ACM Digital Library, pp. 42-55.

[15] Lethbridge, T.C., Sim, S., and Singer, J. (2005), “Studying Software Engineers: Data Collection Methods for Software Field Studies”, Empirical Software Engineering, 10 (3), July 2005, pp. 311-341.

[16] Hamou-Lhadj, A., Lethbridge, T.C., and Fu, L. (2005), “SEAT: A Usable Trace Analysis Tool”, International Workshop on Program Comprehension (IWPC), St. Louis, May, IEEE Computer Society Press, pp. 157-160.

[17] Murray, A. and Lethbridge, T.C. (2005), “On Generating Cognitive Patterns of Software Engineering” CASCON 2005, Toronto, October, IBM, in ACM Digital Library, pp. 129-139.

[18] Lethbridge, T.C., Lyon, S., and Perry, P. (2007). 'The Management of University - Industry Collaborations Involving Empirical Studies of Software Engineering', Shull, F., Singer. J, and Sjøberg, D. Eds, Guide to Advanced Empirical Software Engineering, Springer, pp. 257-284

[19] Singer, J., Sim. S., and Lethbridge, T.C.. (2007). 'Software Engineering Data Collection for Field Studies', Shull, F., Singer. J, and Sjøberg, D. Eds, Guide to Advanced Empirical Software Engineering, Springer, pp. 9-34

[20] Lethbridge, T.C. and Laganière (2004), R. “Object-Oriented Software Engineering: Practical Software Development in UML and Java”, McGraw Hill, 2nd ed

[21] Farah, H. and Lethbridge, T.C. (2007), “Temporal Exploration of Software Models: A Tool Feature to Enhance Software Understanding”, WCRE 2007, Vancouver, IEEE Computer Society, pp. 41-49.

[22] Forward, A. and Lethbridge, T.C. (2008) “A Taxonomy of Software Types to Facilitate Search and Evidence-Based Software Engineering”, Cascon 2008, IBM and ACM, pp. 179-191.

[23] Forward, A., Lethbridge, T.C., and Brestovansky, D. (2009), “Improving Program Comprehension by Enhancing Program Constructs: An Analysis of the Umple language”,

International Conference on Program Comprehension (ICPC) 2009, Vancouver, IEEE Computer Society, pp. 311-312.

[24] Badreddin, O., Forward, A., and Lethbridge, T. (2012), “Model Oriented Programming: An Empirical Study of Comprehension”, Cascon, ACM. Winner of Best Student Paper award.

[25] Garzon, M., Lethbridge, T.C., Aljamaan, H., and Badreddin, O. (2014) “Reverse Engineering of Object-Oriented Code into Umple using an Incremental and Rule-Based Approach”, Cascon 2014, ACM, 91-105

[26] Aljamaan, H., Lethbridge T.C. (2015) “MOTL: a Textual Language for Trace Specification of State Machines and Associations”, Cascon 2015, ACM.

[27] Lethbridge, TC, LeBlanc, RJ, Sobel, AEK, Hilburn, TB, Diaz-Herrera, JL (2006) "SE2004: Recommendations for undergraduate software engineering curricula", IEEE Software 23 (6), 19-25

[28] Skuce, D. and Lethbridge, T.C. (1995). 'CODE4: A Unified System for Managing Conceptual Knowledge'. International Journal of Human-Computer Studies 42, pp. 413-451.

[29] Whitney, MJ., Bernstein, M., de Mori, R., Kontogiannis, K., Corrie, B., Müller, H., Tilley, SR., Merlo, E., Mylopoulos, J., Wong, K., Johnson, JH., McDaniel J, Stanley M (1995) "Using an integrated toolset for program understanding" CASCON 1995, pp. 59-75

[30] Holt, RC, Schürr, A., Sim, SE, Winter, A. (2006), “GXL: A graph-based standard exchange format for reengineering”, Science of Computer Programming 60 (2) April, pp149-170

[31] Holt, RC, (1997) “An Introduction to TA: The Tuplie-Attribute Language”, http://www.swag.uwaterloo.ca/pbs/papers/ta.html

[32] Ebert, J, Kontogiannis, K, Mylopoulos, J, (2001), “Interoperability of Reengineering Tools” Dagstuhl Seminar, http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=01041

[33] Umple (2015) “Umple User Manual”, http://manual.umple.org

[34] Anquetil, N. and Lethbridge, T.C., (1998), "Extracting Concepts from File Names; a New File Clustering Criterion", *20th International Conference on Software Engineering,* Japan, April, pp. 84-93

Reflections on Twenty-five Years of CAS Innovation

James R. Cordy
School of Computing, Queen's University
Kingston, Ontario, Canada
cordy@cs.queensu.ca

ABSTRACT

Twenty-five years ago IBM Canada founded the Centre for Advanced Studies at the IBM Toronto Laboratory, and a year later hosted the very first CASCON conference, which was shortly to grow to be the premiere Canadian venue in computer science and software engineering. From the beginning CAS and CASCON have been a catalyst and key enabler for Canadian researchers, kickstarting one of the largest and best known computing and software engineering communities in the world. In particular, CASCON has been the platform that has launched two of the key innovations of my own research career. In this paper I reflect on the important role that CAS and CASCON have played and continue to play in the Canadian computing research community, and tell the stories of these two results and their eventual impact on research and practice.

Keywords

IBM CAS, CASCON, 25 Years

1. INTRODUCTION

Twenty-five years ago IBM Canada founded the IBM Centre for Advanced Studies (CAS) [29, 28] at the IBM Toronto Laboratory, and a year later hosted the very first CASCON conference [19], highlighting collaborative research results with CAS university partners and students. Recognizing the need for a broader Canadian conference to bring together researchers from across this vast country, within a few years CASCON had taken on a broader mandate, and soon grew to become the premiere conference in computer science and software engineering in Canada. Today it is one of the most important international conferences in the world in areas such as software engineering, database technology, and the smart internet.

Meanwhile, the Centre for Advanced Studies itself rapidly became a model for university-industry interaction. Under the founding leadership of CAS Director Jacob Slonim and Associate Director Arthur Ryman, CAS sought out university professors to partner with IBM developers and researchers in a relationship that was unique in the world at the time. Collaborations of a different kind were forged, and soon innovations in compiler technology, the seeds of model-driven architecture, radically new database interaction techniques, and other ground-breaking innovations became regular products of CAS partnerships. CAS' success rapidly became visible world-wide, and other companies and universities began to use the CAS model as a template for successful university-industry relationships. In 2006 this success was recognized by NSERC with the Leo Derikx Award for Excellence in continuing university-industry interaction, and under the leadership of current Director Joanna Ng, CAS continues today with innovations in business intelligence, cloud computing, smart applications and services, big data and more.

2. FORGING A RESEARCH PLATFORM

The influence of CAS and CASCON on the Canadian academic computing community cannot be overestimated. In many ways, the dominant success and prominence of Canadian researchers in international software engineering and database technology can be attributed in large part to the opportunities for meeting, exchanging ideas and forging new research partnerships that CASCON has facilitated over the years. In a country as large as ours, distance can be a hindrance to building the necessary "critical mass" for a vibrant research community, and there is no doubt that the annual gathering at CASCON has been a key part of building the relationships between researchers that have propelled the Canadian computing community to international pre-eminence in areas such as software analysis and maintenance, where up to 40% of papers presented at international conferences and workshops come from Canada.

A case in point is the founding of CSER, the Consortium for Software Engineering Research, which began in the mid-1990s as a result of the annual meeting of researchers from several CAS related projects and others in the software engineering area at CASCON in Toronto. With the strong support of CAS, the National Research Council, several industrial partners and a group of ten universities from across the country, CSER has been a unique collaboration platform and a continuing satellite event attracting software engineering researchers from across the country for two decades now. In 2000 CSER also won NSERC's Leo Derikx Award, thus tracing two such recognitions to the influence of CAS.

3. LAUNCHING YOUNG INNOVATORS

One of the unique aspects of CASCON is its lack of a registration fee. This may seem like a small thing, but for student attendees it has been an important enabler. Thanks to the continuing and generous support of IBM Canada, four generations of young Canadian researchers have been able to gather each year to present their research, seek feedback from both industry and academia, meet their student and senior colleagues from across the country, and begin to forge the relationships that would help them to find their place in the Canadian research and industrial communities.

Virtually all of the current top researchers in the country in software engineering, database technology and other areas were first introduced to the research world at CASCON, most often in the Technology Showcase, where early results and student submissions have always been welcome. This unique aspect of CASCON has introduced generations of Canadian student researchers, including virtually all of my own graduate students, to presenting and discussing their work in public with experts and other researchers. The presence of IBM Canada laboratory developers as well as academics at the showcase has helped students to understand both how their work relates to other research, and perhaps more importantly, how it can be focussed to address the real issues of software practice.

In many cases, these encounters have also opened new opportunities for collaboration and contacts for CAS and other IBM internships or later employment with IBM. Almost all of the twenty or so of my own students who have gone on to positions at IBM first met their IBM contacts at the Technology Showcase of CASCON. Similarly, CASCON is frequently the place where those going on to academic positions have first met the researchers of other institutions that they would later go on to join. When I attend CASCON now, some of the new research students I meet are of the fourth generation of CASCON students - supervised by the former students of the former students of the students that I first met at CASCON over two decades ago.

4. A KEY TO MY RESEARCH CAREER

CAS has been a part of my own research career since its inception, and my association with the IBM Toronto Laboratory goes back even further, beginning with a collaboration on the compiler technology S/SL [20] with Profs. Ric Holt and David Wortman at the University of Toronto in the mid-1980's which later became a part of the IBM compiler technology suite [7]. Thus I can recall the excitement and anticipation in the research community as the new IBM Centre for Advanced Studies was planned and founded, and my pride at appearing in the very first CAS annual report in 1990 [27].

Virtually all aspects of my research have appeared at CASCON at some point, either as Technology Showcase demonstrations of early versions of research tools such as TXL [12], presentations of new research ideas such as NICAD [25] at CSER, or as full CASCON research papers on topics such as early model-driven engineering [14], database migration [1], autonomic systems [16], parsing technology [31, 32], or web applications and services [24, 4].

Two particular stories come to mind to demonstrate the influence of CASCON on my research and industrial career.

4.1 Design Recovery and Legasys

Perhaps the biggest influence on my personal career originated with one of my first CAS collaborations, with CAS Associate Director Arthur Ryman and my colleague Prof. David Lamb at Queen's. As part of this collaboration Arthur and David described one of the earliest visions of what we now call Model Driven Engineering (MDE). Dubbed the "Theory-Model Paradigm", the idea called for round-trip software engineering, in which mathematical models are used to specify the high-level characteristics of software systems, from which code could be automatically derived, and conversely code could be analyzed to automatically extract its high-level model, completing the cycle and forming a software engineering method based solidly on mathematics. The "theory" played the role of what we now call a "metamodel" in MDE. Arthur's vision was later prototyped, complete with model visualization, in the 4Thought framework [26].

