Academic Background Description

I have requested that my transcripts be forwarded directly to PEO at the following address:

Professional Engineers Ontario
101-40 Sheppard Ave. W.,Toronto, ON M2N 6K9
T: 416 224-1100 or 800 339-3716
www.peo.on.ca

B. A. Sc. Software Engineering, University of Ottawa (Sep 1997 – April 2001)

Location: 100 Queen St, Ottawa, ON K1P 5T8
Dean: Dr. Tyseer Aboulnasr

I was the first official graduate of an accredited Software Engineering program in Canada (June 2001 convocation). The current curriculum is available at http://uottawa.ca/academic/info/regist/calendars/programs/1209.html. Courses included a mix of core engineering, ethics, computer science, software modeling, software design and implementation, and business courses.

I also worked as a teaching assistant in both Computer Science and Engineering courses in English and French. Topics include software quality management, test-driven development, file management, C programming and a general introduction to Engineering.

M. Sc. Computer Science, University of Ottawa (Sep 2001 – Sep 2002)

Location: 800 King Edward Ave, Ottawa ON, K1N 6N5 Canada
Supervisor: Dr. Timothy C. Lethbridge

My masters' research focused on the issue of documentation quality. In particular, which attributes of documentation better predict its usefulness to the reader and how this information can be monitored and parameterized to provide a better perspective about the relevance of documentation in a software project.

The thesis was titled Software Documentation: Building and Maintaining Artefacts of Communication and is available for download at http://www.site.uottawa.ca/~tcl/gradtheses/aforward/

A listing of relevant publications is available at https://www.researchgate.net/profile/Andrew_Forward/publications/

Abstract of Masters Thesis (2001 – 2002)

Software documentation is an important aspect of both software projects and software engineering in general. In fact, documentation engineering has become a popular sub-domain in the software engineering community. Unfortunately, the current perception of documentation is that it is outdated, irrelevant and incomplete. For the most part, this perception is probably true. Regrettably, the documentation concern cannot be resolved by simply mandating more and better documentation. This approach fails to resolve the problem as the solution ignores the fundamental goals of software engineering. The role of documentation in a software engineering environment is

to communicate information to its audience and instil knowledge of the system it describes. Documentation should efficiently allow for future software development without hindering current progress.

An engineered solution to the documentation problem would involve allocating appropriate resources to document adequate knowledge about the system to the extent that both current and future development will optimally benefit.

Unfortunately, neither do we fully understand the impact of documentation on current or future development, nor what aspects of documentation contribute to its ability to communicate effectively. We do not really know to what extent we should document in order to balance the trade offs between, on the one hand, allocating too many resources for documentation thus hindering present development; and, on the other hand, not allocating enough resources and thus hindering future development. To complicate matters, increasing documentation resources does not necessarily improve future software development (the ultimate goal of documentation) because we do not really understand what defines documentation quality.

Our research focuses on the issue of documentation quality. In particular, which attributes of documentation make it effective to the audience and how can this information be monitored and parameterized to provide a better perspective about the relevance of documentation in a software project.

Ph. D. Computer Science University of Ottawa (Sep 2006 – Sep 2010)

Location: 800 King Edward Ave, Ottawa ON, K1N 6N5 Canada
Supervisor: Dr. Timothy C. Lethbridge

For my PhD, I worked with the Cruise Lab (Complexity Reduction in Software Engineering) and conducted several online surveys to gather data about the attitudes of software practitioners, and developed a model-oriented programming language (Umple) available at http://try.umple.org.

Within the team, I introduced continuous integration, continuous deployment, automated testing and source code control. I helped open source the Umple project, and built much of the first versions of the compiler / pre-processor. This work included the refactoring of Umple to be written in itself.

The thesis was titled The Convergence of Modeling and Programming: Facilitating the Representation of Attributes and Associations in the Umple Model-Oriented Programming Language. Available for download at http://www.site.uottawa.ca/~tcl/gradtheses/aforwardphd/

Published several papers on the topics of model oriented programming languages, software modeling, code generation, and model-driven prototyping. Full list available at https://www.researchgate.net/profile/Andrew_Forward/publications/

Abstract of PhD Thesis (2006 – 2010)

This thesis investigates approaches to model-driven development (MDD) in which developers can keep using their familiar textual programming languages, but with additional model oriented concepts. The added concepts include associations and attributes as found in the Unified Modeling Language (UML), as well as concepts from software patterns and other common programming abstractions.

By keeping text at the forefront of development, we maintain all of the advantages of text, without having to sacrifice the benefits of diagrams. By allowing a model to be equally expressed in either diagrammatic or textual form, we enable what we have termed text-diagram duality, a duality that benefits programmers and modelers alike.

We explore why software developers in some situations prefer diagrams, but tend to revert to textual means to write and maintain software systems. To explore the capabilities of modeling in code, we developed a model-oriented programming language called Umple. At its core, Umple is a family of object-oriented languages enhanced with additional abstractions. Umple supports both platform-independent models (PIM), as well as platform specific models (PSM). Umple currently integrates with Java, PHP and Ruby; referred to as base languages throughout this thesis. Our research focuses on investigating the opportunities and obstacles we discovered in the course of implementing and using UML-like associations and attributes in Umple.

It is our hypothesis that current features available in object-oriented languages can be enhanced with a more model-oriented approach, providing a textual form for modeling concepts that have been primarily available diagrammatically. By providing modeling abstractions in a programming language, the complexity and size of the resulting systems, we argue, is reduced and more developers, particularly those who are used to writing code, will be more eager to adopt modeling practices. At the same time, our approach maintains the benefits of diagrammatic approaches to software development, since Umple can be rendered and edited as a UML-like diagram.

Our primary contributions to the field of computer science are as follows: First, we provide an empirical investigation on the nature of modeling practices. Second, we present the design, implementation and analysis of a model-oriented language, Umple. The language is presented as enhancements to existing programming languages including Java, PHP and Ruby

Engineering Experience Record

I have 13 years of experience in software engineering as outlined below. Specific details of Application of Theory, Practical Experience, Management of Engineering, Communication Skills and Social Implications of Engineering in these positions are also outlined. The experience is listed in reverse chronological order.

