**MEMORANDUM**

|  |  |
| --- | --- |
| **TO:** | Paula R. Wellson University of Texas at Austin, Department of Computer Sciences |
| **FROM:** | Janice Swiczhier |
| **SUBJECT:** | Progress on the Scheme Debugging Project |
| **DATE:** | April 3, 1999 |

This memo describes the progress I have made to date on my independent-study project of writing a report on debugging in Scheme. In this memo, I review the nature of the project and describe work I have completed, work I am currently engaged in, and work I plan to complete by the end of the project. As I described in my memo of June 21, this project will result in a technical report whose purpose is to provide readers with practical information on developing and debugging programs in Scheme, supplementary to the material in your textbook, *An Introduction to Scheme and its Implementation*.

**Project Description**

The report is aimed at students in computer science (undergraduate and graduate) who have previous programming experience, but are new to Scheme. The information in this report is needed because readers who have developed programs using compilers for other languages may be unfamiliar with the approaches available with an interactive interpreter and debugger.

**Project Scope**

In my earlier memo, I proposed to cover the following high-level topics:

* Loading the debugging module into the interpreter
* Establishing break levels
* Applying backtrace
* Managing dependencies
* Saving and loading a customized heap image of the Scheme system
* Debugging local definitions
* Debugging native code procedure calls
* Debugging when using functional programming style
* Program design and implementation strategies
* Using stubbed procedures
* Differences between RScheme and other Scheme systems

In my current outline, these are divided into three major parts, with an addendum for topic 11. The three parts are: (a) basic debugging procedures: topics 1–3, (b) advanced debugging procedures: topics 4–7, and (c) general program development strategies: topics 8–10.

**Work Completed**

I have completed first drafts of the sections in part A on loading the debugging module, break levels, and apply-backtrace. I intend to make note of additional material for these sections while working on the later sections, if further background information is needed.

**Present Work**

I am currently working on the sections in part B. Since these sections are highly interrelated, I am working on them roughly in parallel. I am also currently researching information on other Scheme systems for section 11; I have located information on Gambit and DrScheme. I expect the current work to be completed by the end of this week, July 25th.

**Future Work**

Next, I will draft the sections in part C and the addendum on other Scheme systems. Finally, I will fully revise the entire draft, integrating further material where deficiencies have become evident during work on other sections. The final report will be ready for your review on August 6th.

**Conclusion**

Thus far, the project is proceeding well. I have not run into any major problems, nor do I anticipate any in the remaining work. A current outline is attached.

**TENTATIVE OUTLINE**

1. Introduction
2. Basic debugging procedures
   1. Loading the debugging module into the interpreter
   2. Break levels
      1. How to get into them
      2. How to get out of them
   3. Apply-backtrace
      1. How to turn it on
      2. How to interpret output
3. Advanced debugging procedures
   1. Managing dependencies
      1. Reloading code
      2. Rebuilding data structures
   2. Saving and loading a customized heap image of the Scheme system
   3. Debugging local definitions
   4. Debugging native code procedure calls
4. Program development strategies
   1. Debugging when using functional programming style
   2. Program design/implementation strategies
      1. Top-down
      2. Bottom-up
      3. Middle-out
   3. Stubbed procedures
      1. Prompting for user input
      2. Using canned data
5. Addendum: Differences between RScheme and other Scheme systems
   1. Gambit
   2. DrScheme