The Theory-Model Paradigm presented two interesting research problems for myself and my students - automating the forward engineering aspect, that is, deriving code from models (then called "meta-programming"), and automating the reverse engineering aspect, that is, extracting models from code ("design recovery"). Biggerstaff [6] had recently described a framework for something similar to the latter process, and 4th generation languages such as IDMS [15] and MARK IV [21] had been generating code from higher level descriptions for over two decades.

The 4Thought challenges were more ambitious, however - Arthur had described a formal, mathematical description of theories and models as Entitity-Relationship (ER) databases, and the challenge was to generate or analyze code to and from formal ER models in Prolog representation. At the time we had just completed the first public release of the TXL source transformation system [12], and we thought that, since Prolog is a source representation, perhaps source transformations between the Prolog ER model representation and C or other language code would be a way to address these issues. Arthur didn't agree, and I can recall many lively discussions about whether or not we could use TXL to do what he had in mind.

In the end, both worked out well. Working with David and Arthur, PhD student Kevin Schneider designed a formal architectural theory for Turing+ programs [22], and very quickly demonstrated the design recovery of an ER architectural model from source code, using a multi-stage TXL source transformation that, in 1991, took some 16 hours of computing time (the same transformation now takes seconds on modern hardware). This tool was later demonstrated in the Technology Showcase of CASCON 1994, where it was seen by the former University of Toronto student colleagues who were later to become our business partners.

Meanwhile, MSc student Medha Shukla demonstrated a source transformation to generate C code from formal ER models using a novel second-order process that instantiated a library of example-based code templates as assembled programs specified as ER design models [14]. This technique was far ahead of its time, and 17 years later became the subject of my keynote at the International Conference on Software Language Engineering (SLE'09) [9].

Intrigued by the demonstration they had seen at CASCON'94, my two old student colleagues visited Queen's in 1995 to explore whether Kevin's design recovery technique could be applied to COBOL code, and in particular to the problem of code analysis for Year 2000 risks. Crafting a COBOL grammar was a challenge, but with Queen's colleague Prof. Donald Jardine we worked quickly to demonstrate analysis of COBOL code for potential customers at the Bank of Nova Scotia. Shortly thereafter Kevin, Donald and I founded Legasys Corporation in Kingston to develop a custom formal theory for Y2K date risks and a custom design recovery process COBOL programs to identify them by extracting an ER model from source code. In essence, this was a task-specific version of Arthur's original idea.

In 1995 we demonstrated the first version of our COBOL date analysis, called LS/2000, at CASCON, this time as an industrial partner contributor to the Technology Showcase. The demonstration was seen by many in the data processing community, but in particular by CAS faculty partner Prof. Frank Tompa, then head of the department at the University of Waterloo. As a result, when companies and government approached Frank for advice on addressing the Year 2000 problem, he was able to point them at our new company, and the rest, as they say, is history.

Having already contracted to work with the Bank of Nova Scotia on their Y2K efforts, on Frank's advice we were approached by IBM Global Services in 1996. After additional demonstrations and discussions IBM contracted an exclusive license to Legasys' technology, and built a special purpose Transformation 2000 processing centre in Scarborough to run LS/2000. Over the next five years IBM and Legasys analyzed and converted more than four billion lines of COBOL, RPG and PL/I source code to identify and eliminate Year 2000 risks for customers in Canada, the USA and abroad.

At its peak Legasys itself employed more than 30 people, including a group of five PhDs who very rapidly advanced the research and tools to scale to industrial systems, and when we returned to academia in 2001 we were able to transfer some of these advances to our academic research. The result was a sequence of more than thirty contributions in software maintenance automation that were published in the top venues in software maintenance and advanced the research of ourselves and many others in the early 2000's. In addition to advances in design recovery [18] and source text processing [23], fundamentally new techniques such as code design queries [8] and agile parsing [17] have opened new opportunities that we continue to explore to this day.

4.2 Clone Detection, NICAD and SIMONE

One of the distinguishing characteristics of CASCON as a conference has been its openness to the presentation of new ideas. I don't know whether this is a reflection of its uniquely Canadian character, its concerted efforts to attract new student research, or simply an accident of the composition of program committees over the years, but I have seen more innovative ideas first presented at CASCON than at any other conference that I attend.

A case in point is our own work on clone detection using source transformation. At Legasys, of necessity we had tuned and reengineered the TXL source transformation engine to scale to transformations of millions of lines of source code. When we returned to academia, this scalability opened the door to new ideas that would simply have been rejected as impractical in our previous research.

I developed one such idea with MSc student Nikita Synytskyy and Prof. Tom Dean at Queen's. We designed a simple, intuitive way to find meaningful near-miss (i.e., slightly changed) code clones - fragments of code in a system that have been copied, pasted and modified. Our method consisted of extracting and collecting all of the fragments of interest (e.g., methods or classes), parsing and pretty-printing them to eliminate commenting and formatting differences, and then using text line differencing to identify similar ones.

What was exciting about this method is that, using the robust parsing technique we had previously presented at CASCON 2003 [31], we could handle malformed and mixed-language sources, such as HTML web pages, and adapt to any language for which we had a TXL grammar. Moreover, it could identify "near-miss" clones, ones that had been copied and changed, in a simple and precise way that existing clone detection methods could not. The challenge of comparing each fragment to every other was handled using an exemplar-based comparison algorithm. We demonstrated this technique on the optimization of two large websites.

Like many good ideas, this one was too simple. So simple, that when it was submitted to a top conference in program comprehension, the reviewers rejected it outright, stating "There is nothing very novel here; while this particular algorithm may not have been proposed previously, it is not particularly innovative." As a matter of fact, that community was so convinced that there was nothing new to be done in clone detection that they gave our paper the worst reviews I have ever seen. Fortunately, we chose to try again and submit the paper to CASCON, where new ideas and practical issues are more readily accepted, and our work was reviewed with the insightful comment, "The approach is very simple, but promising. It has the potential to offer a relatively easy and non intrusive way of identifying and later removing near-miss clones to aid in reducing complexity." It seems that this first reviewer was also not the only member of the CASCON program committee to see the potential in the approach - our paper was awarded Best Paper at CASCON 2004 [10].

In the end, the CASCON reviewers were right about the potential of our simple method. (Actually, they saw it more clearly that we did!) In the following years, PhD student Chanchal Roy carried on the idea, and together we tuned and reimplemented it for generality, accuracy, scalability and performance to produce one of the most popular clone detectors, NICAD [25], which has been used in scores of clone analysis studies in academia and industry. NICAD takes advantage of the language independence of the technique in a plug-in architecture that has been used for more than a dozen languages, including C, C++, Java, C#, Python, PHP, Javascript, XML, WSDL, Make, and others. It has scaled to analyze systems of more than 60 million lines [13], and has recently been shown to be the most accurate code clone detection tool in the world [30].

The NICAD framework has been extended to handle graphical models as well as code, in the model clone detector SIMONE [3]. SIMONE analyzes Simulink and Stateflow models to find near-miss submodel clones in automotive industrial models such as those at General Motors [2]. SIMONE itself has been extended to work on behavioural models such as UML sequence diagrams [5], and there is more to come.

The foresight and vision displayed by the CASCON community in understanding and accepting the potential of our simple idea all those years ago still surprises me. We could not have foreseen that our original simple, naive idea would spawn a decade of research yielding over 30 direct contributions, 9 patent references, more than 500 citations and a research tool with hundreds of users in academia and industry. In light of this influence, at CASCON 2014 our paper was recognized as the Most Influential Paper of 2004 [11].

5. THE CASCON TEAM

Over the past 25 years there have been many more examples of beginnings at CASCON that have shaped my career both formally and informally, as for example the discussion in the hallway with Prof. John Mylopoulos that began my sabbatical and collaboration with the University of Trento in design recovery of requirements from natural language legal documents. These less formal interactions are a big part of what makes CASCON so valuable.

Over the years I have served on the program committee of CASCON many times, and have chaired the conference twice. As a member of the CASCON steering committee and the CAS advisory committee at various times, I've had the chance to witness the dedication and commitment of the many people who make it possible up close. Jacob, Arthur, Kelly, Gabby, Joe, Joanna, Stephen, Marin, Jimmy, Cheryl, Debbie, there have been simply too many of you to list in detail, and I can't possibly remember everyone. You've turned CAS and CASCON into a national treasure, and on behalf of the whole Canadian research community, I thank you all.

6. REFERENCES

[1] R. Abu-Hamdeh, J. R. Cordy, and P. Martin. Schema Translation Using Structural Transformation. In *CASCON '94*, pages 202–215, 1994.

[2] M. H. Alalfi, J. R. Cordy, and T. R. Dean. Analysis and Clustering of Model Clones: An Automotive Industrial Experience. In *CSMR-WCRE'14*, pages 375–378, 2014.

[3] M. H. Alalfi, J. R. Cordy, T. R. Dean, M. Stephan, and A. Stevenson. Models Are Code Too: Near-miss Clone Detection for Simulink Models. In *ICSM '12*, pages 295–304, 2012.

[4] A. A. Almonaies, M. H. Alalfi, J. R. Cordy, and T. R. Dean. Towards a Framework for Migrating Web Applications to Web Services. In *CASCON '11*, pages 229–241, 2011.

[5] E. Antony, M. H. Alalfi, and J. R. Cordy. An Approach to Clone Detection in Behavioural Models. In *WCRE'13*, pages 472–476, 2013.

[6] T. J. Biggerstaff. Design recovery for maintenance and reuse. *Computer*, 22(7):36–49, July 1989.

[7] I. H. Carmichael and S. Perelgut. S/SL Revisited. In *CASCON '95*, pages 11–20, 1995.

[8] J. Cordy, K. Schneider, T. Dean, and A. Malton. HSML: Design Directed Source Code Hot Spots. In *IWPC '01*, pages 145–, 2001.

[9] J. R. Cordy. Eating our own Dog Food: DSLs for Generative and Transformational Engineering. In *GPCE'09 and SLE'09*, pages 1–1, 2009.

[10] J. R. Cordy, T. R. Dean, and N. Synytskyy. Practical Language-independent Detection of Near-miss Clones. In *CASCON '04*, pages 1–12, 2004.

[11] J. R. Cordy, T. R. Dean, and N. Synytskyy. Practical language-independent detection of near-miss clones. In *CASCON '14*, pages 2–2, 2014.

[12] J. R. Cordy, C. D. Halpern-Hamu, and E. Promislow. TXL: A Rapid Prototyping System for Prog. Language Dialects. *Comput. Lang.*, 16(1):97–107, Jan. 1991.

[13] J. R. Cordy and C. K. Roy. Tuning Research Tools for Scalability and Performance: The NICAD Experience. *Sci. Comput. Prog.*, 79:158–171, Jan. 2014.

[14] J. R. Cordy and M. Shukla. Practical Metaprogramming. In *CASCON '92*, pages 215–224, 1992.

[15] Cullinane Corporation. IDMS - the Data Dictionary Driven Data Management System, 1971.

[16] E. Dancy and J. R. Cordy. STAC: Software Tuning Panels for Autonomic Control. In *CASCON '06*, 2006.

[17] T. R. Dean, J. R. Cordy, A. J. Malton, and K. A. Schneider. Agile parsing in txl. *Automated Software Engg.*, 10(4):311–336, Oct. 2003.