Emp09 - Senior Software Developer, *CENX, Inc. (Oct 2010 - present)*

Location: 202B Main Street, Ottawa, ON K1S1C6, Canada
Supervisor: Steve Rycroft

I worked as a software developer with a Carrier Ethernet interoperability company called CENX. My responsibilities focused on building solutions to new and complex client problems. The results typically led to entire teams being formed based on my initial work to integrate those solutions into core product offerings for other clients.

I built tools to manage 15 million client locations used in our Market application (Sys22). This included the design, implementation and deployment of the applications, as well as creating a benchmarking environment to help determine (and improve) throughput.

I worked as a data scientist and developed systems to intelligently process gigabytes of inconsistent data (from spreadsheets, word and html documents, html, PDFs, email, log files, data structures and other database servers) involving hundreds of different formats and hundred of thousands of source files (Sys19). The work involved integrating an automation process of managing, delivering and analyzing the data (Sys21) with the human process of defining new translation rules as new formats and structures were received. I also helped to scale these systems to processing hundred of thousands of data points on a daily basis. This was achieved through isolated measurement and analysis of sample systems, refining process algorithms and optimizing caching and IO access.

In several instances I was involved in the creation of domain specific languages (DSL) to help document client needs using client terminology. Using a DSL enables the direct execution, visualization and documentation of these system within the same syntax as understood from the domain experts. In particular, I provided a DSL for integrating and interacting with 3rd party circuit ordering APIs (Sys20), for building telecom network topologies (Sys17, Sys18), and auditing customer data (Sys19).

As part of our approach to quality assurance we analyzed our systems both for correctness as well as load capabilities. To manage these testing needs I built a KPI (key performance indicator) generator to simulate network conditions like latency, jitter and frame loss (Sys16). This was used to provide isolated datasets to facilitate correctness testing, to demonstrate functionality without physical network connectivity, as well as generate data for adequate load and performance benchmarking.

In addition to system development, I also introduced, managed and improved several automation and process controls including:

- Automated server monitoring with Monit (Qa07).
- Continuous integration (CI) and delivery (CD) using CruiseControl, Cijoe, PHP (Qa08),
- Behavioural Driven Development (BDD) with concepts like test-first development using tools like Rspec, Autotest, Guard, Selenium and Web Driver (Qa09),

- DevOps (Qa10) processes and systems to automate and enable scalable cloud-based applications.

Emp08 - **Software Consultant** *- 4079914 Canada Inc (since June 2000)*

Location: Ottawa
Supervisor: Self-Employed

I have worked as a software consultant on several large and small companies and projects with duties ranging from design, implementation, testing, measurement, as well as on-going support.

I helped develop a Use Case Editor called UCEd (Sys07); an open source system for building use cases from semi-structured text as well as allow simulation of the interactions between use cases.

I built a travel agency website (Sys08) including a news publishing platform geared for the Ottawa area, and optimized for target search terms.

I was a senior software architect for SiteValet.com (Sys09) a system for building virtual ISPs for small to medium sided hotels, inns and bed-and-breakfasts.

I worked with a company called ArrowMight helping to migrate their distance learning software to a more scalable and robust system (Sys10) as well as created an internal application, called ARM (Sys15), for an easy-to-use relationship manager between ArrowMight employees, support organizations, program delivery facilitators and suppliers.

I worked with HatchMortgage.com (Sys11) as an application developer to streamline the mortgage application process with social media integration, REST API and JQuery UI. I developed and integrated a Mortgage API backend with a Wordpress plugin to present the complexities of home financing in a clear and precise manner.

I worked with Uhber.com (Sys12) as a full-stack developer to help the team build a social media platform to create the *ultimate homepage*. I worked with a small team of designers and product owners on this project. The software development process incorporated an iterative feedback cycle of requirements gathering, feature delivery and feedback. The application included automated deployment, monitoring, testing and infrastructure.

I supported a project called Client-signup.com (Sys13) which helps dentists and doctors go paperless when registering new patients. I worked with a partner to design, development, deploy and test the system. In addition to the application functionality, I was involved in developing systems to automate server configurations / infrastructure, application deployment, testing, and monitoring.

I worked with a team on developing a joint, state-of-the-art grants management system to deliver financial and non-financial benefits, such as improved user experience and risk minimization for The Natural Sciences and Engineering Council of Canada (NSERC) Grants (Sys14). The implementation included business process architecture, business transformation and change management. A large-scale data migration effort was also executed to consolidate data from multiple existing legacy systems.

My role with the NSERC Grants application was to help build a business oriented web service tier, called CRMi, between the front-end grant applications and the back-end XRM (extended relationship management) system built using Microsoft SharePoint and Microsoft CRM.

Emp07 - Ph. D. Computer Science University of Ottawa (Sep 2006 – Aug 2010)

Location: 800 King Edward Ave, Ottawa ON, K1N 6N5 Canada
Supervisor: Dr. Timothy C. Lethbridge

In addition to my research and development of a model-oriented programming language, I also brought my industry experience of software project management to ensure the work being developed in the Cruise Labs was of sufficient quality and depths for use in production systems.

Introduced continuous integration, continuous deployment, automated testing and source code control (Qa06). Helped develop and later open source the Umple project (Sys06), and built much of the first versions of the compiler / pre-processor. This work included the refactoring of Umple to be written in itself. Some of the work in Emp08 used Umple as it's core language.

I am actively involved in the Undergraduate Capstone Open Source Projects (UCSOP) and Open Academy (OA) initiatives volunteering as a mentor to 4th year computer science and software engineering students completing their *project* course. Each semester, a new set of 4-8 students join the Umple open source project contributing bug fixes, features, and updated documentation. Our remote team meets once in person at the start of the semester for a weekend boot-camp introducing the students to Umple. Our team then meets officially once a week thereafter for a status update, but informal communication was fostered through instant messaging, emails and through the development mailing list.

