

Indiana University, Bloomington

Tools and Techniques for Use of the Scheme Programming Language in Undergraduate Education

Principal Investigators:

Christopher T. Haynes
R. Kent Dybvig
Daniel P. Friedman
LoriLee Sadler
George Springer

1. OVERVIEW OF GOALS AND OBJECTIVES

The exceptional simplicity and power of the Scheme programming language that account for its widespread use in computer science research also explain its great potential in education. The full potential of Scheme has not been realized due to the lack of a quality programming environment with a graphic user interface, limited pedagogic materials, and limited awareness of Scheme's pedagogic potential. The goal of this project was to make such software and pedagogic materials widely available.

2. CURRENT STATUS AND ACCOMPLISHMENTS

Extensive pedagogic materials have been developed and made available via the web for five courses and several workshops, as outlined below in the materials and dissemination sections. A software system named SWL, for "Scheme Widget Library", is nearing completion. It is in beta-test and is being ported to Windows 95/NT. SWL is a multithreaded graphics and windowing library for Scheme and a set of tools developed using this library. These tools include:

- a window-based interaction environment with editing capabilities, a powerful history mechanism, and a template mechanism to help guide programming exercises for novice programmers;
- a source-file viewer capable of highlighting matching variable bindings and references, procedure call sites, assignable variables, and other interesting program characteristics;
- a graphical inspector for Scheme objects, including continuations and procedures, with hooks into the source-file viewer;
- a graphical profiler that is fully integrated with the source-file viewer to allow programmers to visualize program "hot spots," i.e., portions of their programs that are executed most frequently;
- a lecture and tutorial scripting tool that allows split-screen viewing of lecture/tutorial scripts and interactive evaluation of Scheme programs; and
- a general-purpose HTML viewer supporting Scheme "applets".

A preliminary version of Scheme++ has been developed and used in several courses. This is an extension of Scheme to support object-oriented programming inspired by C++ object implementation techniques.

The Scheme Repository, a large (currently over one megabyte) collection of information on Scheme from a variety of sources, has been ported to the web and greatly expanded. The home page at <http://www.cs.indiana.edu/scheme-repository> was accessed about 23,000 times in the last 22 months.

3. PLANS FOR REMAINDER OF PROJECT

Further tools based on the SWL system are under construction. Most notable is a graphical debugger and object inspector for Scheme. It supports a point-and-click interface and will give Scheme implementations (with powerful macro systems, which are what has prevented this up to now) what most

C implementations have had for quite a while -- the ability to automatically associate errors with points in program files.

Course development continues. For example, a complete set of lecture scripts for the "Introduction to Programming Languages" course is being completed this semester, about a third of which is devoted to modeling (in Scheme) the static and dynamic semantics of a major subset of Java.

A portable version of Scheme++ is being completed, along with documentation.

Design and implementation of a module system for Scheme is underway. It will allow easy tailoring of Scheme environments to particular sub or supersets of Scheme features. Among other benefits, this would allow the safe hiding of some of Scheme's less savory features (such as dotted pairs) in introductory classes, forcing students to learn good programming habits before allowing them access to more powerful constructs.

Maintenance of the Scheme Repository will continue.

4. MATERIALS THAT HAVE BEEN DEVELOPED

Whenever possible the materials developed through this project are available via the project web cite: <http://www.cs.indiana.edu/eip>. At present this includes several of the papers and presentations listed below, and extensive materials used in the IU Summer Scheme Workshops of 1995 and 1996, and several courses:

- C211: Introduction to Computer Science
- C311: Introduction to Programming Languages
- C431: Assemblers and Compilers I
- C432: Assemblers and Compilers II
- C511: Programming Languages

The software systems mentioned above will be added upon completion.

5. DISSEMINATION ACTIVITIES

In addition to the materials distributed via the web listed above and the publications and presentations listed below, we have offered the following nine weeks of workshops, attended primarily by teachers of computer science at undergraduate institutions.