[18] T. R. Dean, J. R. Cordy, K. A. Schneider, and A. J. Malton. Using Design Recovery Techniques to Transform Legacy Systems. In *ICSM '01*, pages 622–631, 2001.

[19] A. Gawman, J. K. Pachl, J. Slonim, and A. Stilman, editors. *Proc. 1991 Conf. of the Centre for Adv. Studies on Collab. Research.* IBM, 1991.

[20] R. C. Holt, J. R. Cordy, and D. B. Wortman. An Introduction to S/SL: Syntax/Semantic Language. *ACM Trans. Prog. Lang. Syst.*, 4(2):149–178, Apr. 1982.

[21] Informatics, Inc. MARK IV, 1971.

[22] D. A. Lamb and K. A. Schneider. Formalization of Information Hiding Design Methods. In *CASCON '92*, pages 201–214, 1992.

[23] A. Malton, K. Schneider, J. Cordy, T. Dean, D. Cousineau, and J. Reynolds. Processing Software Source Text in Automated Design Recovery and Transformation. In *IWPC '01*, pages 127–, 2001.

[24] A. Y. Mao, J. R. Cordy, and T. R. Dean. Automated Conversion of Table-based Websites to Structured Stylesheets Using Table Recognition and Clone Detection. In *CASCON '07*, pages 12–26, 2007.

[25] C. K. Roy and J. R. Cordy. NICAD: Accurate Detection of Near-Miss Intentional Clones Using Flexible Pretty-Printing and Code Normalization. In *ICPC '08*, pages 172–181, 2008.

[26] A. Ryman. Foundations of 4Thought. In *CASCON '92*, pages 133–155, 1992.

[27] A. Ryman and R. Helms. Centre for Advanced Studies 1990 Project Summary. Technical report, IBM Canada Laboratory, 1991.

[28] J. Slonim, M. Bauer, P.-A. Larson, J. Schwarz, C. Butler, E. Buss, and D. Sabbah. The Centre for Advanced Studies: A model for applied research and development. *IBM Syst. J.*, 33(3):382–396, 1994.

[29] J. Slonim, P. Larson, D. Sabbah, M. Bauer, and E. Buss. CAS: A Centre for Applied Research and Development. In *CASCON '92*, pages 91–112, 1992.

[30] J. Svajlenko and C. K. Roy. Evaluating Clone Detection Tools with BigCloneBench. In *ICSME'15*, pages 131–140, 2015.

[31] N. Synytskyy, J. R. Cordy, and T. R. Dean. Robust Multilingual Parsing Using Island Grammars. In *CASCON '03*, pages 266–278, 2003.

[32] A. D. Thurston and J. R. Cordy. A Backtracking LR Algorithm for Parsing Ambiguous Context-dependent Languages. In *CASCON '06*, 2006.

How to Train a Software Engineer in Canada: Through IBM CAS

Eleni Stroulia

Department of Computing Science
University of Alberta
stroulia@ualberta.ca

ABSTRACT

CAS has been an invaluable resource to the Canadian software-engineering community and myself personally. Through my collaborations with CAS, I have been able to explore a number of research problems throughout my academic career in Canada over the past 18 years. In this chapter, I reflect on my trajectory as a researcher and I discuss some key points that characterize the formative role that the IBM CAS played in it.

Keywords

Software-engineering, academia-industry collaboration, interdisciplinary research, R&D, CAS, Center for Advanced Studies, CASCON, IBM, Canada.

1. The Short Story

Every so often, I find myself surprised at where I am! My current role of a Software-Engineering Professor at the University of Alberta has not come to me through strategic planning and follow-through with studious labour. It has rather happened through a series of exciting opportunities, most of which did not seem very important at the time but turned out to be defining for my career as a faculty member in Canada and as a researcher in the field of software. Pursuing these opportunities has certainly taken a lot of hard work but it never felt like that! Reflecting on the sequence of these momentous accidents, I can see a few different common threads among them and none seems more important than my collaboration with IBM and CAS.

I found myself a newly hired Assistant Professor with the Department of Computing Science in 1997. During my PhD work in Artificial Intelligence, I developed a framework for reflecting on using a system's functional specification to reflect on its run-time behaviour and learn from its failures; the objective of this learning method was either to evolve the system specification to allow new behaviours, or to "fix" the system's implementation to meet its as-desired specifications. The methodology of my work relied on cognitive theories of reflective failure-driven learning and machine-learning methods for example-based learning. My view of "systems" was abstract and rather removed from their software implementations. In spite of the fact that my PhD research was not standard-issue software-engineering, I was lucky enough to land my position with the University of Alberta, eager to migrate my knowledge to my newly adopted area and to actually study and contribute to software engineering.

The rest of this paper tells the story of how I, and my academic progeny, developed in and contributed to software engineering, in close collaboration with IBM, CAS and CASCON.

2. My Work with IBM and CAS

Relatively to other contributors to this volume, I became involved with CAS relatively late, in the 2000s. Still my involvement has been formative to myself and my students, especially since when I joined the ranks of academic faculty I had just moved to software engineering.

2.1 Becoming a Canadian Software Engineer

My first software-engineering project was "CELlest" [1] [2] (CEL Legacy Enhancement Software Technologies) aiming at an automated method for developing web-based clients for legacy mainframe systems by reverse engineering the legacy-system interface and wrapping it with web-based and mobile clients. The project was funded through a CRD grant with a local SME, Celcorp, and I developed the proposal with the support and guidance of Paul Sorenson, my then department chair; Paul and I co-supervised my first PhD student, Mohammad El-Ramly. The key idea of this work was "to automate screen-scraping practices through machine-learning methods". Circumventing the problem of code understanding, the CelLEST toolkit learned a model of the legacy-system dynamic behaviour using as examples traces of its interaction with users. This behavioural model provided the basis for designing and (semi-)automatically constructing different types of front-end clients that could invoke the legacy-system interface.

This project provided me with a perfectly timed context, in which to evolve myself into a "software evolution and maintenance” researcher. The research topic was “hot” with the Year2K problem looming, my PhD research gave me the overarching methodology framework, and I got to study and learn everything about *legacy-system migration, reverse engineering*, and *dynamic analysis*.

I presented this work at the 2000 CASCON, in the context of the “Moving Legacy Systems to the Internet” workshop I co-organized with Kostas Kontogiannis, Assistant Professor at Waterloo at the time. This was not my first presentation on CELlest but it certainly felt like the most important at the time.

My first CASCON presentation was a unique learning experience for me, and it convinced me to continue returning to this conference every single year after that.

CASCON is a unique venue for software engineering: the audience includes key researchers in all software subdisciplines and IBM developers with deep knowledge of all current technologies. Presenting at CASCON demands a combination of theoretical contributions and potential for practical impact.

The next year (November 2001) IBM established the Eclipse consortium, and adopted the open-source licensing model for the Eclipse IDE. This was a uniquely inspired move by a software company, and it generated a huge response among academics, who, for the first time, could use and contribute to a “real tool” and have a whole community potentially trying and adopting their research software. Having attended a hands-on workshop at CASCON my then PhD student, Zhenchang Xing [3] [4] [5], focused his research on developing a plugin for analyzing, refactoring and evolving object-oriented code. Funded through two Eclipse innovation grants (JReflex 2003 and 2004), he developed UMLDiff, an algorithm for analyzing and comparing, at the UML-design level, subsequent versions of object-oriented code. UMLDiff, its implementation in the JDEvAN Eclipse plugin, the evolutionary analyses it supported, and its refactoring and version catch-up methods are still among my most highly cited contributions, which is, to a great extent, because of IBM’s support for the Eclipse community.

My experience working on Eclipse and the JDEvAn plugin committed me to a lifetime of industrially driven research: real-world practice and tools are fertile grounds of research problems and contributing to them affords realistic opportunities of technology adoption.

2.2 Replicating CAS in Alberta

By 2004 the CAS model, of academic research teams working on research projects driven by real-world practice in close collaboration with IBM teams, was mature and had demonstrated its effectiveness in enabling innovative research. Around that time, Paul Sorenson and Bernie Kollman (IBM VP, Public Sector Alberta) started working with the Government of Alberta to establish a new CAS in Alberta, to establish a locus of software research activity in Western Canada and help diversify the local economy.

As a mid-career professor at the University of Alberta at the time, I attended some of the organization meetings and I became keenly aware of the opportunities made possible by the collaboration of government, industry, and academia.

CAS Alberta was conceived as a partnership between government, industry, and academia. The government provides the regulatory framework based on the society needs and plays the role of a public entrepreneur. The industry, IBM, defines the scope of the activity, and contributes human and other resources. The University contributes human capital, with valuable knowledge and skills. The synergy enables co-creation of new knowledge through research, straightforward pathways to intellectual property creation, professional training for University students, and production of skilled personnel.

Since 2005, when CAS Alberta was established, my students and I have benefitted hugely from having two points of access to the broad community of IBM researchers.

2.3 Becoming an IRC

Through the process of developing CAS Alberta, I became aware of the nascent Service Science discipline, put forward by IBM. In the context of CAS, I started working closely with Jim Spohrer, Paul Maglio and especially Kelly Lyons, the then Director of CAS. The key premise of Service Science is that “an interdisciplinary approach [is called for] the study, design, and implementation of services systems – complex systems in which specific arrangements of people and technologies take actions that provide value for others.[1]”

By adopting the open-source model, and through its financial investment to related research, IBM nurtured a generation of researchers and developers and defined object-oriented development practice around the Eclipse IDE. CAS became the focal point of a substantial subset of this community. IBM and CAS have been following the same practice for a number of other technologies, substantially enhancing the impact potential of academic research.

Inspired by this idea, in 2008, with Kelly Lyons and Paul Messinger, Professor with the School of Business at the University of Alberta, we started working on service delivery through virtual-worlds platforms, becoming

[1] https://en.wikipedia.org/wiki/Service_science,_management_and_engineering

popular at the time with Second Life as the prime example. Our research was quite innovative in that it studied marketing problems, such as analyzing consumer perceptions and behaviours, relying on software platforms and methodologies. Publications produced through this interdisciplinary collaboration [6] [7] with equal involvement and participation by the academic (Paul Messinger and myself) and industry partners (Kelly Lyons and Stephen Perelgut) are being consistently and continuously cited until today.

The breadth of IBM's intellectual capital and the diversity of its research investments across disciplines, to a great extent enabled through CAS, played a key role in its development of the interdisciplinary service-science movement. This research agenda was at the forefront of a more general appreciation for interdisciplinary research as essential for tacking society's complex problems.

By 2009, I was inspired by and strongly committed to the Service-Science agenda, which, in my view, provides a rich context for software-engineering research, including economic and human-process concerns. My strong collaboration with IBM throughout the recent past and my deep interest in the Service-Science agenda led to IBM investing in my program through an Industrial Research Chair, co-funded by Alberta iCore (Informatics Circle of Research Excellence) and NSERC. From 2009 till today, this funding enabled me to substantially scale up my research team and productivity and to expand my research agenda around two major themes.