Emp06 - Senior Application Developer, *Library & Archives Canada (Jan 2006 – Dec 2010)*

Location: 550 Place de la Cité, Gatineau, QC, J8T 0A7
Supervisor: Jacob Rafoul

Software application developer building systems to manage the preservation, storage and circulation of Canada's heritage known as Care of Collection CoC (Sys04). I was also involved in the requirements, design and prototyping of a high density storage system HD-CMD (Sys05) to manage rare and low circulation library material (books, audio, magazines, etc). My role was as a senior software developer implementing features, automated testing, manual testing, defect tracking and fixing, and 3rd party component integration via web services, XML and SOAP.

I designed, developed, and tested several modules within Care Of Collections including: Circulation, Physical Management, and Client Services. My main contribution included: Location Maintenance, Container Maintenance, Barcode Print Services, Retrieval Requests and Renewals, and Copy Records. I also created a bridge with the authentication team (CIM) and built a mechanism to tie into the authentication proxy server (iChain) and communication with CIM to control and verify access to the certain aspects of the CoC application (i.e. do you have the authorization to view this data, or perform this particular task).

I also acted as a Liaison between application design, development and the business units including requirements, estimation, development, testing, mentoring and project planning. Guided configuration management team to integrate SVN. Introduced PHPUnit / SimpleTest, Selenium and WebTest and phpUnderControl (Qa05).

I interfaced with the core COC database, and helped design caching mechanisms (both within the application and at the data level), I helped plan, implement and deliver new features on request, resolved defects and worked with related teams to resolve issues outside of my direct responsibilities where possible.

As an application maintainer, I was involved in bug tracking and fixing. I tracked, updated, reported and resolved issues using the Mantis bug tracking system. Using a test-driven approach, most corrections are made by first introducing a failing-test that isolates the bug, and then once passing, that bug is now represented as a regression test to avoid the defect returning to the system.

I was involved in the software documentation process. I created internal software documentation to share design and architecture amongst other software developers. During weekly status meeting, I would present technical progress reports to outline new interfaces, functions and coding practices available to be used by other developers. I also presented workshops on various subjects like Automated Testing (WebTest, SimpleTest, PHPUnit), Web UI testing (Selenium) with accompanying documentation to help others improve on their testing skills.

In summary:

- Involved in full SDLC life cycle of CMS project, and integration with MIKAN for AMICAN project.
- Developed framework and strategy for Application Management which was used by all web application development including CoC
- Involved with the systems analysis team responsible for gathering and analyzing all business and functional requirements from LAC staff. This involved reviewing existing workflow, procedures, databases structures, applications and sub-systems in order to determine the best solution.
- Evaluated existing LAC legacy technologies to determine whether they could fulfill business and functional requirements; analyzed and evaluated alternative technology solutions.
- Transformed functional requirements into technical designs, and software implementation as part of the development team.
- Ensured new technical designs integrated with existing applications and were consistent with industry and GOC technology direction.
- Modified PHP, JavaScript, HTML and Oracle procedure source code required to implement new functionality.
- As Part of the Application Framework developed routines that manage / search and coordinate user privileges for all AMICAN applications.
- Used RCS, and SVN source control commands in telnet/Unix/Linux environment to manage all PHP, HTML and JavaScript source.
- Maintained E/R Data models (documented data structure and developed data dictionary) and worked with DBAs to implement all necessary Oracle 9i/10g database changes.
- Roles also included unit testing, system testing, integration and regression testing of all PHP, HTML and JavaScript code and data enhancements.
- Corrected all errors in PHP, HTML and JavaScript source code as bugs were detected.
- Data migration and source code road-map to move the existing system to the target environment
- Delivered adequate documentation including road maps, requirements documents, use cases, automated test scripts and defect tracking notes

A second major project that I worked on while at the Library and Archives was a High-Density version of the COC application called HD-CMS (Sys05). I was involved with the requirements gathering, use case development, prototyping and application infrastructure of the high density CoC application delivering functionality to manage the storage, preservation, and circulation of the institution's archival and published holdings in a high-density facility.

In particular, I was involved in the following areas of work

- Application architecture of CMS-HD
- Technical feasibility analysis for the buy-versus-build decision
- Involved with requirements gathering and conducting a feasibility analysis for integrating said features into the existing institutions infrastructure

• Developed use cases and prototype material
• Managed the application environment with Subversion including merge and branch creation to manage the code base in parallel with core COC work
• Data migration and source code road-map to move the existing system to the target environment
• Transformational development, deployment and testing
• Delivered adequate documentation including road maps, requirements documents, use cases, automated test scripts and defect tracking notes

The following technologies during these projects: PHP 5.2.1 (OOP), CSS, Bash, RCS, SVN, Fusebox 3 Methodology, Oracle 9i/10g RDBMS, XHTML, JavaScript 1.6, AJAX, CSS 2.0, XML, TOAD, SQL Developer Putty 0.5, RCS 1.1, Mantis 0.18, UNIX (Tru64 v5.1), Linux (SUSE), Apache web server 1.3, Dreamweaver 8, Mozilla Firefox 1.5, 2.0, Internet Explorer 6.0, SOAP, Web Services, and iChain. Search APIs / technologies included SOLR and K2.

Emp05 - Senior Consultant, *Deloitte (Mar 2003 – Dec 2005)*

Location: 100 Queen St, Ottawa, ON K1P 5T8
Supervisor: Arman Mirchandani
Experience: 2 years, 9 months

Worked as a technology integration consultant at Deloitte focusing on public sector projects. My most significant project involved upgrading the Canadian Federal Governments' accountability software with the development of the Program Architecture (and Horizontal Architectures), including planning, budgeting, reporting and review of *The Blue Book* (a fiscal account of the governments spending). The application was built using ASP.Net, C#, JavaScript, SqlServer 2000 (Sys03).

