ABSTRACT

Joint research between National University of Singapore (NUS) and ST Electronics Pte Ltd (STEE) started through a broad collaboration agreement, seven years ago. Collaboration was intensified by a research project that also involved two other partners, namely University of Waterloo and Netron Inc. This new project led to development of a reuse technology called XVCL. The continuity of collaboration was sustained through a project-driven approach, that evolved around development and application of XVCL. Collaboration helped STEE to advance reuse practice via application of XVCL in several software product line projects. Early feedback from industrial applications along with many other inputs from STEE helped NUS team validate and refine XVCL reuse methods, explore new research directions, and ensured that academic research results remained in sync with industrial realities. In the paper, we describe our modes of collaboration, results, and factors that were critical to sustain collaboration and benefits gained over years on both sides.

Categories and Subject Descriptors
D.2.2 [SOFTWARE ENGINEERING]: Design Tools and Techniques; D.2.10 [SOFTWARE ENGINEERING]: Design – Representations; D.2.13 [SOFTWARE ENGINEERING]: Reusable Software - Domain engineering;

General Terms
Design, Languages, Experimentation

Keywords
reuse, software product lines, university-industry collaboration

1. HISTORY OF COLLABORATION

ST Electronics Pte Ltd (STEE) is a Singapore-based company developing turn-key software solutions in a wide range of domains, for local and international markets including defense and home-land security applications. In 1998, STEE initiated a programme to develop a Common Application Platform (CAP), with the objective of providing fast and cost effective customized solutions in the Command and Control domain. CAP was built to form a foundation of reusable components designed to serve as low-level reuse libraries as well as higher-level services designed to facilitate implementation of design patterns. Even though the reuse solutions developed for CAP have been used in many projects across STEE, and the programme has been considered as a big success, certain weaknesses of component-based reuse have also been exposed, and the approach has barely been able to keep up with the customer evolving expectations of shorter time to market and more cost effective solutions.

Software engineering research team at School of Computing, National University of Singapore (NUS) has been involved in research on product line approach to reuse [4] since 1995. Early experiments with Frame Technology from Netron, Inc. [3] led to the definition and implementation of XVCL reuse method and tool [11]. Further development of XVCL and its applications in industrial projects and lab studies have become the main driver for STEE-NUS collaboration that has intensified over time, and brought benefits in terms of both academic research results and industrial projects.

STEE and NUS collaboration journey started in 1998 when a Memorandum Of Understanding (MOU) was established between the two institutions. The first few projects were focused on reliable component platforms in Mission Critical Command and Control Systems, addressing some of the problems in expediting component-based reuse identified by STEE’s CAP programme. These first projects delivered concrete and useful benefits to both sides, helping STEE in technology selection, while at the same time helping NUS to bring industry-related projects to their students. These positive experiences acted as a booster for the collaborative spirit and brought the two sides closer together looking for more and bigger exploration in the area of product lines.

The governments of Singapore and Ontario, Canada, have established a joint research programme to boost collaboration among universities, and involving industrial partners from both countries. In 1999, we submitted project proposal to investigate methods for cost-effective, reuse-based development of reliable mission-critical software systems. The proposed project involved four partners, namely NUS, STEE, Netron Inc. (Toronto), and University of Waterloo. NUS provided software engineering and reuse expertise. University of Waterloo contributed in areas of software reliability – failure detection, fault tolerance and availability. STEE has been developing command and control mission-critical systems for customers in Singapore (such as Ministry of Defense, police and civil defense) as well as abroad, and had extensive experience in mission-critical system domain. STEE was also a potential client for the technology we intended to develop in this project. Our Canadian industrial partner, Netron, Inc. contributed to our project with reuse tools and their rich
experience in implementing reuse solutions in companies [3]. Our proposal was among five projects selected for funding, out of some 30 proposals submitted.

Before formulating the project proposal, we had already had an established working relationship among project partners, though not all four of them together and not in the exact scope of the proposed project. NUS and University of Waterloo had discussed research on the interplay between reuse and reliability. As mentioned earlier, the link between STEE and NUS was already established through the MOU and close collaboration was already in place. Finally, the NUS team worked with Frame Technology developed by Netron, Inc. before the joint Singapore-Ontario project, and had accumulated experiences in that area and described them in publications. Those earlier links between partners helped us agree on common research goals.

