

# A Framework and Tool for Porting Assessment and Remediation

Donald P. Pazel, Pradeep Varma, Amit Paradkar, Beth Tibbitts,  
Ashok Anand, Philippe Charles

IBM Research

[pazel@us.ibm.com](mailto:pazel@us.ibm.com), [pvarma@in.ibm.com](mailto:pvarma@in.ibm.com), [paradkar@us.ibm.com](mailto:paradkar@us.ibm.com), [tibbitts@us.ibm.com](mailto:tibbitts@us.ibm.com),  
[ashokanand@in.ibm.com](mailto:ashokanand@in.ibm.com), [pcharles@us.ibm.com](mailto:pcharles@us.ibm.com)

## Abstract

*We present a framework and tool for porting assessment and code remediation in commercial porting projects, primarily targeting inter-platform C/C++ porting. Our framework has a modular architecture for handling real-world code bases, a key part of which is an extensible knowledge base of porting issue types. Porting issue detection and remediation are achieved through dynamically-linked detector and remediation “plug-ins” which can be augmented over time. Analysis and remediation can be performed over multiple intermediate forms, such as the AST or error logs. Analysis and remediation results can be presented as a single assessment report, or the changes can be incorporated into the original code base. Furthermore our framework has been integrated into Eclipse [1], providing a user interface and allowing further capability for integration with other Eclipse extensions. A subset of our tool has been prototyped and benchmarked against real-world customer code.*

We describe a tool for inter-platform C/C++ porting whose key components include a C/C++ parser from the Edison Group [2], plug-ins for analysis and remediation, and an Eclipse-based [1] user interface. The parser produces an AST representation for the program. The analysis and remediation plug-ins are Java classes (or “rules”) that work on the AST, and produce the detected porting issues in an encapsulated form. Eclipse provides an IDE as presented to the end user and a tool platform with extensibility to interoperate with an open community of tool enhancements. For example, our porting tool operates with the CDT (C/C++ Development Tool for Eclipse) such that the built-in C/C++ editing, inspection, build, and debugging tools can be used to augment our tool’s capabilities.

The current components build on EDG’s AST and error log representation for porting issue analysis and remediation. A tool component manages porting issues and their relationships with each other, including issues across multiple files. The tool could easily be extended for testing, including code coverage and debugging.

The prototype has been tested against several customer code bases, each covering many include files and thousands of lines, expressions and statements. Exemplary types of porting issues detected include implicit cast issues, variable scoping issues, and API-analysis relating to migration from Sun/Solaris C to Linux/GCC. Hundreds of porting issues with interdependencies have been detected during benchmarks. Each detected porting issue has a suggested remediation. In some cases, semi-automated remediation is offered, with optional user interaction.

Extensibility and modularity are novel design points of our tool. Additionally, our semi-automatic remediation approach distinguishes us from formal, fully-automatic approaches such as Stratego and CodeBoost [3]. Other work enumerates C compiler differences [4], application-specific (OpenVMS) porting [5], and suggesting portable coding practices [6].

## References

- [1] <http://eclipse.org>
- [2] <http://www.edg.com>
- [3] <http://stratego-language.org>
- [4] N. Iwata., M. Hanazava, “Porting Classes and Coding Rules for C”, in Proc. IEEE TRON’93, pp. 144-155, 1993.
- [5] N. B. Kronenberg, W. M. Cardonza, and R. Jagannathan, “Porting OpenVMS from VAX to Alpha AXP”, CACM, Vol. 36, No. 2, pp 45-53, Feb 1993.
- [6] T. Toshikio, “Approaches to Making Porting More Productive”, In Proc. IEEE TRON’95, pp. 73-85.