I also helped introduce automated testing with NUnit, NUnitAsp, JsUnit, and Mock objects (Qa01). I managed the continuous integration server using CruiseControl.Net (Qa02), worked with performance and memory profiling using RedGate ANTS (Qa03), and Selenium (Qa04). I also managed several small teams. I was exposed to entire SDLC including requirements, modelling, methodologies, architecture principles, software engineering, testing & validation, and database management.

Emp04 - M. Sc. Computer Science, University of Ottawa (Sep 2001 – Sep 2002)

Location: 800 King Edward Ave, Ottawa ON, K1N 6N5 Canada
Supervisor: Dr. Timothy C. Lethbridge

Developed a documentation plugin to Eclipse (Sys02) to enable direct access to system documentation within the development environment, as well as a tracked usage patterns to predict document usefulness.

Emp03 – Software Tester at Nortel Networks (May 2000 – Aug 2000)

Location: 800 King Edward Ave, Ottawa ON, K1N 6N5 Canada
Supervisor: Shawn McCormick

My role was with the Optical Carrier Team in the testing group to build, deploy and execute the optical carriers in a staged environment to ensure the equipment met both functional and non-functional requirements. In particular, I worked with the OC-3, OC-48 and OC-192 multiplexers. My work involved the configuration of networking hardware under different topologies, as well as configuring and testing the networking software by simulating various types of network traffic.

In additional, I wrote a support web application (Sys01) to present the results of the automated testing and to manage network asset reservation (e.g. IP addresses, OC equipment, etc.). The technologies included HTML, CGI, Perl, HTTP, XML and Apache.

Emp02 – Teach Assistant University of Ottawa (Jan 1999 – Apr 2001)

Location: 800 King Edward Ave, Ottawa ON, K1N 6N5 Canada
Supervisor: Dr. Tyseer Aboulnasr

I was a teaching assistant for several undergraduate courses at the University of Ottawa. Courses included:

- Introduction to Engineering Computation. Emphasis is on the design of algorithms and their implementation for solving engineering problems using C.
- File Management and Data Structures. The concept of abstract data types. Simple methods of complexity analysis. Trees. The search problem: balanced trees, binary-trees, hashing. Sorting. Graphs and simple graph algorithms: traversal, minimum spanning tree. Strings and pattern matching.
- Introduction to Computer Science. Introduction to computer based problem solving for scientific applications. Design of algorithms and algorithms descriptions. 4th generation languages. Software packages. Structured program development. Modular and object-oriented programming. Program testing.
- Advanced Topics in Software Quality. How to assure it and verify it, and the need for a culture of quality. Avoidance of errors and other quality problems. Inspections and reviews. Testing, verification and validation techniques. Process assurance vs. Product assurance. Quality process standards. Product and process assurance. Problem analysis and reporting.

In the teachings listed above, I would deliver classroom instruction, as well as coordinate and lead hands-on lab work. I provided feedback to the supervising professor regarding the student's progress as well as insight to help guide student evaluation.

The technologies used included: Java, JUnit, Unix/Linux, Windows, C, C++, PHP, Pascal, and XML.

Emp01 – Adventures in Engineering and Science (May 1999 – Aug 2000)

Location: 800 King Edward Ave, Ottawa ON, K1N 6N5 Canada
Supervisor: Michelle Bennett

Adventures in Engineering and Science (AES) is an award-winning, not-for-profit, bilingual educational outreach program committed to introducing young minds to the wonders and merits of science and engineering. In addition, AES hosts a summer computer camp where students learn about animation, multimedia-enriched web sites, robotics (LEGO MindStorms, ROBIX), and how to create games and computer hardware.

I held two roles within the AES organization. First, I was a IT instructor specializing in Software Engineering, presenting interactive science and engineering workshops to children aged 5 to 14. I was also involved in developing the computer / IT related projects for the weeklong summer sessions. My second role was as Camp Coordinator, where I lead a staff of 20, and managed registrations for the summer camp, reporting directly to the AES president.

Engineering Categories

Application of Theory

Requirements analysis: I have been involved with JAD (Joint-Application Development) sessions between individual team in Emp05 (Sys03), and Emp06 (Sys04). I have done use case analysis in Sys03, Sys04, Sys05 based on requirements gathering as well as user story in Emp08 (Sys09-Sys15), and Emp09 (Sys16-Sys21).

Model / Object oriented analysis: My PhD work (Emp07) allowed me to become an expert in modeling and Object-Oriented analysis, which we applied to development of Umple (Sys06), a model-oriented programming language. This language was used in the design of Qa08, Sys10, Sys11, Sys15.

Software design: All of the systems developed required careful design including managing the complexity of the overall solution, incorporating version control and upgrade strategies to manage production systems as well as ensure

Software testing and quality assurance: Software quality, measurement and automation have been a core part of projects that I have worked on (Qa01-Qa10), this includes the application of testing theory including White/Grey and Black box testing, System and Integration testing, Smoke Tests, Performance Testing and Monitoring as well as new concepts such as Automated Testing, Test-Driven Development, Continuous Integration, Continuous Deployment and Delivery.

Statistical Analysis and Distribution. The Sys16 project enabled us to generate traffic patterns that were random in nature, yet satisfied the desired distribution including controlling for mean, standard deviation, and outliers with varying sample periods (little t) and analysis periods (big T).

Principles of usability engineering and user-centered design: The Sys15 project was geared towards infrequent use by non-technical individuals so we focused on a minimalistic design. Sys12 and Sys13 incorporated aspects of responsive design to ensure the application was usable on any type of device (phone, tablet, laptop, etc).

Principles of compilers and language design: At Emp09, I built several Domain-Specific-Languages (Sys16-Sys22) to enable system development to be done in a declarative manner by our technical analysts and engineering team (non-programmers). I also applied parsing, language design and compiler technology during my PhD in the Umple (Sys06) model-oriented language.

Database concepts and theory: My work with Library and Archives Canada Emp06 (Sys04, Sys05), I was involved in the design of databases for the management of archival material, as well as the performance and optimization tuning. My work in Sys09, Sys10, and Sys12 are other examples where the database design was an integral part of my role in system development. Emp09 also involved the extensive use of database theory both with SQL backed applications (Sys22), and document based / NoSQL storage (Sys19, Sys20, Sys21).