As our project progressed, the following three focus areas emerged:

- Definition of the XVCL language.
- XVCL-based pilot project [9].

Definition of the XVCL language was facilitated primarily through collaboration between NUS and Netron. Netron shared their experience with frame technology [3] both through visits to Singapore and through short attachment of NUS students at their Toronto office. These visits and feedback provided by Netron (and in particular by Paul Basset) was essential to the successful definition of a simple yet practical XVCL language.

Implementation of the XVCL processor was done by students at NUS. As we chose XML as a vehicle for defining and then implementing XVCL, we could benefit from use of open-source components. In 2002, the resulting processor was also made public at SourceForget [11].

Experimentation with XVCL-based product lines was done through a pilot project for Computer Aided Dispatch (CAD) in the domain of Police and Fire emergency dispatch [13]. This pilot project started with definition of use cases and study of feature variants both within a police system but also across other areas of the civil defense domain. Use cases were provided by STEE to establish an understanding of requirements, while NUS students explored how XVCL could be applied to handle feature variants across the product line. Implementation was done jointly by NUS students and XVCL team. The pilot project demonstrated that the XVCL approach was very capable of handling variants in CAD product lines, and the result formed an incubator for new experimentation and application of the XVCL technique.

Currently, we leverage on the XVCL technique to explore product lines and reuse in various domains, and we also venture into other research areas inspired by results we were getting on the way.

2. Modes of Collaboration

Initially, collaboration was facilitated through an Industrial Attachment, where Honours and Master students were attached to STEE for research projects related to Reuse Frameworks. The progress of work and intermediate results were discussed during meetings. Issues arising in those discussions allowed NUS to better understand real-world software development, and problems that were on top of the STEE’s agenda. On the other hand, STEE got understanding of research directions at NUS and assessed their relevance to STEE’s goals.

The Singapore-Ontario joint project was critical for realizing the potential of the synergies that apparently existed between our institutions. The very process of formulating and writing the proposal allowed project partners get closer to each other, and understand each other needs and constraints. Many useful ideas and targets emerged at that time. During the Singapore-Ontario project, we learned quite a bit about how to (and how not to) do joint research involving geographically distant research teams including both industrial and academic parties.

We worked with software requirements abstracted from real-world projects, contributed by the STEE. In Singapore, overall control and evolution was facilitated through weekly (or bi-weekly) research meetings with participation from both NUS and STEE. These meetings served as a communication channel where:

- Result, ideas and findings were shared.
- Feedback was provided.
- Future work was brainstormed and outlined.

Initial meetings focused on clarifying requirements and documenting them in a standard way. Subsequent meetings concentrated on discussing novel approaches to modeling “requirements with variants” [8], as it was needed for reuse via product line approach, and on novel techniques for designing generic software architectures, capable of handling variant requirements in an effective and simple way. These empirical studies were instrumental in gaining insights into laws governing the design of “flexible software”, i.e., software that is easy to change and adapt to fit various reuse contexts. We tested the limits of what could be achieved to this end by conventional architecture-centric, OO and component-based programming techniques, and with this understanding it became possible to observe the value of meta-level enhancements that were later implemented into the XVCL method.

An important aspect of these meetings was to facilitate more communication between research teams working on different focus areas[9]. These meetings also served as a vehicle for sharing of experiences and findings well before publications were written, resulting in faster and more agile direction changes. We believe it helped us a lot to accelerate and effectively shape our research.

These meetings played an increasingly important coordination role, as over time more and more parallel and incremental projects branched out from the main stream of our work.

Meetings have also helped us to strengthen the partnership between NUS and STEE resulting in continued collaboration beyond the joint Singapore-Ontario project. Through these meetings, the collaboration entered the stage where new projects were initiated, executed and shared without any formal agreement between the parties (apart from the general MOU).

