Bachelor's Thesis

Automatic Generation of a .NET interface for a C++-library

Problem Statement

In large-scale software projects, sometimes it becomes necessary to use more than one programming language. To make sure the different components work together, care needs to be taken to develop proper interfaces that account for the differences in the programming language. For instance, when a project in a managed language has to use a library written in an unmanaged language, it is necessary to account for proper allocation and deletion of objects to prevent segmentation faults and memory leaks.

While this task is important, it is also very mechanistic and tedious to do by hand, as most parts of such a wrapper are the same for each wrapped class. To automate this process, tools like SWIG [1] have been developed which can generate glue code to bind the different languages together.

While these tools reduce the workload for the developer, they do not eliminate it. Typically, it still is necessary to provide some sort of interface description which the tool then uses to generate the glue code. Furthermore, they can only generate "generic" glue code. If a project uses a certain paradigm that has to be taken into account when generating the interfaces, a lot of work still has to be done by hand in order to make the resulting code work properly.

Task

ARCADE, a program analysis tool for industrial control software developed at i11, has a C++ implementation in form of a library with a command line interface. However, for interfacing with a GUI and potentially CoDeSys, both written in C#, there is a project of glue code written in C++/CLI to allow library functions to be called from C#. The aim of this thesis is to provide a code generator that parses the ARCADE source code, finds functions and classes annotated as interfaces and generates the glue code for these components automatically. Whether this is easier to be done by using a tool such as SWIG or developing the tool completely from scratch will have to be determined during the early part of this thesis.

Qualifications

A good grasp of the C++ programming language is required in order to understand the source code that is to be parsed and generated. However, the implementation of the thesis itself does not have to be done in C++.

Advisor

Marcus Völker, M. Sc. RWTH
voelker@embedded.rwth-aachen.de

[1] http://www.swig.org/