Practical Experience

I have had practical experience with the engineering of large software systems at the CENX (Emp09), Library and Archives Canada (Emp06), and Deloitte (Emp05). I also have done industrially-based research with IBM during my PhD (Emp07). I have observed similar engineering challenges with respect to project management, software process and the importance of automation and traceability. I have been involved in tradeoffs among reliability, scalability, usability, maintainability and efficient in the use of resources. I have worked in various development environments and methodologies including Waterfall, Iterative, and Agile.

Management of Engineering

I have lead teams, managed employees, and mentored colleagues in most of my employment roles. At Emp05, I lead a team of development working on Sys03. During my PhD (Emp07), I mentored several of master's students and worked closely with other PhD students. As a follow-up to my PhD, I have also acted as a mentor and supervisor overseeing 4th computer science and software engineering students as they complete their fourth-year project working on the Umple open source system (Sys06). At Emp09, I managed two software product groups and oversaw several developers. I have been involved in planning features, scoping and scheduling for product deliveries and managing progress.

Communication Skills

I have published 20 peer reviewed paper, successfully defended a masters and PhD thesis. In my work experience I have developed client-facing presentations, internal development documentation as well as incorporated automation into the process, building specification and change-log documentation from available sources such as source code repositories and in-line code comments. As part of the UCOSP / OA initiative, I have been mentoring up to eight 4th year students each semester as they complete their project course by working on Umple (Sys06).

Social Implications on Software Engineering

In Sys11, we worked to simplify the complicated process of mortgage approvals with an intuitive interface that gathered data in an ever-increasing level of detail such that approval ranges could be provided with a minimal amount of up-front data.

In Sys12, we were building the ultimate homepage using social connections to provide a more personalized experience. But as such, there is a trade-off between incorporating the social relevance for search with the need for individual privacy. To address these issues, we incorporated a straight-forward privacy policy where homepage links were public, friends-only or private.

Application security and privacy is a key concern with regard to password protection, privacy protection and financial / credit-card protection. Several safeguard have been used to protect users by one-way encrypting passwords, using trusted 3rd party payment gateways and 3rd party authentication proxy servers. Finally, using network firewalls helps to protect access to intranet applications where access from outside secure tunnels (i.e. VPN) is not required.

At the Library and Archives, the HD-CMS application (Sys05) represented a shift away from typical library storage based on subject matter to storing material based on physical characteristics. This shift has major implications on mislabeled material being stored incorrectly, and due to the sheer size of the storage facilities would most likely be lost to the public forever. The engineered solution had to limit the impacts of single point-of-failures both within the inconsistencies of human nature as well as hardware / software defects.

System and Engineering Task Summary

Sys01. OC Test / IP Reservation System. 2000. A web application for reserving optical carrier test equipment including hardware equipment, and IP addresses. The application also tracked the results of OC test cases for managing regressions for new versions of the software.

Sys02. Document Aura. 2001. An Eclipse plugin that integrated system documentation into a developer's IDE (integrated development environment). The application incorporated much of the research from my masters' thesis on relevant software documentation.

Sys03. Program Architecture. 2003 - 2005. A suite of inter-related web applications for planning, budgeting, reporting and reviewing of government spending. Application helps produce what's known as *The Blue Book* (a fiscal account of government spending). The application was built using ASP.Net, C#, JavaScript, SQLServer 2000.

Sys04. Care of Collections (COC). 2006-2009. The COC application is an enterprise-level Intranet application dealing with multi-million record databases to provide a single point of access to the Library and Archives' collection. The system interacts with and manages information within library and archival environments as well as helping to manage the circulation, and physical management of cultural information. The COC system was intended to hold 5 million records of enterprise-critical content and at the time of my work on the project held over 2 million rows in a very complex structure relational database with complex relationships and large variable size text fields.

The COC application has a high number of integrated and dependent functions with multiple main modules of varying size and attributes. These modules consolidate the physical management, intellectual management and access to LAC's holdings by delivering functions of circulation and maintenance; conservation treatment; digitization, imaging, microfilming; and audio-visual preservation.

The system integrates with existing and still/developed LAC components including a complex archival holdings management system and repository (MIKAN) to integrate with very complex physical management system (CoC) and very complex contact information management system (CIM).

MIKAN is an integrated Library archival management system and is also cultural information management system with over 250 users from the NCR and Canada-wide regional offices. This system is the primary corporation wide mission critical archival system used by the National Archives.

The COC application leveraged metadata and full-text search technologies, Enterprise search technologies, and Enterprise Content Management Web Services Portal technologies, as well as Client Relationship Management Electronic commerce REST (Representational State Transfer) OpenURL. The applications support Networking protocols including Z39.50 and Dublin Core. The system stores descriptive metadata about items in LAC's vast archival collection and is searched by over 250 users.

There is a web-based interface for the entire system. The system was developed using HTML, JavaScript, CSS stylesheets, Internet Explorer 6, DOM, DHTML, Apache Web Server, Fusebox, PHP, C and Oracle 9i/10g (RDBMS), including Oracle Text for keyword searching running in a HP Tru64 UNIX Clustered server environment using Storage Area Network (SAN) technology and LINUX (SUSE) environment.

Sys05. COC Management System High Density (HD-CMS). 2009 - 2010. The CMS-HD application extends the existing COC application framework to support the circulation and physical management of cultural information in a high density facility. The key requirements and major engineering challenges for a HD facility is loss avoidance. The HD-CMS application was designed with fall-back mechanism to promote information cross-checking without an overly complicated design. In addition, a separate mechanism was designed to manage the high-traffic (high probability of loss) activity of moving books and artifacts into the HD facility versus the day-to-day lower-traffic process of circulating those artifacts (by definition the artifacts destinted for these facilities were rare and rarely circulated material).