An important facilitator of the Singapore-Ontario project were mutual visits between Canadian and Singaporean partners. In particular, our collaborator from Netron (Paul Basset, a co-founder of Netron and its Director of Research) visited NUS-STEE team in Singapore twice. The objective of the first visit (one week) was to focus and scope problems that were to be solved by means of the reuse technology. During the second visit (10 days), we conducted detailed reviews of technical solutions, identified the need for refinement of the reuse technology which set the ground for development of XVCL, and formulated plans for the rest of our project. At the same time, our Canadian collaborator conducted a full-day public seminar on reuse technology that was also attended by all the researchers (students, faculty and company staff) involved in the project. Towards the end of the project,
when work on formulating XVCL intensified, two NUS graduate students spent one month at Netron, Inc.

Project collaborator from the University of Waterloo visited NUS team twice to work on reliability requirements addressed in our project. NUS and University of Waterloo faculty spent part of their sabbatical working with overseas partners. Three students from the University of Waterloo spent two weeks at NUS, and two students from NUS spent one month at the University of Waterloo.

3. SIGNIFICANCE OF RESULTS

We focused our first bigger project on internet-enabled Computer Aided Dispatch Systems (CAD for short). Figure 1 depicts a basic operational scenario, roles and elements of a CAD system for Police.

![Figure 1. An operational scenario in CAD system for Police](image)

At the basic operational level, all CAD systems are similar – basically, they support the dispatch of units to incidents. However, there are also differences across CAD systems. The specific context of the operation (such as police or fire & rescue) results in many variations on the basic operational scheme. If we ignore commonalities, then each CAD system in a specific context becomes a unique application that must be developed from scratch and maintained as a separate product – an expensive and inefficient solution. In our project, we planned to apply a product line approach [4] that would allow us to exploit commonalities among CAD systems, and engineer CAD systems from a common base of reusable software assets, so-called product line architecture. We expected such a reuse-based approach to radically cut development and maintenance cost. The pilot project confirmed those expectations, and also led to formulating and testing in practice novel methods to realize such benefits. Partial results of this first pilot project have been described in a number of conference papers and a summary of methods developed in this project - in [13].

In the first pilot project, we observed certain limitations of modern but conventional programming methods that we used (programming languages, design techniques, component platforms and architectural approaches) to achieve levels of genericity that were required to define effective reuse solutions. In particular, we observed that many components of CAD systems were very similar to each other, but still it was not possible for us to represent them in a simple generic form. We saw this a major bottle-neck for effective reuse. In studies that followed the CAD project, we studied this “weak generics” phenomenon in a wide range of programming techniques and application domains. In our studies of new, well-designed programs, we typically found 50%-90% of code contained in such similar program structures, repeated many times (often called clones in the literature). For example, the extent of the redundant code in Java Buffer library was 68% [6], in parts of STL (C++) - over 50% [1], in J2EE Web Portals – 61% [12], and in certain ASP Web portal modules – up to 90% [9]. The last two results are from the projects at STEE in which XVCL was applied to combat the problem of counter-productive repetitions. Most of the repetitions that we found were counter-productive for maintenance and signified untapped reuse opportunities.

In those studies, we realized that software similarities, especially large granularity, design-level similarity patterns, created opportunities for reuse within a given system, or even across similar systems. Unfortunately, at times, conventional methods – component based, architecture-centric approaches as well as language-level features such as generics – failed to provide effective means to reap benefits offered by software similarities.

Common sense suggests that developers should be able to express their design and code without unwanted repetitions, whenever they wished to do so. We defined XVCL technique to tackle exactly this problem, with minimum impact on main-stream programming techniques commonly used today. With XVCL extensions to OO and component-based techniques, we could effectively unify similarities found in above mentioned projects. We got evidences from lab studies and industrial applications of XVCL that this kind of reuse can further reduce development and maintenance effort [9][12][13], on top of productivity gains due to skilled use of architectural approaches and other today’s most advanced development technologies.

