Fact Extraction and Code Auditing with Columbus and SourceAudit

Rudolf Ferenc, Árpád Beszédes and Tibor Gyimóthy
University of Szeged, Department of Software Engineering
{ferenc|beszedes|gyimi}@inf.u-szeged.hu

Abstract. *Automatic fact extraction from software systems is the fundamental building block in the process of understanding the relationships among a system's elements. We demonstrate the reverse engineering framework called Columbus which is able to automatically extract facts from C++ source code and how the extracted facts can be used in practice. We also mention a special-purpose tool that was developed on top of the Columbus framework. This tool, called SourceAudit, is a code auditor that is able to investigate source code and check it against rules that describe the preferred properties of the code.*

To comprehend a software system we need to know many different things about it. We refer to this information as *facts* about the source code. *Fact extraction* is an automatized process during which the subject system is analyzed with analyzer tools to identify the source code's various characteristics and their interrelationships, and to create abstract representations of the extracted information. The form of these representations is prescribed by *schemas*, which are descriptions of the form of the data in terms of a set of entities with attributes and relationships. A *schema instance* is an embodiment of the schema which models a concrete software system. To make the results of fact extraction widely usable, we further process the schema instances to take various new formats.

Columbus [2] is a reverse engineering framework developed in cooperation between the University of Szeged, the Nokia Research Center and FrontEndART [3]. The main motivation behind developing this framework was to create a toolset which supports fact extraction and provides a common interface for reverse engineering tasks in general. The graphical user interface of the framework is called *Columbus REE* (Reverse Engineering Environment). Further tools are also incorporated into the framework (mostly command line), which actually do the C++-specific tasks, like analyzing the source code and further processing the results.

The extraction process within the Columbus framework is outlined in [1]. The process is very similar to the traditional compilation process. It consists of five consecutive steps where each step uses the results of the previous one. These steps may be performed in two different ways: using Columbus REE, or using the command-line tools with our compiler wrapping technology mentioned in [1]. The steps of the process are the following:

1. Acquiring project/configuration information
2. Analysis of the source – creation of schema instances
3. Linking of schema instances
4. Filtering the schema instances
5. Processing the schema instances

Because different re- and reverse engineering tools use different schemas for representing their data, the schema instances must be further processed to achieve tool interoperability. The processing may consist of transforming the schema instance into another format and/or applying further computations on it. Currently the following are included in the Columbus REE:

- *PPML* – Preprocessor Markup Language and *CPPML* – C++ Markup Language. XML documents that have structures of the Columbus schemas.
- *GXL* with the structures of the Columbus schemas.
- *UML XMI*. Class diagram of the analyzed project.
- *Famix XMI*. *CodeCrawler* document.
- *RSF*. *rigi* document.
- *HTML*. Hypertext documentation.
- *Metrics*. Object oriented metrics.
- *Design Patterns*. Recognition of design patterns.

We developed a special code auditing tool based on Columbus – called *SourceAudit* – which is able to investigate source code and check it against rules that describe the preferred properties of the code. These rules mostly involve issues related to coding style, but in some cases they extend the warning reporting capabilities of the compiler. The checked rules are organized into rule packages and the tool can be freely extended with new packages. The tool can be used in command line and integrated with popular IDE-s (e. g. Microsoft Visual Studio and Borland C++Builder).

References

1. R. Ferenc, Á. Beszédes, and T. Gyimóthy. Extracting Facts with Columbus from C++ Code. In *Tool Demonstrations of CSMR 2004*, pages 4–8. Mar. 2004.
2. R. Ferenc, Á. Beszédes, M. Tarkiainen, and T. Gyimóthy. Columbus – Reverse Engineering Tool and Schema for C++. In *Proceedings of ICSM 2002*, pages 172–181. Oct. 2002.
3. Homepage of FrontEndART Software Ltd. http://www.frontendart.com

Proceedings of the 20th IEEE International Conference on Software Maintenance (ICSM'04)
1063-6773/04 $20.00 © 2004 **IEEE**