Sys06. Umple. 2007-current. A modeling tool and programming language family to enable what we call Model-Oriented Programming. It adds abstractions such as Associations, Attributes and State Machines derived from the UML to object-oriented programming languages such as Java, C++, PHP and Ruby. Umple can also be used to create UML class and state diagrams textually. The Umple tooling has been written in itself.

Sys07. The Use Case Editor (UCEd). 2001. A software design and prototype system that integrates a set of tools for use cases editing, domain model editing and requirements simulation. UCEd is an open source system for building a use case simulator using Java, SWT, and XML. The editor allowed for the creation of use cases from semi-structured prose, and along with a domain model enabled use case simulation. Based on use cases and a domain model description, UCEd generates a state model that realizes the use cases. The state model is then used as a basis to generate a prototype with a graphical user interface to simulate the use cases.

Sys08. TravelOnlyOttawa.ca. 2004 - 2007. A travel agency website and blog focusing on travel in the Ottawa area. It provided a back-end editor for non-technical individuals to manage content, write articles and maintain the site. The engineering challenge for this project was to need to enable fast and efficient editing of content with the importance of fast and efficient page loading. Our optimization included HTML/CSS code generation from the back-end PHP system, with a focus on minimizing page-sizes and optimized static pages from the web server.

Sys09. SiteValet.com. 2008 - current. A Software-As-A-Service (SaaS) application for hotels, motels and inns. The application allowed owners to manage their hotel website through a portal geared specifically towards the accommodations' industry and tracked a variety of amenities such as physical characteristics, local attractions, room rates, etc. The engineering challenge was to build a data model that sufficiently met the needs of various types of properties without being overly complex for non-technical users and allow for customizations so that not all *websites* looked the same. To address these challenges, we built a template engine from which designers could build themes from which users to choose. These templates included both stylistic components (placement of elements like navigation, colours, images, etc) as well structural (inclusion of certain type of back end data).

The engineering challenges included dealing with managing a large number of raw high quality photo images. Large photos take longer to upload, require more resources to process and greater bandwidth to present to our end-users. The application had to cope under these conditions in near real-time performance using techniques ranging from high performance image processors, adaptive image cropping, asynchronous / threaded worker queues, as well as using user interface techniques to improve the perception of real-time.

I was involved in evaluating and ultimate implementing support for various enterprise cloud computing solutions at the time including Amazon S3, Amazon EC2, Amazon RDS, Slicehost / Rackspace, and Heroku Ruby Cloud Platform.

The project involved the following technologies: Ruby (1.8/1.9), Ruby on Rails (2.3.4), GIT, XHTML, JavaScript 1.6, AJAX, CSS 2.0, XML, MySQL, LighthouseApp, Nginx, Passenger, Mongrel, Max OS X, Linux (Ubtuntu 8.04), FireFox, Internet Explorer (6,7,8), Ruby Unit, Amazon S3, Rackspace, Slicehost, DNS, crontab.

Sys10. Learning Management System (LMS). 2009 - current. The ArrowMight education program is an at-home adult learning literacy program and computer course for English speaking Canadians, Aboriginal and Immigrant learners. The LMS tracks and records the progress of all students to successful program completion and provides a) Financial accountability for government, private sector and Aboriginal funders b) Easy independent monitoring and reporting for education providers including colleges and community organizations, c) Computerized end-of-program results quickly reducing administration and evaluation costs. The engineering challenge was to build a data model that allowed the flexibility to alter the program as it evolved, with proper program versioning to ensure accurate historical analytics. The project also included a data transformation piece to migrate the legacy prototype to the new system, including near-time synchronization as both systems ran in parallel.

Sys11. HatchMortgage.com. 2013. A web application to make streamline the mortgage broker process using transparency of data, qualification criteria and cost. The engineering challenge was present the complicated topic of loan approval metrics to the masses. This was achieved by focusing first on gathering macro-level data, but allowing for further refinement in a context-aware manner (i.e. asking the right questions at the right time).

Sys12. Uhber.com. 2012. Social media platform to help build the *ultimate homepage*. The engineering challenge was to build a product that responsively adapted for both mobile and desktop/laptop user interfaces, that provided sufficient benefit to an individual to warrant traction as well as offer enhanced *crowd* capabilities to foster social growth.

Sys13. Client-signup.com. 2013. A backend server and database and a tablet application targeted at helping dentists and doctors go paperless when registering new patients. The engineering challenge was to integrate the registration process with new paperless technologies with the already established back-end (and offline) systems. The solution also needed a low infrastructure footprint to ensure we were not adding *yet another system*. This was accomplished by supporting both on-site deployments as well as cloud based SAAS (Software-As-A-Solution) for a zero-footprint install feature.

Sys 14. NSERC Grants / CRMi. 2012. Grants application that involved functionality around the main stages of Grants Management: Arrange, Advertise, Apply, Assess, Award, Admin, Audit and Acquit. The interchange between CRM and web portal CRMi was designed to include a testing capability for service interoperability without the presence of the third party, allowing both CRM and the web portal be developed and tested in isolation. The engineering challenge was to build a proxy between client applications and the raw XRM data. The API included several key features including: a web front end which included documentation about the API itself, the ability to query the data directly for debugging the fault isolation, between the XRM backend, the API service itself and the client applications. Finally, the API was built with sample/stub capabilities to enable client applications to test interactions with the API without the need for a connection to a live XRM system. This proxy enabled faster and better parallel development. Clients of the NSERC API (i.e. the grant web application and administration portal) could access sample data for features not yet fully available from XRM, to enable localized testing of those client application. The XRM development team could also access their functionality in the same manner as client applications but in a more direct manner, reducing the impact of client application bugs on XRM testing.

The engineering challenges of CRMi included

- Uncertain whether the project teams could integrate dynamic and rapidly changing applications. The application included two separate parallel developments of a portal user interface (available to the public) and a content / relationship management system on the back-end (CRM). Correctness and accuracy could not be proven for the portal, as our development and insight/available information was bounded by the CRM handshake with the portal.