On one hand, my team focused on core software-engineering issues of designing, developing and evolving service-oriented software systems. With access to state-of-the-art IBM tools, my two PhD students, Mike Smit [9] [10] and Marios Fokaefs [11] [12], developed methods for the deployment and autonomic management of service-oriented applications on (virtual) infrastructures and for systematically evolving service-oriented compositions, in the WS* and the REST styles.

In parallel, my team has been working on three areas of intense software-development activity for the purpose of delivering innovative types of services: virtual worlds as a platform for business and training, web 2.0 support for distributed communities of practice, and sensor-based systems. Recognizing the importance of collaboration in software development, my PhD student Ken Bauer [8], focused on analyzing the collaboration patterns of development teams, continuing on some early ideas of the JReflex project. The Smart-Condo [13] project was also born out of this service-delivery theme. Aiming at supporting seniors to live independently longer in their own homes, the Smart-Condo platform integrates a variety of sensors to observe the activities of the home occupants, and to analyze them for evidence of problems that family members and formal care givers may be able to address. The Smart-Condo project involves researchers from Computing Science, Rehabilitation Medicine, Industrial Design, Pharmacy, Nursing and Education, and clinicians from the Glenrose Rehabilitation Hospital. Motivated by a social agenda, it has attracted many undergraduate and graduate students who have been part of a unique training sandbox.

2.4 Beyond Research

I would be terribly remiss if I constrained my reflection on my experience with CAS solely on my development as a researcher. CAS has played a key role in my overall professional development as well as that of my students.

I have already alluded to the challenge of addressing the CASCON audience of software-engineering research experts and developers of state-of-the-art software tools. Having gone through the experience myself first in 2000, I have made a point of having all of my students present their work to CASCON, whether in the main conference, or in the technology showcase, or the various associated workshops, and to the CAS University Days.

Quite beyond developing their presentations chops, my students have benefitted from their interactions with all the other senior researchers attending these events. With my mind's eye, I can clearly see Hausi Müller, Jim Cordy, and Gabby Silberman "grilling" my students, not unkindly but definitely persistently. And since most people join the CAS events year after year, the community has a long memory and students are motivated to keep "doing better".

CASCON and the associated CSER meeting enabled my graduate students, especially those pursuing a PhD, to develop a professional network during their studies, which has been instrumental for their career development: two of my PhD students joined Marin Litoiu's group as PDFs. One of them has since joined Dalhousie as an Assistant Professor and we continue to meet regularly ever year at CASCON.

2.5 Taking CAS Collaborations beyond CAS

CAS has served the role of a professional-training clearing house, for most Canadian software-engineering researchers.

The close-knit relationships that I have established with researchers in the CAS community throughout these past fifteen years have transferred beyond CAS.

Just to name a few examples, Kelly Lyons, Mark Chignell and I have worked closely in the context of the GRAND (GRaphics Animation and New meDia) NCE[2]; Kelly Lyons and I have just obtained a Strategic Project grant; Hausi

[2] http://www.grand-nce.ca

Müller, Marin Litoiu and I are currently collaborating in the context of the SAVI Strategic Network[3]; and Mark Chignell and I are now working in the same AGE-WELL NCE[4] work-package, based on the Smart-Condo project.

Rewarding and successful research collaborations beget future collaborations. CAS has been an incubator of a thriving collaborative research community across Canada.

3. Concluding Remarks

Hard as it is to trace the twists and turns that have brought me where I am, it is impossible to imagine what I might have been without CAS. It has been a formative influence in the research agenda that I have been pursuing, it has provide me with many of the tools I have used to conduct this research, and it has given me a forum to communicate and get feedback on the outcomes of my work. At the same time, it has given me a community, from which to draw creativity and support. I am looking forward to the future, which I am certain will be shared with many of the same people.

Acknowledgements

I am grateful to Kelly Lyons, Wendy Powley, Hanan Lutfiya, Hausi Müller, Kenny Wong, Marin Litoiu, Jim Cordy and Mike Bauer for their support, advice and good humour. I want to recognize Jacob Slonim, the famed first CAS Director, whom I got to meet much later; I have enjoyed tremendously my interactions with him and I have found his commitment to his students inspiring. I want to acknowledge the generous support of Don Aldridge, Stephen Perelgut, Monica Sawchyn, Bernie Kollman, Marcellus Mindel, and Joanna Ng; it has been a privilege and a pleasure with all of them. Finally, I want to thank all my students who enriched this journey.

REFERENCES

[1] Eleni Stroulia, Mohammad El-Ramly, Paul Sorenson, and Roland Penner. 2000. Legacy systems migration in CelLEST. In *Proceedings of the 22nd international conference on Software engineering* (ICSE '00). ACM, New York, NY, USA, 790-.

[2] Eleni Stroulia, Mohammad El-Ramly, Paul Iglinski, and Paul Sorenson. 2003. User Interface Reverse Engineering in Support of Interface Migration to the Web. *Automated Software Engg.* 10, 3 (July 2003), 271-301.

[3] Kenny Wong, Warren Blanchet, Ying Liu, Curtis Schofield, Eleni Stroulia, and Zhenchang Xing. 2003. JRefleX: towards supporting small student software teams. In *Proceedings of the 2003 OOPSLA workshop on eclipse technology eXchange* (eclipse '03). ACM, New York, NY, USA, 50-54.

[4] Zhenchang Xing and Eleni Stroulia. 2007. Differencing logical UML models. *Automated Software Engg.* 14, 2 (June 2007), 215-259.

[5] Zhenchang Xing and Eleni Stroulia. 2007. API-Evolution Support with Diff-CatchUp. *IEEE Trans. Softw. Eng.* 33, 12 (December 2007), 818-836.

[6] Eleni Stroulia, Mihnea Galeteanu, Paul Pacholski, Kelly Lyons, and Paul Messinger. 2010. Virtual worlds for modeling complex processes between people and systems. In *Proceedings of the 2010 Conference of the Center for Advanced Studies on Collaborative Research* (CASCON '10), Hausi A. Müller, Arthur Ryman, and Anatol W. Kark (Eds.). IBM Corp., Riverton, NJ, USA, 394-396.

[7] Paul R. Messinger, Eleni Stroulia, Kelly Lyons, Michael Bone, Run H. Niu, Kristen Smirnov, and Stephen Perelgut. 2009. Virtual worlds - past, present, and future: New directions in social computing. *Decis. Support Syst.* 47, 3 (June 2009), 204-228.

[8] Ken Bauer, Marios Fokaefs, Brendan Tansey, and Eleni Stroulia. 2009. WikiDev 2.0: discovering clusters of related team artifacts. In *Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research* (CASCON '09), Patrick Martin, Anatol W. Kark, and Darlene Stewart (Eds.). IBM Corp., Riverton, NJ, USA, 174-187.

[9] Mike Smit and Eleni Stroulia. 2011. Automated State-Space Exploration for Configuration Management of Service-Oriented Applications. In *Proceedings of the 2011 IEEE International Conference on Web Services* (ICWS '11). IEEE Computer Society, Washington, DC, USA, 468-475.

[10] Michael Smit and Eleni Stroulia. 2013. Simulating Service-Oriented Systems: A Survey and the Services-Aware Simulation Framework. *IEEE Trans. Serv. Comput.* 6, 4 (October 2013), 443-456.

[11] Marios Fokaefs and Eleni Stroulia. 2012. WSDarwin: automatic web service client adaptation. In *Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research* (CASCON '12), Hans-Arno Jacobsen, Ying (Jenny) Zou, and Judy Chen (Eds.). IBM Corp., Riverton, NJ, USA, 176-191.

[12] Marios Fokaefs and Eleni Stroulia. 2015. Using WADL Specifications to Develop and Maintain REST Client Applications. In *2015 IEEE International Conference on Web Services (ICWS)* pp.81-88, June 27 2015-July 2 2015

[13] Iuliia Vlasenko, Ioanis Nikolaidis, and Eleni Stroulia. 2015. The Smart-Condo: Optimizing Sensor Placement for Indoor Localization. In *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol.45, no.3, pp. 436-453, IEEE.

[3] http://www.savinetwork.ca

[4] http://agewell-nce.ca

Runtime Monitoring of Webservice Conversations

A Tale of One CAS Project

Jocelyn Simmonds
Computer Science Department
University of Chile
jsimmond@dcc.uchile.cl

CAS Student 2008-2009

Marsha Chechik
Department of Computer Science
University of Toronto
chechik@cs.toronto.edu

IBM CAS Faculty Fellow (2006-present)

1. HOW DID WE GET STARTED?

The WebSphere Integration Developer (WID) is a family of IBM products that enables companies to create and consolidate their business processes. The WID product suite allows the integration of traditional workflow processing based on human task interaction, as well as the choreography of activities between different systems, using a Service Oriented Architecture (SOA) [8]. A client solution is a set of mainframe applications, local or remote services, choreographed processes in BPEL, or components built with J2EE. Solutions implemented in WID are hard to debug because of the different technologies used, the distributed nature of services, and the existence of message queues.

In 2004, Marsha approached IBM with an idea to use automated verification, and, specifically, runtime monitoring, as a way to improve quality of client solutions. Runtime verification is a scalable technique that, when given some notion of correctness, monitors the system as it runs, checking for behavioral (or some other) conformance with the spec. Back then, Kelly Lyons was the head of CAS, and Marsha worked closely with Julie Waterhouse (IBM CAS), Bill O'Farrell (IBM Toronto Lab) and Grace Lo (IBM Toronto Lab) to get this proposal articulated and funded.

The intention of the project was to add different runtime monitoring components to the WID suite. We originally hoped to provide users with an intuitive way of defining specifications for systems described using IBM's newly developed Business State Machines (BSM) language. BSM offered a higher level of abstraction than traditional text-based languages, where programs are state machines that represent Java programs, and were run as web services, meaning that they can be distributed. BSM is a visual language, and it was hoped that this could be leveraged to improve understanding of analysis results.

We also aimed to study different techniques (both static and dynamic) for scalable checking for behavioral conformance in an industrial setting. Finally, we wanted to research how to provide end users with useful output from our tools which would enable them to debug their applications. Specifically, we wanted to propose and rank recovery plans which users can then select for execution.

2. UNDERSTANDING BUSINESS STATE MACHINES (2004 - 2006)

The first person with a CAS fellowship for this project was Jonathan Amir. As a Masters student, Jonathan was interested in improving the debugging tools offered by the WID suite. Jonathan obtained a IBM Fellowship to study the runtime verification of state machines, specified as Business State Machines (BSM), Over 2005, Jonathan worked closely with Bill O'Farell's group at the IBM Toronto Lab, and specifically with Jon Bennett and Julie Waterhouse (CAS), in order to achieve the goals of this project. Jonathan started by formalizing BSM, defining the operational semantics of the language by translating it to a state transition system. He also defined and implemented a prototype runtime monitoring framework for BSM, integrating it with WID. This first prototype allowed users to specify multiple runtime monitors using regular expressions, which were then converted into finite-state automata and used for runtime monitoring. Although the state-machine runtime, being built at IBM Rochester, was not yet ready at the time, Jonathan had a chance to conduct several case studies using a BSM simulator created by Bill O'Farrell's group. The results of this project are published in Jonathan's Masters thesis, titled "Runtime Monitoring of State Machines - Formalism and Analysis" [1]. Despite being a brilliant student, and having written an excellent thesis, Jonathan left the project without continuing to a Ph.D.

