1 Introduction

The Software Improvement Group (SIG) offers specialized services that helps companies gaining insight in their legacy code bases. To this end, the internal design structure of a code base is captured by the SIG in a so called source model. A source model can be used, for example, to estimate the amount of work needed for future modifications or to report on various software metrics of the code base under consideration.

To extract source models from a code base, the SIG uses existing and develops new generic source model extraction frameworks. Upon these frameworks specific applications can be built, like the SIG software document generation tool DocGen [5], applications for code quality assessments and tools for code transformations [8].

1.1 The ASF+SDF Meta-Environment

The ASF+SDF Meta-Environment\(^1\) [3] is a generic language development, analysis and transformation framework developed by CWI (Centrum voor Wiskunde en Informatica) and used by the SIG.

SDF (Syntax Definition Formalism) allows for natural, context free and modular description of a language grammar. ASF (Algebraic Specification Formalism) is used to combine SDF rules with rewrite actions and thus can be used to interpret, analyse and transform instantiations of the language described by SDF.

1.2 Island Grammars

For many projects it is desired only to concentrate on specific parts of the complete code base. It is therefore not desired to have a complete language grammar definition in SDF. In other cases it might not even be possible to define a complete and correct language grammar since the original grammar specifications are inaccessible or lost.

A solution to this problem is the use of island parsing techniques, which only requires knowledge of the language grammar in parts that are relevant for the current project. Island Grammars [7] formalize such a parsing technique and can be easily expressed in SDF.

1.3 PGen and SGLR

Currently the transition from an Island Grammar (IG) defined in SDF to a so called island parser that can extract source models, is based on two parts of the MetaEnvironment: PGen and SGLR. Their relationship is as follows: PGen generates a parse table from a syntax definition in SDF that is used by SGLR (Scannerless Generalized LR Parser) to form a parser for that syntax definition. More information on the generation of parsers from island grammars using PGen and SGLR can be found in [6].
2 Need for the project

Practical applications of this approach at the SIG showed that the use of PGen/SGLR to generate island parsers can be problematic in certain situations. Some of the encountered problems are:

- Unstable behavior when extending or combining island grammars, probably due to problems in ambiguity filtering.
- Some generated parsers take a long time to execute and/or demand a large memory capacity.
- When an island parser does not function properly, it is not always clear which IG constructs cause the erroneous behavior.
- (less important for this thesis project) The current approach requires the standalone tool SGLR (developed in C) to be called from Java. A 100% Java solution is preferred for ease of deployment.

The above mentioned problems make the development of solutions based on Island Grammars a tedious job. Therefore the SIG wants to explore alternative approaches to the generation of parsers from Island Grammars.

There are numerous parser generators currently available that could replace the current implementation using PGen and SGLR. The question is how to translate the required island grammar behaviour to these tools. Secondly, there are competing techniques to IG (for example iterative lexical analysis [4]) that may prove to be easier to use and yet provide the same advantages as IGs. Finally, it might be interesting to look at possibilities to combine Island Grammars and these competing techniques and investigate if that could solve some of the aforementioned problems.

3 Project goal

Formulated in a single sentence the graduate project can be defined as: 

"(Alternative) Implementation Strategies for Island Parsers".

In this context the term island parser refers to the tool that is derived from an IG (defined in SDF or some other grammar description format) and performs the translation between an input text and its corresponding syntax tree. Ideally, these new island parsers should transparently replace the old ones in the framework that was discussed in section 1.

At the University of Delft the graduation year is divided into two parts; a research assignment which has a study load of 15 European credit points (10 weeks or 420 hours of work) and a master thesis project which has a study load of 45 European credit points (32 weeks of work). It’s possible to let the research assignment be a well defined preparatory part of the thesis project. Sections 3.1 and 3.2 globally summarizes the requirements for the research assignment and the master thesis project.