- Uncertain how to develop integration code for an unknown API and uncertain logical requirements. Imperative programming defines how to (logically) achieve the intended results. The project team could not determine ***the how*** because the API handshake was unknown and a moving target for both the CRM and portal.

- Uncertain whether an application firewall could continuously resolve API divergences within the performance targets of the project. The application firewall had to be generic to various possible web portal designs and platforms (MS SharePoint-based, dynamic .Net-based, etc.) and, additionally, due to security constraints, the CRM could not be queried directly, which introduced performance uncertainties.

To deal with the uncertainties above, we developed a language translator capable of bridging the gap between technology agnostic applications (such as the web portal) and a proprietary CRM. Incorporating an enhanced approach to software quality, we developeda Multi-Level Test Driven Development (MLTDD) methodology which when applied enabled for improved stabilization of API divergences across a multitude of applications. In addition, we were able to shield both applications from changes to one another further promoting parallel development. This framework supported development environments where façade API could be quickly conceived and integrated with production environment where the APIs were locked and cached using code generation to satisfy performance and load requirements.

Sys15. ARM (Arrowmight Relationship Management). 2010 - 2011. ArrowMight had a need for a simple and focused CRM (Customer Relationship Management) system to deal with the various spreadsheets, including online resources like GoogleDocs GoogleSpreadsheets. The engineering challenge was to balance the very specific current needs of the system with their future needs. We used Umple (Sys05) to build our model with a dynamic back-end data model combined with code generation to quickly support changing CRM needs.

Sys16. Kpigen. 2010-2011. A spreadsheet-based language for building sample data that was shaped to certain statistical distributions. The application supported ranges, averages, spikes and standard-deviation curves. The engineering challenge that kpigen addressed was to provide the ability to generate large volumes of data, but with known patterns to enable efficient testing of our network alarms, faults and graphing software. It also enabled us to demonstrate key features of our analytics software without network connectivity.

Sys17. OSL. 2010-2012. The Offered-Services-Study (OSS) specific language. This system is a previous generation to Sys17. This language is tied to an OSS database which provides an inventory for telecom inventory systems. The language provides an OSS specific dialog for describing circuits and can populate inventory systems by ingesting a Plan Of Execution (POE) specification. The engineering challenge is to provide a language specific for telecom domain engineers to support new vendors, new topologies (e.g. microwave, point-to-point, dual, etc) and operators.

Sys18: CM-Build. 2012-current. An application for taking aggregated and cleansed data about a network topology (from Sys2 for example) and organizing into a structured manner for populated a network inventory database. The engineering challenge here is to trade-off between building generic recipes to populate all network topologies with the flexibility to support client-specific needs.

Sys19. CM-Audit. 2011-current. A data aggregation and warehousing system that can be configured to collect data from various sources including SQL based data stores, comma separated files, spreadsheet files, WORD, PDF, XLS files as well as plain-text / config files. The system builds on top of lessons learned in Sys1 (which focused solely on geographic based information) into a generic ingesting, consolidation, cross-referencing system. The engineering challenge is to build a process of automatically categorizing and ingesting information with no external information about what has been provided. The system needs to provide dynamic inferences between data sources and content such that a schema can emerge and evolve.

Sys20. Webit. 2010-2011. A language geared at web application automation when official APIs are not available. The engineering challenge was to build a robust language for automating interactions with a web application (that does not offer an official machine interface). The language needed to be sufficient detailed to support multiple applications, as well as be readily understood by domain experts. Webit integrated directly with browses like Chrome, IE and Firefox and was used to automate the ordering of circuits within Emp06.

Sys21. Grid. 2010-2013. A data retrieval and delivery mechanism for moving files between servers and notifying interested parties of new content. The engineering challenge was to support various network transfer protocols including SSH, FTP, SFTP, SCP, and IMAP to collect, transform and deliver source content to one of more targets. The abstraction needed to focus on the needs of data transfer as well as provide sufficient protocol details (e.g. port, host, mode, keys, passwords).

Sys22. Market and Geotidy. 2010-2012. A web-based system (Market) for helping telecommunications marketplace for the inventory, selling and buying of Ethernet services (ESL). Key engineering challenges: Handling large volumes and semi-structured data for consistent inventory of ESL data, applying heuristics to normalize the automatically cleanse the data as well as provide the ability to easily manage fallout requiring human intervention. The system needed to easily adapt to new data sources, as well as easily integrate new heuristic transformations. The system also needed to be efficient and scale to millions of records. Finally, searching geographic proximities needed to be supported (i.e. find results within 500 meters, 1km, etc) and efficient

Qa01. NunitWeb. 2003-2005. Developed an extension to Nunit to enabled UI testing from the command line (browserless). This was achieved by simulating the web environment including managing cookies, session and user requests and enabled hundreds of UI tests to be part of the continuous integration process.

Qa02. CruiseControl.net. 2003-2005. Managed a *build* server dedicated to continuously integrating four separate projects in Sys02 to check for integration issues. By integrating early and often we were able to easily identify both expected and unexpected changes to our application interfaces (API).

Qa03. RedGate ANTS. 2003-2005. Performed system and memory profiling in Sys02. The ANTS product enabled us to: profile our applications for memory consumption issues, test our system under simulated loads for performance bottlenecks as well as allowed us to compare database schemas between environment to highlight discre

Qa04. Selenium. 2003-2005. Introduced automated testing with NUnit, NUnitAsp, JsUnit, and Mock objects (Qa01). Managed the continuous integration server using CruiseControl.Net (Qa02). Worked with performance and memory profiling using RedGate ANTS (Qa03). and Selenium (Qa04). Managed several small teams. Exposed to entire SDLC including requirements, modelling, methodologies, architecture principles, software engineering, testing & validation, and database management.

