

Address: Department of Computing and Information Science,
Queen's University, Kingston, Ontario, Canada

Description of Project:

Objective:

To design and implement a programming language which will allow literary scholars to collect data from texts being studied by specifying requirements rather than the algorithm needed to obtain them.

Starting Date: 4/71

Completion Date: (initial version) 9/73

Current Status:

A first version of the language has been designed, and an implementation is nearing completion. Work is in progress on the design of a more advanced version of the language with many additional facilities.

Computers Being Used:

The first implementation is on an IBM 360/50. Later work will also be carried out on a Burroughs B6700 and a PDP 11/45.

Methodology or General Approach:

In recent years the computer has become widely used as a tool by literary scholars to collect information from texts being studied, and two journals, Computers in the Humanities and SIGLASH Newsletters, are devoted to this topic. Up to the present time, programs for each project have usually been written individually, and since literary scholars rarely have access to programming assistance, it has become necessary to mount programming courses specifically designed for these users. Unfortunately, programming languages such as FORTRAN, ALGOL, and even SNOBOL, are not well suited to this type of work, and few literary scholars wish to become proficient in programming. The objective of this project has, therefore, been to design a language which will enable literary scholars to express their analysis requirements without acquiring expertise in conventional programming languages.

The programming language being designed results from consideration of the varied requirements which have been encountered in some years of consultation and discussion with many literary scholars.

A first version of the language has been prepared which will handle a high proportion of such requirements. Subsequent extensions are under consideration.

Documentation:

A description and definition of the language will be available in 7/73, and a language manual for use with the first implementation in 10/73.

Contact for Future Reference:

(1) Dr. M. Levison, Department of Computing and Information Science, Queen's University, Kingston, Ontario, Canada.

Source of Funding: National Research Council of Canada Grant A8319.
IBM Research Fund.

* A COMPILER BUILDING SYSTEM FOR SMALL-SCALE COMPUTERS *

Name of Project: A Compiler Building System for Small-Scale Computers

Organization: California State University, Chico

Principle Investigator: Orlando S. Madrigal, Ph.D.

Address: Computer Science Department
California State University, Chico
Chico, California 95926

Description of Project:

Objective:

The design and implementation of a Compiler Building System for a small-scale computer that will be capable of processing a large set of programming languages.

Starting Date: 2/1/71

(Estimated) Completion Date: 6/1/73

Current Status:

The project is currently at a stage where all the system modules are being run as a single program on a CDC 3150. Compilers for subsets of Fortran and Algol have already been constructed using the current implementation.

Computers Being Used: CDC 3150 and HP2100

Languages Involved: Fortran and Algol

Methodology and General Approach:

Various researchers in the area of programming languages and translators have attempted to show the theoretical ways of designing a single compiler for different languages. The usual approach has been to show the various methods of syntax analyses with some coverage on semantics design. Most of these research projects have been geared for implementation on large computers (i.e. McKeeman [3], Wegbreit [4], and Wirth [5]).

The major research goal for this project has been the construction of a compiler building system for a small-scale computer. This system is of great use and interest to the compiler-oriented students of the Computer Science Department at California State University, Chico.

The system has been designed with two distinct phases, namely: all compiler routines, up to, and including the parser; and the semantics processor.

The parsing phase involves a bottom-up scheme similar to the method proposed by Feldman [2]. The output of the first phase is a list of macro instructions which is the input to the semantics processor.

The semantics processor is a macro-based system similar to Cheatham's proposal for macros [1]. This part of the compiler building system has been designed to have the capability as a stand-alone system, and hence, it is capable of being used as a macro processor only. It can output either high-level or assembly language code.

The project is in the final phase of implementation on a CDC 3150. A version of the semantics processor has been implemented on a HP 2100 computer with 8K memory. This implementation has been highly useful to students who are greatly interested in macro processors for mini-computers where users have hands-on capability. The entire compiler building system, after complete implementation on the CDC 3150, will eventually be implemented as a single system on the HP 2100. Portions of the system will be micro programmed to take advantage of the micro programming capabilities of the HP 2100.

Documentation:

Existing Documentation:

- (1) Madrigal, O. S., "A Compiler Building System," Class handout for CSCI 350 - Compiler Theory, CSU, Chico.

Potential Documentation:

- (1) Pon, Laymond, "Syntactic Macro Processor for Mini Computers," M.S. Thesis in preparation, to be ready May 1, 1973.
- (2) Rehberg, James, "A Compiler Builder for the CDC 3150," M.S. Thesis in preparation, to be ready May 15, 1973.

Name of Contacts for Future Reference:

- (1) Orlando S. Madrigal, Computer Science Department, California State University, Chico
Chico, California 95926.
- (2) Laymond Pon, 764 Seventh Ave., Sacramento, California 95818.
- (3) James Rehberg, Computer Science Department, California State University, Chico
Chico, California 95926.

Source of Funding:

Computer Time and equipment have been donated by CSU, Chico. Graduate students working for CSU, Chico are doing all the programming tasks for the entire system.

References:

- (1) Cheatham, T. E., Jr., "The Introduction of Definitional Facilities into Higher Level Programming Languages," Proceedings Fall Joint Computer Conference, 1966, 623-637.
- (2) Feldman, J. A., "A Formal Semantics for Computer Languages and its Application in a Compiler-Compiler," Communications of the ACM, 9, 1 (Jan. 1966), 3-9.
- (3) McKeeman, W. M., et al, A Compiler Generator, Prentice-Hall, Englewood Cliffs, N. J., 1970.
- (4) Wegbreit, B., "Studies in Extensible Programming Languages," Technical Report ESD-TR-70-297, Harvard University, Cambridge, Mass., May, 1970.
- (5) Wirth, N., "The Design of a PASCAL Compiler," Software Practice & Experience, 1,4 (Oct.-Dec. 1971), 209-333.