Summer '95 Scheme Workshops:

Introducing Scheme: June 12 - 23

Using Scheme to Understand Programming Languages: June 26 - July 7

Summer '96 Scheme Workshops:

Introducing Scheme: June 10 - 14

Intermediate Topics in Scheme: June 17 - 21

Advanced Topics in Scheme: June 24 - 28

Using Scheme to Understand Programming Languages: July 1 - 5

Compiling Scheme: July 8 - 12

These workshops were offered at Indiana University in collaboration with PIs of the Oberlin College EI grant. Collaboration with Oberlin also included the SIGCSE'96 Scheme workshop listed below.

Christopher Haynes served as program chair and Daniel Friedman as a panelist of the 20th Anniversary Scheme Workshop at the ACM Principles of Programming Languages Conference, January 1996.

Christopher Haynes co-organized the Workshop on Functional Languages in the Introductory Computing Curriculum at the ACM International Conference on Functional Programming, May 1996. R. Kent Dybvig served as program chair for the conference as a whole.

6. PUBLICATIONS AND PRESENTATIONS

Daniel P. Friedman and Matthias Felleisen, *The Seasoned Schemer*. MIT Press, 1996.

Daniel P. Friedman and Matthias Felleisen, *The Little Schemer*, Fourth Edition. MIT Press, 1996.

R. Kent Dybvig, *The Scheme Programming Language*, Second Edition. Prentice-Hall, 1996.

Christopher Haynes, "Compiling: A high-level introduction using Scheme." Indiana University Computer Science Technical Report No. 469 (September 1996). Submitted to SIGCSE'97.

Christopher T. Haynes, K. Fritz Ruehr, Rhys Price Jones, Richard M. Salter, "Introducing the essence of programming through Scheme." Workshop at SIGCSE, February 1996.

Eric Hilsdale, J. Michael Ashley, R. Kent Dybvig and Daniel P. Friedman, "Compiler construction using Scheme." In the proceedings of the Functional programming languages in education conference (December, 1995),

LNCS 10:22, P. H. Hartel and M. J. Plasmeijer, eds., Springer-Verlag, pp. 251-268. Carl Bruggeman, Oscar Waddell, and R. Kent Dybvig, "Representing Control in the Presence of One-Shot Continuations." Proceedings of the Conference on Programming Language Design and Implementation, 1996, pp. 99-107.

Oscar Waddell, "The Scheme Widget Library User's Manual." Indiana University Computer Science Technical Report, 1996.

Christopher Haynes, "Teaching Programming Languages Using a Scheme-based Executable Formalism." Presented at the NSF Workshop on Exploring Formal Methods in the Early Computer Science Curriculum, City University of New York, September 1995.

Hsianlin Dzen and Christopher T. Haynes, "Type reconstruction for variable-arity procedures." In Proceedings of the Conference on Lisp and Functional Programming, 1994, pp 239-249.

Daniel P. Friedman, "An Introduction to Scheme." Presented at the First International Forum on Computer Programming in Education, ITESM, Mexico, February 1994.

Christopher T. Haynes, "Scheme in Computer Science Higher Education." Presented at the First International Forum on Computer Programming in Education, ITESM, Mexico, February 1994.

7. EVALUATION ACTIVITIES

Evaluations were obtained from participants at all seven summer workshops and the SIGCSE workshop. Course evaluations were also obtained for the C211, C311, C431, C432, and C511 courses.

8. BENEFITS SEEN AND EXPECTED

The results of the workshop and course evaluations were positive. Indeed the workshop evaluations were overwhelmingly enthusiastic.

Experience using on-line lecture scripts in the introduction to programming course, and more recently in the programming languages course, has been very positive. It has allowed more material to be presented and made the material come alive for the students, since every concept is immediately demonstrated with running programs. The curricular changes in themselves have also made a substantial improvement in the choice and extent of material covered, and in student comprehension.

We will shortly be prepared for large-scale class testing of the software developed as part of this project, which we anticipate will be eagerly embraced by students.