In the recent ASP Web Portal (WP) Product Line project, STEE applied state-of-the-art design methods to maximize reusability of a Team Collaboration Portal (TCP) in other contexts. Still, a number of problem areas were observed that could be improved by applying XVCL to increase the genericity of a conventional solution. The benefits of a “mixed strategy” ASP/XVCL solution for TCP were the following:

- Short time (less than 2 weeks) and small effort (2 persons) to transform the TCP into the first version of a “mixed strategy” ASP/XVCL solution.
- High productivity in building new portals from the ASP/XVCL solution. Based on the ASP/XVCL solution, ST Electronics could build new portal modules by writing as little as 10% of unique custom code, while the rest of code could be reused. This code reduction translated into an estimated eight-fold reduction of effort required to build new portals.
- Significant reduction of maintenance effort when enhancing individual portals. The overall managed code lines for nine portals were 22% less than the original single portal.
- Wide range of portals differing in a large number of inter-dependent features supported by the ASP/XVCL solution. The reader may find full details of this project in [9]. We followed up with projects using more advanced Web technologies such as PHP, J2EE and .NET, obtaining similar results [12].

4. BENEFITS AND LESSONS LEARNED

Seven year long industry-university collaboration led to development of a reuse technology and its pilot application in industry. Collaboration allowed researchers at NUS to validate, and evolve their ideas to be in sync with industrial needs and practices. Early applications of XVCL in industrial projects provided invaluable quantitative and qualitative information to clearly see areas of improvements and move research in new, unexpected directions. Through the NUS research collaboration, STEE has gained a better understanding of the practical reuse
problems and limitations. This helps in assessing risk and outlining roadmaps for conventional reuse as well as encouraging exploration into new dimensions of reuse through “mixed strategy”.

Inputs from STEE were very helpful in refining NUS curriculum with new Software Engineering Project course [5]. The course teaches students how to apply design principles and “best practices” in a team-based software development situation.

An international research project under the Singapore-Ontario joint research initiative played an important role in shaping collaboration and accelerated the process of forming mutual understanding of goals, ways to work together and share the results. Further joint work, already without any formal agreements or structures, was very much driven by benefits both sides were getting from the sustained collaboration.

We make very little long-term plans for our collaborative work. We let the past results and the current need drive the selection of projects we embark on. Such a relaxed and open attitude towards collaboration requires much trust. In return, we can always focus on our strengths, and not miss the best opportunities that current situation has to offer. For example, STEE applied XVCL to support a Web Portal Product Line under unexpected business pressure, without much prior planning. The results were so good that NUS initiated a number of research studies in the Web domain that revealed unique opportunities for our techniques in this new domain [10][12]. In a short time, we advanced our understanding of the interplay and synergy between advanced Web technologies and our technique, and learned what it took to turn these findings into further improvements of our approach.

Adopting a new technique always brings some overheads and XVCL is no different. It is essential to understand and evaluate trade-offs involved. Our current and future work focuses on empirical studies in various application domains and interpretation of the results, comparative studies of XVCL and other similar methods, tool support for applying XVCL, and methodological aspects of the “mixed strategy” approach involving synergistic application of conventional techniques and XVCL.

Engineering processes play an important role in industrial software development. Currently, we know how XVCL-enabled “mixed strategy” solutions can raise productivity of small teams of highly-skilled expert software developers. We have yet to learn what it takes to inject “mixed strategy” methods into more complex team structures and industrial development processes. Working on those issues is an important direction for our future university-industry work.

5. ACKNOWLEDGMENTS

The following STEE staff participated in various phases of our collaborative project: Tan Beng Suan, Chan Chin Wei and Soe Myat Swe. The following NUS students participated in the project: Damith C. Rajapakse (PhD student, contributed section on PHP experiments), Hamid A. Basit (PhD student, developed Clone Miner), Zhang Hongyu (Research Scientist under joint Singapore-Ontario Programme and PhD student), Zhang Weishan (Postdoctoral Fellow), Soe Myat Swe, Shen Ru, Peng Chuaning, Chen Liang, (Master students), Pradeepika Irangani, Lam Vu Tung, Cao Yang, Vitali Maximov, Pavel Korschunov (Research Assistants). Paul Bassett’s advice was invaluable in formalizing XVCL and understanding its applications.

REFERENCES