

C nach Eiffel: Automatische Übersetzung und objektorientierte Umstrukturierung von Legacy Quelltext

Marco Trudel

Lehrstuhl für Softwareentwicklung
ETH Zürich
Schweiz
marco.trudel@inf.ethz.ch

Abstract: Ist es möglich einen Teil der riesigen in C entwickelten Code-Basis wiederzuverwenden um die Vorteile moderner Programmiersprachen wie Typsicherheit, Objektorientierung und Verträge nutzen zu können? Dieser Beitrag präsentiert eine Quelltext-zu-Quelltext Übersetzung und objektorientierte Umstrukturierung von C Code nach Eiffel, eine moderne objektorientierte Programmiersprache, und das zugehörige Tool C2Eif. Die Migration ist komplett automatisch und unterstützt die gesamte C Sprache wie sie in der Praxis verwendet wird. Die erstellten Eiffel Programme verfügen über die Eigenschaften guter objektorientierter Programme, wie beispielsweise lose Kopplung und starke Bindung von Klassen sowie geeignete Datenkapselung. Zudem setzen die Programme auch fortschrittliche Merkmale wie Vererbung, Verträge und Ausnahmen ein, um ein klareres Design zu erreichen. Unsere Experimente zeigen dass C2Eif C Anwendungen und Bibliotheken von signifikanter Grösse (wie z.B. `vim` und `libgsl`), sowie anspruchsvolle Benchmarks wie die GCC Torture-Tests, bearbeiten kann. Der erzeugte Eiffel Quelltext ist funktional äquivalent zu dem ursprünglichen C Quelltext und nutzt einige Eigenschaften von Eiffel um sichere und einfach zu korrigierende Programme zu erstellen.

1 Automatische Übersetzung

Wann immer möglich übersetzt C2Eif C Sprachkonstrukte in äquivalente Eiffel Konstrukte. Das ist der Fall für Funktionen, Variablen, Anweisungen, Ausdrücke, Schleifen und Grundtypen mit ihren Rechenoperationen. In Fällen, in denen in Eiffel keine entsprechenden Konstrukte verfügbar sind, bietet das Tool Simulationen der C Konstrukte und Unterstützung durch Bibliotheken an. So werden beispielsweise Sprungbefehle (`break`, `continue`, `return` und `goto`) durch strukturierten Kontrollfluss und Hilfsvariablen simuliert. Für Pointer wird Bibliotheks-Unterstützung angeboten: sie werden zu Instanzen einer generischen Bibliotheks-Klasse `CE_POINTER [G]` übersetzt, welche die volle C Pointer Funktionalität unterstützt. C Structs werden in Eiffel Klassen übersetzt, die von einer Bibliotheks-Klasse `CE_CLASS` erben. Durch die Verwendung von Reflexion ermöglicht diese Klasse ihre Instanz in ein Objekt zu konvertieren, welches das genaue Speicherlayout des C Structs hat. Dies kann für Pointerarithmetik und Interoperabilität mit vorkompilierten C-Bibliotheken notwendig sein.

2 Objektorientierte Umstrukturierung

Die objektorientierte Umstrukturierung extrahiert Elemente mit gutem Design, die in hochwertigem C-Code vorhanden sind, und drückt sie durch objektorientierte Konstrukte und Konzepte aus. Die Umstrukturierung in C2Eif besteht aus vier Schritten, die sich solche implizite Designelemente zu Nutzen macht: (1) *Quelldatei*-Analyse erstellt Klassen und füllt sie basierend auf den C Quelldateien, (2) *Methodensignatur*-Analyse verschiebt Methoden in Klassen auf dessen Daten sie arbeiten, (3) *Aufrufs-Graph*-Analyse verschiebt Felder und Methoden in Klassen in denen sie exklusiv verwendet werden, (4) *Vererbungs*-Analyse erstellt Vererbungsbeziehungen zwischen Klassen anhand ihrer Felder. Neben diesen Kernelementen des objektorientierten Designs werden auch Verträge und Ausnahmen erstellt, basierend auf GNU C Compiler (GCC) Annotationen, Anforderungen an die Argumente von Funktionen sowie Verwendungen der `set jmp` Bibliothek.

3 Fazit

Eine Übersicht [Tru13] über verwandte Arbeiten im Bereich der objektorientierten Übersetzung zeigt, dass bisherige Ansätze Einschränkungen in Bezug auf Vollständigkeit, Automatisierung, Anwendbarkeit und Qualität der Übersetzung haben. Im Gegensatz präsentiert unser Beitrag einen Ansatz welcher sich durch folgende Merkmale auszeichnet:

- Die Übersetzung ist *komplett automatisch* und im frei verfügbaren Tool C2Eif implementiert. Benutzer müssen nur ein C Projekt angeben und C2Eif erstellt ein objektorientiertes Eiffel Programm, das kompiliert und ausgeführt werden kann.
- C2Eif unterstützt die gesamte C-Sprache wie sie in der Praxis verwendet wird, einschliesslich Pointerarithmetik, die Benutzung von nativen System-Bibliotheken (zB für I/O), Inline-Assembler-Code und uneingeschränkte Sprungbefehle.
- Eine umfangreiche Auswertung mit realer Software von beträchtlicher Grösse zeigt dass unsere Übersetzung objektorientierten Code mit hoher Kapselung erzeugt und Vererbung, Verträge, und Ausnahmen angemessen einsetzt.
- Die Übersetzung führt keine potentiell inkorrekten Transformationen durch und erzeugt somit Programme die funktional äquivalent mit den C Programmen sind.

Ausführliche Informationen können in den Veröffentlichungen [TFNM13, TFN⁺12], der Dissertation [Tru13] und auf der Projektwebseite [C2E] gefunden werden.

Danksagung: Die hier präsentierten Forschungsergebnisse stammen aus der Zusammenarbeit mit Carlo A. Furia und Martin Nordio.

Literatur

- [C2E] C2Eif. Der C nach Eiffel Übersetzer. <http://se.inf.ethz.ch/research/c2eif/>.
- [TFN⁺12] Marco Trudel, Carlo A. Furia, Martin Nordio, Bertrand Meyer und Manuel Oriol. C to O-O Translation: Beyond the Easy Stuff. In *WCRE*, 2012.
- [TFNM13] Marco Trudel, Carlo A. Furia, Martin Nordio und Bertrand Meyer. Really Automatic Scalable Object-Oriented Reengineering. In *ECOOP*, 2013.
- [Tru13] Marco Trudel. *Automatic Translation and Object-Oriented Reengineering of Legacy Code*. Dissertation, ETH Zurich, 2013.