A Proposal for a follow-up 1-Year
ABB Research Project with

Dr. Jonathan I. Maletic
Dr. Michael L. Collard
Department of Computer Science
Kent State University

Executive Summary

Problem Statement
ABB software systems frequently require adaptive maintenance changes due to updates in integrated 3rd party hardware and software, especially prevalent in embedded systems nearing their end of life. These changes include hardware component obsolescence, platform changes, compiler changes, and OS and other 3rd party API changes. These changes usually involve the developer determining which lines of code must be changed and then repetitively apply this change to all parts of one or more code bases. This type of change is error-prone, as it is manual and highly repetitive. In addition to adaptive maintenance changes, refactoring changes to the code base are increasingly becoming important to software developers. These changes are usually done to improve non-functional characteristics of the software, such as performance, maintainability, reliability, etc. Both performance and maintainability are of key importance to the systems ABB creates, as they have a very long deployment and lifecycle.

Proposed Solution
Recent research [1, 2, 4-6] at Kent State University by Jonathan Maletic and Michael Collard, has shown great potential in addressing the problem of frequent adaptive maintenance changes. This research culminated in the creation of a tool and methodology called srcML[1]. SrcML converts C/C++ and Java source code into an XML compliant format with no loss of information or style that can also be converted back to the original source again. Once the source code is in this XML format, XML style sheet translations (XSLT) can be applied, allowing an automated transformation[2] to the code base. XSLT is easy to create and allow developers to think about code and code transformations at a higher level of abstraction than they do today. In addition, the change is applied to all instances in the code exactly the same in a short amount of time.

The first year of this project proved the concept and tool on real ABB code bases. Examples from two different divisions were used and a set of transformations were generated. This continuation project has a few principal goals. The first involves continuing research on making the generation of transformations easier. The second involves researching new applications of the technique, such as refactoring and performance improvement (multi-core transformations). These are two areas that ABB, and other industrial companies, are spending more money on. Finally, research on how to validate the transformations is needed. A differencing tool, or other graphical representation of the original and transformed code would make the validation of the transformations, and the scripts themselves, much easier.
Scope

The scope of this project is a 1 year continuation of the 2008-2009 research project on ABB code systems, specifically targeting improved use and explorations of new applications of the technique. In more detail, the project will research ways to automate the XSLT generation and srcML differencing. The validation of this research will include measures of both efficiency and effectiveness. Efficiency will be shown by determining the time benefit of the tool, when compared to the original time needed to manually transform the software due to the required adaptive change. Effectiveness will involve showing that the tool operates on real industrial source code from ABB, and allows automatic transformation for previously done manual adaptive maintenance activities. Another focus involves researching new types of transformations, such as refactoring for maintenance and performance improvement. These patterns need to be validated on ABB systems and the benefit calculated.

ABB Collaborators
Brian Robinson, Ph.D., Senior Principal Scientist, ABB Principal Investigator

KSU Collaborators
Jonathan I. Maletic, Ph.D., Associate Professor, KSU, University Principal Investigator
Michael L. Collard, Ph.D., Research Associate, KSU, University Co-Principal Investigator

Proposed Main Tasks for the Continuation Project
Below is a list of the proposed tasks for this project. All tasks will have a designated lead organization who is responsible for the overall work item, but all tasks are planned to be collaborative.

1. Determine common types of changes that ABB systems have or need. (Lead: ABB)
2. Research and evaluate srcML’s ability to make transformations of that type. (Lead: KSU)
3. Research ways to better create transformations. (Lead: KSU)
4. Evaluate the tools on past and current adaptive changes in ABB, comparing the results to the manually created changes done before (Lead: ABB)
5. Create final benefit report on ABB software (Lead: ABB/KSU)
6. Create plan and proposal for next phase of collaboration (Lead: ABB/KSU)
7. Present results and tool to ABB Management, determine next steps (Lead: ABB).

Collaborative Benefits
- A meaningful collaboration between ABB and KSU
- Support current research into automated program transformation, validating techniques on industrial systems
- Research ways to automate patterns for transformation
  - KSU: Research support and real systems to work with.
b. ABB: Reduced cost of software maintenance
   • Collaborative publications and measured benefit

Software Tools and Work Products
The tools developed by Maletic/Collard, including the srcML translator and associated infrastructure, are under a GPL (open source license). The intent of their research is to construct open source tools for researcher and industry practitioners. Any new tools developed by the University Collaborators during this research will also be under a GPL. This will allow ABB to use this newly developed software internally at no cost, while still allowing the University Collaborators to meet their research goals.

References (available at www.cs.kent.edu/~jmaletic/)