3.1 The Research Assignment

As was mentioned before the research assignment will be used as a preparatory study for the master thesis, this will be done in the form of a literature study covering (at least) the following topics:

- Short covering of IG theory
- Overview and comparison of fuzzy parsing techniques (Island Grammars, Iterative Lexical Analysis, other techniques)
- The IG to Island parser translation process
- Review of current Island parser implementations
- Available parser generators and their implementation fitness for Island grammars

The research assignment period is closed with a report and an oral presentation of the research at the research colloquia for staff and students. In general, the report should contain the following items:

- A description of and the motive for the research subject
- Scientific and societal importance
- The planning and staging of the research
- Methodological foundation of the research
- Description and comparison of different research approaches
- Evaluation of research approaches
- Description of carried out analyses
• Results and conclusions

For more information on the research assignment, see [2].

3.2 The Master thesis project

The Master thesis project is defined in [1] as:

The master thesis project is an activity with strong ‘engineering’ aspects such as analysis, design and implementation of algorithms, systems, methods and tools.

The outcome of the Master thesis project is a report of the work and the results of the thesis project. The report must also cover the points listed at the end of section 3.1, but now with respect to the thesis project.

4 Relevance for the faculty

The thesis project is supported by the Software Evolution and Research Laboratory2 (SWERL) of the University of Delft. The SWERL focuses its research mainly in the area of program comprehension, program analysis, software exploration and program transformation.

The IG approach makes it possible to develop parsers for (complex) language grammars very fast and with very little effort. These parsers can be used in tools that query, modify or give insight to existing code bases. This way it becomes easier to comprehend the structure of large and complex software systems. Research shows that software engineers spend 50% of their time trying to understand the system they need to work with.

5 Relevance for the involved company

The project aims to solve the companies problems with practical application of island grammars described in section 2.

Better support for Island Grammars in SIG’s software analysis tooling makes it easier and faster to add data gathering that is required for the introduction of new programming languages and/or new analyses.

6 Global Planning of the project

6.1 Project milestones

Project start-up Graduate project proposal (this document)
End of research assignment Literature study report
End of master thesis project Final project report, (experimental) Island parser implementation software
End of graduate period Oral presentation, hand over of project results

6.2 Time schedule for project activities

Weekly recurring event: Evaluation/progress meeting with company supervisor (see 7.5)
Two weekly recurring event: Evaluation/progress meeting with university supervisor (see 7.2)
December 2003: Start of the graduate period at the SIG
February 2004: Delivery of draft literature study report
March 2004: Delivery of literature study report
September 2004: Delivery of draft project report
October 2004: Delivery of final project report and all developed software. Last meeting with the examination committee before the oral presentation
End of October 2004: Oral presentation, hand over of project results

Furthermore, the Software Engineering department of the University of Delft organises regular colloquia3 where MSc students can present their project and its progress to other students and staff of the department. At least two of these colloquia must be attended, and a presentation must be given at one of the colloquium meetings.

7 Project organization

The following sections contain contact information for: the graduate student (7.1), the university supervisor (7.2), the university professor (7.3), the involved company (7.4) and the company supervisor (7.5).

7.1 Graduate student

Name: R.C. van der Leek

\[\text{http://swrl.tudelft.nl/}\]

\[\text{http://sepc.twi.tudelft.nl/cse.htm}\]
7.2 University supervisor
Name: L. Moonen
Telephone nr.: +31 (0)15 278 64 11
E-mail address: leon.moonen@computer.org

7.3 University professor
Name: A. van Deursen
Telephone nr.: +31 (0)15 278 24 86
E-mail address: arie.van.deursen@cwi.nl

7.4 Involved company
Name: Software Improvement Group
Address: Muiderstraatweg 58-A, 1111 PT Diemen, The Netherlands
Telephone nr.: +31 (0)20 314 09 50
E-mail address: info@software-improvers.com

7.5 Company Supervisor
Name: T. Kuipers
Telephone nr.:removed
E-mail address:removed

References