3. A CHANGE IN DIRECTION: BPEL (2007)

The WID runtime monitoring environment defined by Jonathan had the potential of being reused to analyze components specified in other languages. Bill O'Farrell suggested that we try our hand at BPEL (Business Process Execution Language) – an XML-based language that allows Web services in a service-oriented architecture (SOA) to interconnect and share data. BPEL has become the centerpoint of development of Bill's group. Yuan Gan, another Masters student at the Formal Methods group, took over Jonathan's project. She focused on updating Jonathan's monitoring components so as to allow the analysis of BPEL applications. The focus of this new project was on the analysis of the messages that are passed between web service partners – a conversation protocol – essentially a specification that describes the message exchange that partners are allowed to engage in. The analysis question that naturally arises in this context is: "given a collection of partners and a conversation protocol, does this collection satisfy the protocol?" That is, do the partners engage only in conversations specified by the protocol, and can they engage in all specified conversations?

After studying several case studies, Yuan identified UML 2.0 Sequence Diagrams (SD) as an adequate language for specifying *safety* (nothing bad can happen) and *liveness* (something good must happen) properties of BPEL applications. Starting with IBM Rational's SD editor, Yuan implemented a conversation specification editor. This editor then produced finite-state automata for a large sublanguage of Sequence Diagrams, including such features as negation, parallel composition and alternation. These automata were integrated with the runtime monitoring framework built in WID by Jonathan, resulting in a usable monitoring framework for BPEL conversations. The results of the analysis in the form of UML 2.0 Sequence Diagrams in the same editor as used for specification.

Yuan worked closely with Bill's group on integrating her tool, with Julie Waterhouse and then Elena Litani from IBM CAS, as well as with Shiva Nejati, a Ph.D. student in Toronto studying under Marsha's supervision. Together with Shiva, Yuan developed algorithms for translating SDs into automata and identified the kinds of processes that could be checked for safety and liveness. The results of this project were published in a CASCON 2007 paper [6] for which Shiva and Yuan received the Best Student Paper Award. They also formed the basis of Yuan's Masters thesis [5].

Overall, the project was deemed to be a success. Yuan joined Bill's group as a developer (and stayed at the IBM Toronto Lab until a few years ago). Yet the project was limited to looking at messages exchanged, while ignoring their content (parameter values) and, while appearing intuitive, there have been a number of issues with semantics of SDs that were identified but not resolved. More work needed to be done.

4. RECOVERY FROM ERRORS (2008 - 2010)

The project was continued by Marsha's next student (and a co-author of this bookchapter) – Jocelyn Simmonds – then a Ph.D. student at the Formal Methods laboratory in Toronto. She focused on two goals: (1) enriching the property specification language and, (2) at Bill O'Farrell's request, aiming to understand the causes of failure, in order to enable useful feedback to the users.

She and Shiva Nejati began by studying the expressive power of the UML 2.0 Sequence Diagrams language defined in [6], concluding that not only it can express a variety of safety and liveness properties, but it can also capture all of the patterns in the now standard Specification Pattern System (SPS) [2]. The SPS is a pattern-based approach to the presentation, codification, and reuse of property specifications. The system allows patterns like "event P is **absent** between events Q and S" or "S **precedes** P between Q and R" to be easily expressed in and translated between linear-time temporal logic (LTL), computational tree logic (CTL) and other state-based and event-based formalisms. This system has been used in analysis of various software artifacts, and is generally believed to be very comprehensive.

As part of this work, Jocelyn and Shiva included additional SD operators in the SD language, making certain types of properties easier to specify. They also defined a collection of SD templates for property specification, based on the SPS. With Yuan now working at IBM on Bill's team, they were able to update her specification editor to include this new version of the SD language. They also provided a complete specification of the SD property patterns, with the hopes that these would allow WID end users to specify properties in a more intuitive manner. Several case studies were conducted to determine how usable the resulting language was, as well as to understand the monitoring overhead added by the monitoring framework, where the sample applications and properties were provided by Bill's team. The new SD grammar, the SD property templates and the results of the case studies were published in [13, 14].

The second goal of the project was trying to understand the *causes of failure*. The goal of this work was to help users go from the *result* of error (as reported by the runtime monitor) to identifying what parts of their BPEL processes were at fault. In fact, monitoring the system as it ran provided a chance to recover from an error once a problem had been detected. This was deemed critical in the domain of web service applications, as bugs could have been potentially exposed to millions of users before they are found/fixed. Application termination was, of course, not a satisfactory solution in the case of long lived applications.

Working with Dr. Shoham Ben-David (Marsha's postdoctoral fellow at that time), Jocelyn defined a general strategy for dynamic recovery, depending on the type of property that was violated. For violations of properties capturing forbidden behavior, a recovery plan should attempt to return the application to an earlier state, one at which an alternative path that potentially avoids the fault is available. For violations of properties capturing desired behaviors, merely going back is insufficient to ensure that the system can produce the desired behavior. In this case, we used *planning* [3] to compute plans that attempt to redirect the application towards executing new activities that may lead to the satisfaction of the property in question. This research was published at several conferences [10, 12] and formed the foundation of Jocelyn's Ph.D. thesis [9] and a follow-on journal paper [11].

The project was done with active involvement and great help provided by Bill O'Farrell and his team members, including Yuan Gan, as well as members of CAS: Elena Litani and later Leho Nigul.

Was this extension of the project a success? Partially. We did produce several high quality publications, many co-authored with IBMers and validated on some medium-size web service examples. We produced a novel piece of technology, extending and combining several existing pieces of software. For example, we needed to analyze models of the BPEL application at runtime, so we modified an existing LTSA tool [4] to include the translation of BPEL with compensation. At that point in time, planners only returned the first plan found. Thus, we modified Blackbox [7], a SAT-based planner, so as to allow the computation of additional plans. This was accomplished by understanding how SAT-based planners work and modifying how they encode planning problems. We also implemented several new components, for example, calculating desirable application states from the property specification. We intended to file a patent application for it, with Bill's encouragement, although, for various reasons, this application was never finished.

Yet, we struggled with a number of aspects. For starters, the goal of the project was to enable runtime monitoring of complex web services. The word "complex" meant different things to the different people involved, especially when it came to case studies. For IBM, a complex example showed the integration capabilities of the WID suite. For us researchers, complex workflows had concurrent activities with compensation actions. Identifying examples that were "interesting" from both points of view required many rounds of discussion and took a very long time and lots of effort from Elena and Leho. And even then, most complex usage scenarios of IBM's WID tools were, naturally, written by or for IBM's customers, and thus were off-limits to us. That meant that our work was never validated on the truly complex systems, something repeatedly pointed to us by academic reviewers of our papers.

In addition, significant effort was required to turn academic prototypes into parts of an industrial tool. In that respect (and in many others), we are very grateful to Bill and his team for the support they provided. Without their help, we would not have been able to develop key parts of the monitoring framework. However, we met a roadblock when trying to implement a component which presented recovery plans to users. This portion of the WID product was outside the scope of Bill's group, and thus we had to resort to an adhoc solution instead of the planned integration of our tooling in the final product. This, of course, limited the scope and significance of our contribution.

5. IN CONCLUSION

We conclude this brief report with a reflection on the impact of this CAS project on us as well as some lessons learned.

First, this project was truly a team effort and our warmest thanks go to Bill O'Farell, Jon Bennett, Julie Waterhouse, Grace Lo, Elena Litani, Leho Nigul and others on the IBM side. We are very grateful to academic members of the project: Jonathan Amir (currently a software developer in Israel), Yuan Gan (a former IBMer, now with the Ministry of Transportation), Dr. Shiva Nejati (now a research scientist at the Software Verification and Validation Lab, at the University of Luxembourg), and Dr. Shoham Ben-David (currently a NECSIS program director at General Motors). We never would have done it without you! Special thanks go to Bill O'Farell, our one constant player throughout this long project, who always had time, good will and ideas for us.

And what have we learned? First and foremost, the complexity of handling real-world problems. We needed to ensure not only that our solutions provide correct analysis but also work hard at achieving scalability and usability of these solutions. Moreover, we needed to solve problems quite different from our original intent and for which we did not always have the required expertise. Thus, we needed to assemble a research team with complementary skills to address these. We benefitted greatly from having access to domain experts who could help us identify "real" problems. We also benefitted from having case studies, although this particular aspect of the problem was most difficult.

Where are we now? Jocelyn is still concentrating on problems of property specification, efficient monitoring and automated recovery. Yet, she is applying these ideas to a different domain – mobile apps. Marsha has spent the past five years working on issues of modeling and analysis of complex systems, as part of NECSIS – a Network for the Engineering of Complex Software-Intensive Systems for Automotive Systems, with General Motors and IBM as the main industrial partners.

6. REFERENCES

[1] Jonathan Amir. Runtime Monitoring of State Machines - Formalism and Analysis. Master's thesis, University of Toronto, Department of Computer Science, 2005.

[2] Matthew Dwyer, George Avrunin, and James Corbett. Property Specification Patterns for Finite-State Verification. In *Proceedings of the 2nd Workshop on Formal Methods in Software Practice (FMSP'98)*, pages 7–15, 1998.

[3] Richard Fikes and Nils J. Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Journal of Artificial Intelligence*, 2(3/4):189–208, 1971.

[4] Howard Foster, Sebastián Uchitel, Jeff Magee, and Jeff Kramer. LTSA-WS: a Tool for Model-Based Verification of Web Service Compositions and Choreography. In *Proceedings of the 28th International Conference on Software Engineering (ICSE'06)*, pages 771–774, 2006.

[5] Yuan Gan. Runtime Monitoring of Web Service Conversations. Master's thesis, University of Toronto, Department of Computer Science, 2007.

[6] Yuan Gan, Marsha Chechik, Shiva Nejati, Jon Bennett, Bill O'Farrell, and Julie Waterhouse. Runtime Monitoring of Web Service Conversations. In *Proceedings of the 2007 conference of the Centre for Advanced Studies on Collaborative Research (CASCON'07). Best student paper award*, pages 42–57, 2007.

[7] Henry Kautz and Bart Selman. Unifying SAT-based and Graph-based Planning. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 318–325, 1999.

[8] Jasmine Noel. BPM and SOA: Better Together. 2005.

[9] Jocelyn Simmonds. *Dynamic Analysis of Web Services*. PhD thesis, University of Toronto, Department of Computer Science, 2010.

[10] Jocelyn Simmonds, Shoham Ben-David, and Marsha Chechik. Guided Recovery for Web Service

Applications. In *Proceedings of Eighteenth International Symposium on the Foundations of Software Engineering (FSE'10)*, pages 247–256, 2010.

[11] Jocelyn Simmonds, Shoham Ben-David, and Marsha Chechik. Monitoring and Recovery for Web Service Applications. *Journal of Computing*, 95(3):223–267, 2013.

[12] Jocelyn Simmonds and Marsha Chechik. RuMoR: Monitoring and Recovery of BPEL Applications. In *Proceedings of 25th IEEE/ACM International Conference on Automated Software Engineering (ASE'10)*, pages 345–346, 2010.