Qa05. PHPUnderControl. 2006-2010. Continuous Integration server (PHPUnderControl) combined with unit testing using PHPUnit and SimpleTest, as well as user acceptance testing / UI automation using Selenium and WebTest. PHPUnit is a command line testing tool used to verify the smallest components of a system. Selenium is a User Acceptance Testing framework design to allow record-and-playback style testing that acts as a regression testing tool as the user level. WebUnit was a custom-built testing framework that enabled verification of the application within its deployed environment (similar to other unit testing tools like PHPUnit but accessible through a browser as opposed to a command line).

Qa06. CruiseControl. 2006-current. A continuous integration server (CruiseControl) that managed the building and automated testing of several open source projects for my work with the Cruise Labs (Emp06). The systems integrated unit testing from JUnit, PHPUnit, RubyUnit, project compiling using ANT and a continuous delivery system to the online editor (try.umple.org) using Bash. A primary dashboard manages the build and its state (passing or failing), as well as a second dashboard with more detailed analysis of the automated test results.

Qa07. Monit. 2010-current. Manage a cluster of servers that includes a process, and monitoring tool (Monit) at Emp06. The scripts to manage monitoring are automated through a infrastructure language and tool called Chef. The engineering opportunity is to provide server specific monitoring and notifications to proactively access issues like high memory consumption, aborted services

Qa08. Continuous Delivery. 2010-current. Managed a *build server at Emp06* that is a combination of tools including the build-launcher (originally CruiseControl, and later Cijoe), a builder (writing in rake), automated testing and code coverage analysis (using tools like rspec and rcoverage) and a dashboard (custom application written in PHP that summarizes data from the components above).

Qa09. Behavioural Driven Development (BDD). 2010-current. Automated much of the testing infrastructure at Emp06. This includes unit testing using Rspec, Autotest and Guard as well as automated user acceptance testing (UAT) using web browser tools like Selenium and Web Driver. The experience gained with automating UAT tests was used to build Sys5.

Qa10. DevOps. 2010-current. There is a relatively new movement in software engineering to treat infrastructure and operations much like you would any software system known as DevOps. By *developing* your operations you gain much better control and consistency in your environments as the system administration becomes yet another system that is version controlled, automated and consistently deployed. I helped developed many *devops* scripts (using Chef) and manage the roll-out to multiple servers and environments at Emp06.

Published Papers on Software Engineering

The following is a list of my refereed publications, and other major published works. This list can be found at: http://a4word.com/publications.php.

Conference Publications

Omar Badreddin, Timothy C. Lethbridge, Andrew Forward, Maged Elaasar, Hamoud Aljamaan, Miguel A. Garzon. "Enhanced Code Generation from UML Composite State Machines". Modelsward 201

Omar Badreddin, Andrew Forward, Timothy C. Lethbridge. "A Test-Driven Approach for Developing Software Languages". Modelsward 2014.

Omar Badreddin, Timothy C. Lethbridge, Andrew Forward. "Investigation And Evaluation Of UML Action Languages". Modelsward 2014.

Omar Badreddin, Timothy C. Lethbridge, Andrew Forward. "A Novel Approach to Versioning and Merging Model and Code Uniformly". Modelsward 2014.

Hamoud Aljamaan, Timothy C. Lethbridge, Andrew Forward. "Specifying Trace Directives for UML Attributes and State Machines". Modelsward 2014.

Omar Bahy Badreddin, Andrew Forward, Timothy C. Lethbridge. "Exploring a Model-Oriented and Executable Syntax for UML Attributes". SERA (selected papers) 2013: 33-53

Omar Bahy Badreddin, Andrew Forward, Timothy C. Lethbridge. "Improving Code Generation for Associations: Enforcing Multiplicity Constraints and Ensuring Referential Integrity". SERA 2013: 129-149

Omar Badreddin, Andrew Forward and Timothy C. Lethbridge. “Model Oriented Programming: An Empirical Study of Comprehension”. CASCON 2012

Lethbridge, T., Mussbacher, G, Forward, A. and Badreddin, O, (2011) "Teaching UML Using Umple: Applying Model-Oriented Programming in the Classroom", CSEE&T 2011, pp. 421-428.

Omar Bahy Badreddin, Timothy Lethbridge, Hisham El-Shishiny, Margaret-Anne D. Storey, Andrew Forward: Challenges and opportunities in applying research prototypes and findings into industrial practice. CASCON 2010: 414-415

Lethbridge, T.C., Forward, A. and Badreddin, O. (2010), "Umplification: Refactoring to Incrementally Add Abstraction to a Program", Working Conference on Reverse Engineering, Boston, October 2010, pp. 220-224.

Forward, A., Badreddin, O., and Lethbridge T.C. (2010), "Perceptions of Software Modeling: A Survey of Software Practitioners", 5th Workshop From code centric to model centric: Evaluating the effectiveness of MDD (C2M:EEMDD), Paris, June 2010, http://www.esi.es/modelplex/c2m/papers.php.

Forward, A., Badreddin, O., and Lethbridge T.C. (2010), "Umple: Towards Combining Model Driven with Prototype Driven System Development", 21st IEEE International Symposium on Rapid System Prototyping, Fairfax VA, June.

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.

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.179-181.

Forward, A., and Lethbridge, T.C. (2008), "Problems and Opportunities for Model-Centric Versus Code-Centric Software Development: A Survey of Software Professionals", Workshop on Modeling in Software Engineering, in conjunction with ICSE 2008, Leipzig, ACM, pp. 27-32.

Forward, A., Lethbridge, T.C. and Deugo, D (2007), "CodeSnippets Plug-in to Eclipse: Introducing Web 2.0 Tagging to Improve Software Developer Recall", Software Engineering Research, Management and Applications (SERA) 2007, August, IEEE Computer Society, pp. 451-460.

Forward, A. and Lethbridge, T.C. (2002), "The Relevance of Software Documentation, Tools and Technologies: A Survey", DocEng 2002: The ACM Conference on Documentation Engineering, pp 26-33.

Journal Publications

Forward, A., Lethbridge, T.C., Badreddin, O., Solan, J., (2011) "Model-driven rapid prototyping with Umple", IEEE Software Practice and Experience. DOI: 10.1002/spe.1155

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.