[13] Jocelyn Simmonds, Marsha Chechik, Shiva Nejati, Elena Litani, and Bill O'Farrell. Property Patterns for Runtime Monitoring of Web Service Conversations. In *Proceedings of 8th International Workshop on Runtime Verification (RV'08). Selected Papers*, pages 137–157, 2008.

[14] Jocelyn Simmonds, Yuan Gan, Marsha Chechik, Shiva Nejati, Bill O'Farrell, Elena Litani, and Julie Waterhouse. Runtime Monitoring of Web Service Conversations. *IEEE Transactions on Service Computing*, 2(3):223–244, 2009.

International Collaboration

CAS and CASCON: Perspectives from an Outsider

Dennis Smith
Consultant
dbs754smith@gmail.com

ABSTRACT

This paper outlines my experience with CAS and CASCON as an outsider. For 22 years before retiring I was affiliated with the Software Engineering Institute (SEI) of Carnegie Mellon University. I participated in CASCON and related activities over a 15 year period. This participation was a very positive experience and this brief article highlights the participation, and summarizes some of the mechanisms that have led to success of CASCON. At the conclusion of the article, I suggest some ideas toward a framework that may be useful for analyzing CASCON, and offer ideas for avoiding some of the pitfalls that may emerge in the future.

Keywords

Collaborative partnerships, Cross-institutional partnerships, Business models, Open business models.

1. INTRODUCTION

The Centre for Advanced Studies (CAS) was founded in 1990 to facilitate the transfer of research ideas into products [1] [2]. CAS includes such components as 1) the CAS Research labs where graduate students spend a semester or year at Toronto labs working alongside IBM researchers 2) CAS research fellows which award fellowships to faculty members to support their work, enable collaboration between universities and IBM 3) support for CASCON, an annual conference in Toronto that attracts 1300 participants. The conference, which is the largest computer science conference in Canada, offers 3 keynotes, 30 refereed papers, 25 – 30 workshops, an emerging technology track, a technology showcase that has about 50 exhibits of technology projects, and networking opportunities during breaks and meals. The conference is wholly sponsored by IBM.

2. MY PERCEPTION AND EXPERIENCE OF CASCON

My first experience with CAS came in 1994 when I was in the process of hiring Scott Tilley who had just completed his PhD at the University of Victoria. One of his references was a senior employee of IBM. He said that he would have liked to hire Scott, because he had been a good employee. I couldn't figure out how Scott had been an employee of IBM because he had been a full time graduate student. However, as I shortly learned, he, along with many other graduate students, had six month to 1 year internships working at the Toronto labs.

As Steve Perelgut IBM Toronto's Senior Manager for Academic Partnerships later explained, this was simply good business. The program of offering internships to graduate students gave IBM a good understanding of how they worked in a professional technical environment, and a chance to try before they buy. It also provided high quality research expertise for teams at the Toronto Labs, and it saved a substantial amount of money in recruitment, training and potential turnover. In addition this experience sometimes led researchers to focus their PhD dissertations on problems were relevant for IBM. For those PhD students who chose not to pursue a career at IBM, these future leaders might become leaders of the scientific community or potential future CAS faculty research fellows.

A few years later an incident happened that illustrated the collaborative spirit that characterized CASCON. I saw Kostas Kontogiannis at a CASCON conference. We spent about 15 minutes talking about a new Independent Research and Development project (IRAD) that I was going to lead on Enterprise Integration. He agreed to be on the organizing committee, and before we left we had developed a plan for who was going to be involved, when the first meeting would take place, and a rough outline of the topics that would be covered over the next year.

I attended CASCON conferences for about 15 years, and established collaborations with individuals that I met through these conferences, often enriching my own work. This was especially true during several Independent Research and Development (IRAD) projects on which I participated during my time at SEI. People that I knew through CASCON served as team members and review members. I co-organized several CASCON workshops that helped to provide feedback and guidance on issues that included service oriented architectures and enterprise integration.

3. MECHANISMS THAT SUPPORT CAS AND CASCON

Recently, I analyzed the mechanisms that make CAS, CASCON and affiliated organizations unique. These mechanisms all support an ingrained attitude of collaboration. Although most are related to CAS and CASCON, some go beyond IBM and relate to the collaboration of the software community throughout Canada. They include: 1) support by IBM management, 2) role of senior academic faculty, 3) government support 4) availability of Toronto Labs facilities 5) affiliated organizations 6) CASCON conferences, and 7) mechanisms for socialization of PhD students. These all support an ingrained attitude of collaboration.

3.1 Support by IBM Management

The concept of developing a Centre for Advanced Study could not have gotten off the ground in an organization as large as IBM without strong support from IBM management. Gene Hoffnagle, who had substantial experience with external collaborations through such activities as his role with IEEE, TCSE, IWCASE and as editor of the IBM Systems Journal developed some of the initial ideas and impetus for CAS. Managers who evolved and supported the CAS concept through the years have included Jacob Slonim, Gabby Silbeman, Kelly Lyons, and Joanna Ng. Steve Perlegut has been involved for many years as Senior Manager for Academic Partnerships. These senior managers have recognized the benefits that accrue to IBM through this collaborative efforts, and they have learned to leverage the inputs of CAS faculty fellows and PhD students. In addition they have maintained a long range point of view, and have resisted natural pressures to focus on immediate and direct impacts of CAS on immediate products.

3.2 Role of Senior Academic Faculty

Some of the initial group of CAS academic research collaborators included Hausi Muller of University of Victoria, François Coallier of Bell Canada, Ric Holt of University of Waterloo, and John Mylopolous of University of Toronto. These researchers brought expertise on reverse engineering, requirements engineering, software engineering standards, and open source and software documentation. This group had a focus on collaboration – the boundaries between sub-fields were spanned in early work, such as Rigi, Software Bookshelf and requirements engineering. This collaboration was solidified through later generations of researchers, such as Jim Cordy, Kostas Kontogiannis, Scott Tilley, Peggy Storey, Kenny Wong, Tim Lethbridge, Eleni Stroulia, Jens Weber, Gail Murphy and Ladan Tahvildari. Each successive generation is socialized into a collaborative mentality; a spirit of excellence is maintained throughout, and it is unusual to see papers that are written by authors from only one institution.

This collaboration has led to numerous papers in refereed international conferences. The leadership of this group of researchers has also been confirmed by their hosting of a number international conferences, such as International Conference on Software Engineering (ICSE) in Toronto (2007), International Conference on Software Maintenance (ICSM) in Victoria (1998 and 2014) and in Edmonton (2009), International Conference on COTS-Based Systems (ICCBSS) in Ottawa (2003) and Banff (2007), and International Conference on Program Comprehension in Toronto (2001), Banff (2007), Vancouver (2009) and Kingston (2011).

3.3 Government Support

Especially in the early years, generous government support through the Natural Sciences and Engineering Research Council (NSERC) for shared software engineering projects was crucial. There was a recognition that something unique had been established and government support through funding, linkages between work of participants, and the development of other support mechanisms encouraged greater collaboration. For these projects, IBM and other corporate sponsors made important in-kind contributions. Of particular note is the tireless work of Anatol Kark, formerly of National Research Council (NRC) of Canada, who early recognized the significance of the model, and was able to effectively serve as a liaison between government, corporations and academic institutions.

3.4 Availability of Toronto Labs Facilities

The resources of IBM Toronto Laboratories offer substantial value for CAS and CASCON. The CAS program for PhD students offers opportunities to interact with world class researchers on real world problems. The faculty research fellows program provides opportunities for faculty members to apply their knowledge to real world situations and to ground their work in practical problems. Both provide value to CAS researchers in terms as they apply ideas to customer problems.

3.5 Affiliated Organizations

Several affiliated organizations were established that contributed to collaboration between the academic participants. These provided additional support for collaboration between academic partners.

I am most familiar with the Consortium for Software Engineering Research (CSER). CSER represents one mechanism for interaction that has roots in the CASCON approach.

This group has two meetings or mini-conferences per year. These meetings typically last for one day and include a keynote and about a dozen technical talks. The CSER meetings provide for close communication between researchers at early stages of development. By including the work of PhD students in the presentations, the next generation of professionals learns how to develop focused research projects and present the results in a compelling manner.

3.6 CASCON Conferences

CASCON has become a significant and established conference. It is unique in that it is sponsored by a corporation, and yet offers a quality technical program with a strong program committee. The opportunities for sharing have been a key enabler for the development of a strong software engineering community across Canada, as well as in other areas of the world.

CASCON has achieved rigor while providing opportunities for academic and corporate research. Several of its features include:

- High standards for the acceptance of papers. My experience on CASCON program committees compares favorably with experiences on other program committees. Each paper is scored independently and then all papers are discussed in significant detail before final decisions are made.
- Workshops that emphasize collaboration. I have personally organized workshops through the years that included organizers from 3 different countries and 8 organizations; this is more impressive when considering countries of origin: Canada, US, Greece, Iran, Colombia, India, Germany, England, Israel, Poland, Russia, China. The focus of workshops has always been on collaboration across institutions, rather than on orthodoxy or adherence to a specific line of inquiry.
- Technology showcase. This has always been a large event - in 2014 there were about 50 research projects. This event provides an opportunity for direct feedback from the research community to researchers who are often beginning their careers. Feedback comes both from people who are primarily focusing on whether the work meets rigorous academic standards, as well as from others who are primarily focused on whether the work is practical.
- Opportunities for collaboration. There is ample opportunity at breaks, in side meetings, at meals and discussions after conference hours.
- Encouragement of opportunities for collaboration with other corporations. Several noteworthy participants have included Sun Microsystems, KLOCwork, Bell Canada, Computer Associates, and health care organizations.

- Emerging technology track. This track provides an opportunity to keep up to date on research trends and topics that are on the horizon, rather than current state of the practice.
- The keynotes have been generally strong and challenging. An especially worthwhile event has been the related set of talks - frontiers of software engineering practice which have enabled an overview of emerging research.

3.7 Mechanisms for Socialization of Graduate Students

Graduate students are strongly encouraged to collaborate with colleagues at other universities. Mechanisms for accomplishing this are:

- Opportunities to establish friendships, collaboration and research partnerships through direct participation in CAS activities. I've been aware of a number of doctoral students who have been residents in CAS, have developed PhD dissertations through CAS related work and who have continued this collaboration outside the CAS setting.
- Many collaborations continue outside CAS in later career activities such as ICSME, ICSE, IWPC, WCRE, IEEE standards groups - as well as workshops such as MESOCA and VISSOFT. These groups all have leaders who have had significant experience with CAS related activities.
- The goal of collaboration across institutions has been met within CASCON. Most papers accepted in the technical program include authors from more than one institution. All workshops include organizers from several institutions. This type of collaboration has strengthened the field of software engineering beyond the initial confines of IBM.

4. RELATED RESEARCH ON COLLABORATION

Several early studies analyzed the lessons learned from this collaborative effort and recognized its positive contribution to computer science research in Canada, to IBM Canada and to government agencies in Canada that have supported collaborative research. Perelgut, et al. [1] pointed out that as of 1997, “CAS has successfully moved concepts into products, matched students with appropriate jobs, demonstrated IBM's commitment to leadership by integrating current research into products available to customers, and supported academic research by doctoral students and faculty from more than 30 universities. More than 150 professors have received CAS funds”.

One appropriate way of analyzing the costs and benefits of CAS and CASCON can be through business models. However, current published business models apply primarily to organizations where value is added to a product along a value chain primarily by employees of an organization. For CAS case, which focuses on vale added by external organizations, and by its own contributions to the broader Canadian software community, most current business models would strongly underestimate the value that is added by faculty and graduate students of affiliate institutions.

Morris [3] and Zott et al [4] review business model literature in terms of the theoretical underpinnings of the model, number of components, whether empirical support is reported, and the nature of the data. Both analyses cite business models as an emerging analytical method that seeks to explain how business value is created. Morris proposes 6 basic components: value proposition, customer, internal processes and competencies, external positioning, economic model and personal/investor factors. Zott recognizes common themes, but also points out the lack of a common conceptual basis for what should be contained in a business model as well as la lack of consensus on basic concepts. While these models offer potential starting points for analysis, they tend to have the assumption that product value is created by employees of an organization.

A relevant starting point for the analysis of CAS and CASCON would be the model of Chesbrough and Rosenbloom [5] which analyzes the success or failure of 6 different spinoffs of Xerox PARC. This model has relevance because it has been derived from multiple experiences of a large technology corporation using both successes and failures. The Chesbrough model has 6 components: identification of market segment, development of value proposition, definition of the value chain, identification of cost and profit potential, description of the value network, and formulation of competitive strategy. The analysis discusses the successes and failures of the six PARC spinoffs. In a number of cases, the success or failure of the spinoff actually depended on whether the spinoff took a path that differs from the direct sales and large scale closed solution of the parent company. This analysis has some relevance for the CAS case; however, it would not easily account for the direct and indirect value that is added by faculty and PhD student associates. It would also not account for the value that the university community gains in the broader software engineering community, and how this value enhances the reputation of IBM.

This problem is addressed in later work by Chesbrough [6] [7] where he contrasts a closed innovation system with an open innovation system. In open business models companies actively exploit outside ideas and move unused internal technologies outside the organization. [6]. Chesbrough points to IBM innovation in licensing IP in the semi-conductor area, in leveraging development costs for Linux, and in donation of patents to the open source community. This model recognizes that in a rapidly changing technological environment, no single organization has a monopoly on innovative and marketable ideas.

One potential way of dealing with the need for accounting for non-employee contribution would be through formalizing some initial ideas developed by Ng [8] who proposes four languages of innovation: patents, publications, prototypes and productization. These languages each represent different facets of innovation: the first two establish thought leadership, while the last two build business leadership. The productization represents the value that is created by product innovation. The distinction between the languages recognizes that CAS depends on input from academic partners who have different measures of success. As a result the CAS enterprise can only be successful if both the academic and business partners can achieve success in their own domain. This idea could be carried out more formally by mapping the thought leadership dimension to existing business models of open innovation.

Open Innovation is precisely what CAS does. This paper provides a selective snapshot of some of the ways in which CAS has used these concepts in its operation by examining some of the basic mechanisms that have been used over the past 25 years.

In any discussion of technological innovation in North American business, there needs to be a concern for not making short term decisions in order to maximize current stock prices. Mintzberg [9] bluntly states that American companies are trading away their future health for short term results. Lolich [10] believes that brands are over managed leading to a focus on current quarterly profits and to the stifling of technological innovation.

5. CONCLUSION

I have provided a selective snapshot of my personal involvement with the CAS and CASCON community, and the basic mechanisms that have been used for the past 25 years.

Because of my position as a member of the academic community, I have focused on the benefits for them. We conclude that open innovation is precisely what CAS does.

Because of the imperative to estimate costs and benefits, we point to the need for the mapping of recent ideas by Ng [8] to open innovation business models [6,7]

6. REFERENCES

[1] Perelgut, S., Silberman, G., Lyons, K.A., Bennett, K., Overview: The Centre for Advanced Studies. IBM Systems Journal, 36, January, 1997, 474-488.

[2] Slonim, J., Bauer, M. A., Larson, P., Schwartz, J., Butler. C., Buss, E. The Centre for Advanced Studies: A Model for Applied Research and Development. IBM Systems Journal, 33, January, 1994, 382-398.

[3] Morris, M., Schindehutte, A.J., The Entrepreneur's Business Model: Toward a Unified Perspective, Journal of Business Research, Volume 58 (6), 200, pp. 726-735.

[4] Zott, C., Amit, R., and Massa, L. The Business Model: Theoretical Roots, Recent Developments, and Future Research, IESE Business School, WP-862, University of Navarra, Barcelona, Spain, 2010.

[5] Chesbrough, H and Rosenbloom, R.S. The Role of the Business Model in Capturing Value from Innovation: Evidence from Xerox Corporation's Technology Spinoff Companies. Industrial and Corporate Change, Volume 11, Number 3,2002, p. 529-555.

[6] Chesbrough, H. W. Why Companies Should Have Open Business Models. MIT Sloan Management Review, 48(2), 2007, pp. 22-28.

[7] Chesbrough, H.W., Vanhaverbeke, W., and West, J. New Frontiers in Open Innovation. Oxford University Press, New York, 2015

[8] Ng., J. The Four Languages of Innovation: A Model of Industrial Research and Development, IBM:CAS Research Lab, Toronto, 2015.

[9] Mintzberg, H. Productivity is Killing American Enterprises, Harvard Business Review, July-August, 2007, Reprint: R0707

[10] Lolich, L. and Mela, C.M., If Brands Are Built Over Years, Why Are They Managed Over Quarters, Harvard Business Review, July-August, 2007, Reprint: F0707G.

CASCON as an Effective Tool for the Development and Evolution of Research Agendas

Grace A. Lewis
Carnegie Mellon Software Engineering Institute
Pittsburgh, PA USA
glewis@sei.cmu.edu

ABSTRACT

The CASCON conference was established in 1990 as an industrial academic conference in computer science and software engineering. The industrial academic aspect of CASCON makes it an ideal venue for developing research agendas and evolving them over time with input from researchers and practitioners. This paper describes my experience with using CASCON as a tool to develop and evolve a research agenda for service-oriented architecture (SOA) that continues to be cited and used today.

1. INTRODUCTION

CASCON, the Conference of the Center for Advanced Studies on Collaborative Research, was established in 1990 by the IBM Center for Advanced Studies (CAS).[1] Its original intent was to present the work that was being done in the CAS. However, it very quickly turned into a forum for exchange of research and ideas of the academic, public and industrial communities, beyond CAS and IBM. Like many conferences, CASCON features workshops as a way to discuss new and emerging ideas and technologies. The advantage of holding workshops at CASCON is the mix of attendees that come from academic, public, and industrial organizations, therefore truly bridging the often large gap between research and practice.

The goal of this paper is to present the experience of using CASCON workshops as a way to develop and evolve research agendas for new and emerging technologies. Section 2 presents the research agenda for service-oriented architecture (SOA). Section 3 shows how the research agenda was evolved over three years using CASCON as a tool. Section 4 talks about the status and impact of the SOA research agenda today and Section 5 concludes the paper.

2. THE SOA RESEARCH AGENDA

In 2006, in Gartner's words, SOA hype had reached a boiling point [1]. The SOA paradigm was having a significant impact on software development; standardization efforts were progressing; tools were becoming available to support service-oriented systems development; and academic and industrial researchers were working on a range of relevant problems to increase and ease SOA adoption. However, these research efforts were evolving in many directions, without a central compass. This was aggravated by the fact that vendors were framing the SOA discourse with their own agendas. With no clear, commonly agreed upon, overarching themes to focus research activity, there was a danger that important research needs would be overlooked, while other efforts would focus on issues of peripheral long-term significance in practice.

In October 2006, at the Carnegie Mellon Software Engineering Institute (SEI), we started an Independent Research and Development (IRAD) one-year project to develop an SOA research agenda. The goal of the project was to assemble an international research group to analyze the current state of the practice and current research initiatives in SOA and propose a long-term consensus research agenda. In addition to Dennis Smith, also from the SEI, team members included Kostas Kontogiannis, at the time at the University of Waterloo; Marin Litoiu, at the time working as a researcher at the CAS; and Stefan Schuster from the European Software Institute (ESI).

The research started with an exhaustive literature review to discover main research topics in SOA from an academic perspective. We also looked at case studies of successful SOA adoption to also get the practitioner perspective. With this information we proposed an ideal service-oriented systems development life cycle that would support the strategic approach to SOA adoption that was shown in case studies and identified areas of SOA research necessary to fill in the gaps. We then created a taxonomy of research areas required to support short-term and long-term strategic SOA adoption, as shown in Figure 1. Under each research area there were specific research topics, as shown in the sample selection of topics in Engineering in Figure 2. Each research topic was documented with rationale to support the need for research, current research efforts, challenges and gaps, and an annotated bibliography. The summary of the results of the one-year project are documented in [2]. It is important to note that the taxonomy in Figure 1 is an evolution of the

[1] https://www-927.ibm.com/ibm/cas/cascon/

taxonomy presented in [2] that largely took place through CASCON workshops.

3. DEVELOPMENT AND EVOLUTION THROUGH CASCON WORKSHOPS

Once the taxonomy was complete we needed validation from both the research and industry communities. The consensus of the team was to first introduce it at CASCON because we knew we would have the right audience.

3.1 CASCON 2007

At CASCON 2007 we conducted a workshop that we called "SOA Research Challenges: A User Perspective." The aspect of the research agenda that we selected for focus and discussion were the two, often misaligned, service orientation user perspectives. The users from a business perspective are the stakeholders that approve the adoption of service orientation and are interested in the value it that it brings to the organization. Concerns of business users are business case, return on investment (ROI), governance, strategy, and alignment between business and service orientation. The users from a technical perspective are the stakeholders that incorporate services into applications/portals/systems (i.e., service consumers). Concerns of technical users are service usability, service testability, quality of service (QoS), service level agreements (SLAs), governance, technology selection, service identification, and instrumentation and monitoring. We invited speakers from IBM/CAS and academia that could represent the two perspectives. A key outcome of this workshop was the importance of mapping between business processes and services, quality assurance, and testing.

3.2 CASCON 2008

At CASCON 2008 we conducted a follow-on workshop titled "Workshop on SOA Research Challenges: Current Progress and Future Challenges." By this point the taxonomy had continued evolving through other workshops, but also interviews with members of the CAS. We were convinced that there were some areas where what was needed was not more basic research, but rather non-vendor surveys, studies and experiments to produce more concrete guidance for SOA adoption, such as SOA governance, business case for SOA adoption, ROI for SOA adoption, and development processes and practices for service-oriented systems development. An outcome of this workshop was the need for more collaborative research between industry and academia to create real practices; exactly the goal of the CAS.

3.3 CASCON 2009

The last workshop that we conducted at CASCON was in 2009 and the title of the workshop was "Workshop on Multi-Organizational SOA Environments." The goal was to discuss the challenges of SOA implementations when they cross organizational boundaries and not all system elements are under the control of a single organization. Some of the challenges identified included security, runtime monitoring and adaptation, multiple consumers and consumer devices, governance, federation, organizational culture, dynamic discovery and composition of services, and semantic description of services. An interesting outcome of this workshop was the realization that when service-oriented systems cross organizational boundaries, the challenges become the union of traditional challenges of distributed systems, challenges of service-orientation, and organizational challenges. Any ideas to address these challenges would have to consider the tradeoffs between centralization and loose-coupling, governance automation, and consideration of the growing trend towards third-party services, cloud computing, and software-as-a-service.

4. SOA RESEARCH AGENDA TODAY

By 2010 SOA was no longer considered an emerging technology, at least according to Gartner's hype cycle for emerging technologies [4][3]. By this point, in addition to CASCON, the research agenda had been presented and discussed at workshops at the International Conference on Software Engineering (ICSE) and the International Conference on Software Maintenance (ICSM). The PESOS (Principles of Engineering Service-Oriented Systems) workshop is still active at ICSE; MESOCA (Maintenance and Evolution of Service-Oriented and Cloud-Based Systems) is now a symposium that co-locates with ICSME (International Conference on Software Maintenance and Evolution). Even though these events no longer focus on the SOA research agenda, the goal of identifying emerging research challenges remains.

We strongly believe that the research agenda accomplished its goal of getting academia and industry to focus on research areas that really mattered for strategic SOA adoption. The workshop summaries that were published after each workshop and other publications on the SOA research agenda are still being cited today [5][6][7][8][9]. We have first-hand and anecdotal evidence of PhD dissertations that were inspired by topics in the research agenda, such as Norha Villegas' PhD thesis from the University of Victoria [10].

5. CONCLUSIONS

The development and evolution of the SOA Research Agenda would not have been possible without the CAS and CASCON. The interaction with members of the CAS, and CAS-funded projects, professors and students, working on real SOA-related problems, enabled us to identify research topics that were relevant in industry and tie them back to cutting-edge research being conducted in universities and research labs. Being able to present and discuss the research agenda at CASCON workshops enabled us to get insights and feedback from attendees from academia, industry, and the public sector.

CAS and CASCON mainly benefit the Canadian computer science and software engineering community and Canada, but have also benefited people like me outside of Canada. I still collaborate with many of the people that I met at CAS and CASCON. Their easy access to CAS resources, in the form of funding, projects, and internships, enables them to remain in touch with reality and practicality. This is truly a model to be followed by many countries to foster collaboration between academic and industrial research that in the end benefits all of us as the results of their research become part of the software products that we use every day.

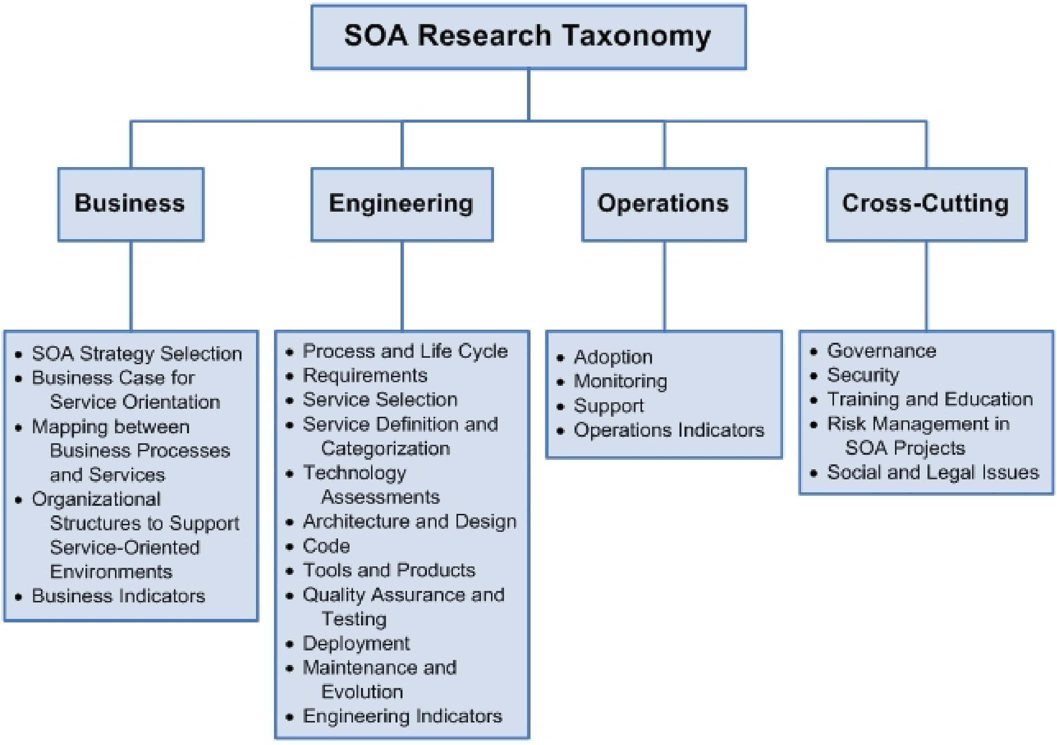


Figure 1: SOA Research Taxonomy

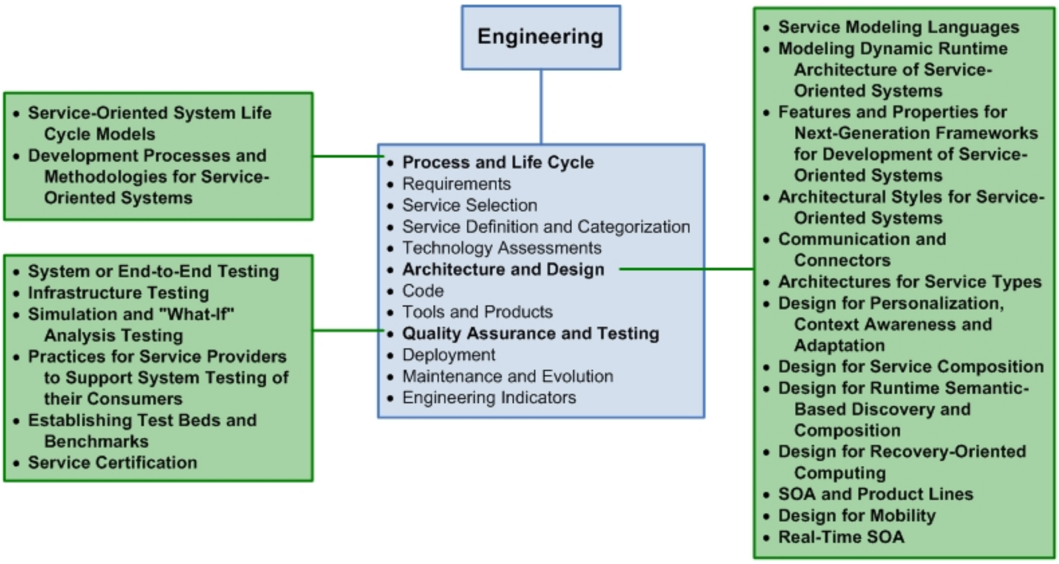

Engineering
• Service-Oriented System Life Cycle Models
• Development Processes and Methodologies for Service-Oriented Systems
• System or End-to-End Testing
• Infrastructure Testing
• Simulation and "What-If" Analysis Testing
• Practices for Service Providers to Support System Testing of their Consumers
• Establishing Test Beds and Benchmarks
• Service Certification
• Process and Life Cycle
• Requirements
• Service Selection
• Service Definition and Categorization
• Technology Assessments
• Architecture and Design
• Code
• Tools and Products
• Quality Assurance and Testing
• Deployment
• Maintenance and Evolution
• Engineering Indicators
• Service Modeling Languages
• Modeling Dynamic Runtime Architecture of Service-Oriented Systems
• Features and Properties for Next-Generation Frameworks for Development of Service-Oriented Systems
• Architectural Styles for Service-Oriented Systems
• Communication and Connectors
• Architectures for Service Types
• Design for Personalization, Context Awareness and Adaptation
• Design for Service Composition
• Design for Runtime Semantic-Based Discovery and Composition
• Design for Recovery-Oriented Computing
• SOA and Product Lines
• Design for Mobility
• Real-Time SOA

Figure 2: Sample of Engineering Research Topics

Acknowledgments

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. This material has been approved for public release and unlimited distribution (DM-0002877).

6. REFERENCES

[1] C. Abrams and J. B. Hill. Achieving agility: SOA will build organizational agility, but watch the hype. Technical report, Gartner, Inc., 2006.

[2] L. Bass, L. Sha, D. Smith, K. Wallnau, J. Hansson, J. Hudak, D. de Niz, P. Feiler, D. Firesmith, M. Klein, K. Kontogiannis, G. Lewis, M. Litoiu, and S. Schuster. Results of SEI independent research and development projects (FY 2007). Technical Report

CMU/SEI-2008-TR-017, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2008.

[3] J. Fenn. Hype cycle for emerging technologies, 2010. Technical report, Gartner, Inc., 2010.

[4] J. Fenn, W. Clark, Y. V. Natis, L. Fiering, M. A. Margevicius, S. Kleynhans, M. Basso, J. Mann, J. Tully, N. Jones, C. Uzureau, A. Newton, S. Ohr, A. Frank, M. McGuire, S. Mingay, D. M. Smith, N. Drakos, W. Andrews, J. Davison, P. O'Donovan, R. J. G. Ball, W. R. Schulte, D. Sholler, S. Morrison, R. F. Mason, A. J. Bradley, J. Edwards, S. Prentice, V. L. Baker, A. Weiner, C. Rozwell, K. Harris, G. Phifer, and P. Basiliere. Hype cycle for emerging technologies, 2009. Technical report, Gartner, Inc., 2009.

[5] M. Kajko-Mattsson, G. A. Lewis, and D. B. Smith. A framework for roles for development, evolution and maintenance of SOA-based systems. In *Proceedings of the International Workshop on Systems Development in SOA Environments*, SDSOA '07. IEEE Computer Society, 2007.

[6] K. Kontogiannis, G. A. Lewis, and D. B. Smith. A research agenda for service-oriented architecture. In *Proceedings of the 2nd International Workshop on Systems Development in SOA Environments*, SDSOA '08, pages 1–6, New York, NY, USA, 2008. ACM.

[7] K. Kontogiannis, G. A. Lewis, D. B. Smith, M. Litoiu, H. Muller, S. Schuster, and E. Stroulia. The landscape of service-oriented systems: A research perspective. In *Proceedings of the International Workshop on Systems Development in SOA Environments*, SDSOA '07. IEEE Computer Society, 2007.

[8] G. Lewis and D. Smith. Service-oriented architecture and its implications for software maintenance and evolution. In *Frontiers of Software Maintenance, 2008. FoSM 2008.*, pages 1–10, Sept 2008.

[9] G. Lewis, D. Smith, and K. Kontogiannis. A research agenda for service-oriented architecture (SOA): Maintenance and evolution of service-oriented systems. Technical Report CMU/SEI-2010-TN-003, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2010.

[10] N. M. Villegas. *Context management and self-adaptivity for situation-aware smart software systems*. PhD thesis, University of Victoria, 2013.