NASA-CR-204386

FINAL

027317

A Toolbox and Record for Scientific Model Development

NASA Grant NCC-2-802 FINAL

MAY 0 6 1997

CC: C.A.S.I.

Rosa T. Rosa

Summary of Research on Research Awards

Thomas Ellman (Principal Investigator)

1 Introduction

Computational scientists and design engineers face recurring problems of specifying, constructing, testing and modifying numerical programs. Scientists often don't know in advance what objects and processes must be modeled in numerical simulation programs. Design engineers often don't know in advance the best way to formulate objectives, constraints and search spaces that are used in programs for automatic design optimization. Both types of professional typically take an experimental approach to the problem. They develop one program, experimentally examine its behavior, and subsequently modify it many times before arriving at a program that is adequate to the task at hand. The process of coding and revising such programs is extremely time-consuming because they are almost always written in conventional programming languages. Furthermore, the resulting programs are difficult to modify and reuse on other problems. In practice, this limits the range of problems that scientists and engineers are able to solve using computational techniques.

The "Toolbox and Record for Scientific Model Development" is intended to address these problems. It supports construction, testing, application and revision of programs for scientific computation and engineering design. The system features a user interface with tools for interactively constructing and testing programs that simulate and optimize the behavior of physical system. The system generates a record of the program construction, revision and testing process. The record encodes the series of programs that were constructed, the assumptions on which they are based, the tests that were applied and additional documentary information. The record is subsequently referenced by tools for controlling application of programs and by tools for revising programs to handle new situations. The system is based on three distinct bodies of research: (1) Deductive synthesis of numerical simulators; (2) Interactive formulation of design optimization strategies; and (3) Knowledge-based management of legacy codes.

2 Deductive Synthesis of Numerical Simulators

The behavior of many physical systems can be specified in terms of algebraic and ordinary differential equations. Simulators for such systems can often be constructed from combinations of integration and root-extraction routines. In one body of work [Ellman and Murata,

1997], [Ellman and Murata, 1996], we extended the technology of deductive program synthesis to handle problems of this type. Deductive synthesis operates by proving that for every legal input there exists an output satisfying the program specification. As a side effect, the theorem prover generates a program for computing the output from the input. We addressed two important technical issues in extending deductive synthesis to problems of this type. First, we needed to decide what portion of mathematical analysis must be formally axiomatized. We developed an axiomatization that captures only the relation between integration and differentiation, as inverses of each other. A complete axiomatization of analysis was not needed. Second, we needed to choose a logical formalism that would be sufficient for reasoning about second-order functions like integration and root extraction, which take functions as inputs. We developed an approach that uses first-order horn logic: Functions are reified as objects and coded as lambda expressions. The knowledge base includes a theory of lambda term equality in order to reason about functions. This approach avoids using a computationally cumbersome higher-order logic. We implemented and successfully tested the system on a variety of realistic simulation problems in the domains of sailing yachts and jet aircraft, as well as several synthetic problems. This research demonstrates that deductive synthesis can be extended to handle specifications in the form of algebraic and ordinary differential equations and implementation languages including second-order functions.

3 Interactive Formulation of Design Optimization Strategies

Numerical design optimization algorithms are not generally reliable when applied to problems with pathologies, such as ridges, discontinuities, plateaus or non-evaluable regions. Design engineers may nevertheless obtain reliable results on such problems by carefully formulating search spaces, objective functions and constraints. Unfortunately, current software tools provide little support for this formulation process. In a second body of work [Ellman *et al.*, 1996], [Ellman *et al.*, 1995], we adapted program transformation technology to develop an environment for interactive formulation, testing and reformulation of design optimization strategies. Our system includes a language for representing optimization strategies that involve multiple search spaces at different levels of abstraction; multiple objective and constraint functions at different levels of approximation; and multiple stages of optimization. Our system also includes a catalog of transforms for converting one strategy into another, and a mechanism for keeping track of strategy derivations and computational experiments. We used the system to construct and test a variety of strategies for designing sailing yachts and jet-engine nozzles. We used a statistical method to evaluate the reliability of these strategies in terms of expected CPU time to obtain a specified level of design quality. Our results demonstrate a factor of 5 speedup in the yacht domain, and a factor of 3 - 10 speedup in the nozzle domain, depending on the desired level of quality. We view this work as an important first step in automating the entire strategy formulation process. To begin with, our system relieves engineers of the burden of coding strategies in conventional programming languages. In addition, our strategy language formalizes hitherto informal strategies for improving the reliability of numeric design optimization. Finally, this research provides a framework for posing questions for future research and codifying the results of future experiments.

4 Knowledge-Based Management of Legacy Codes

Systems for automated design optimization of complex real-world objects can, in principle, be constructed by combining domain-independent numerical routines with existing domain-specific analysis and simulation programs. Unfortunately, such "legacy" analysis codes are frequently unsuitable for use in automated design. They may crash for large classes of input, be numerically unstable or locally non-smooth, or be highly sensitive to control parameters. To be useful, analysis programs must be modified to reduce or eliminate only the undesired behaviors, without altering the desired computation. To do this by direct modification of the programs is labor-intensive, and necessitates costly re-validation. In a third body of work[Keane and Ellman, 1996], [Keane, 1996], we implemented a high-level language and run-time environment that allow failure-handling strategies to be incorporated into existing Fortran and C analysis programs while preserving their computational integrity. Our approach relies on globally managing the execution of these programs at the level of discretely callable functions so that the computation is only affected when problems are detected. Problem handling procedures are constructed from a knowledge base of generic problem management strategies. We showed that our approach is effective in improving analysis program robustness and design optimization performance in the domain of conceptual design of jet engine nozzles.

5 Summary of Benefits

We expect several long term benefits to result from this research. To begin with, our system includes interactive program development tools that promote more rapid development of new programs for scientific computation and engineering design. Our system also maintains a record of the program development process, and the assumptions on which it is based, which promotes more reliable use of such programs by scientists and engineers. We also expect the use of such records to promote wider sharing of programs within communities of scientists and engineers. Finally, we expect that our research will ultimately promote deeper understanding among scientists and engineers of the properties and behavior of the programs they use in their day to day work.

References

[Ellman and Murata, 1996] T. Ellman and T. Murata. Deductive synthesis of numerical simulation programs from networks of algebraic and ordinary differential equations. In *Proceedings of the Eleventh Knowledge-Based Software Engineering Conference*, Syracuse, NY, 1996.

[Ellman and Murata, 1997] T. Ellman and T. Murata. Deductive synthesis of numerical simulation programs from networks of algebraic and ordinary differential equations. *Automated Software Engineering*, 1997. In Press.

[Ellman *et al.*, 1995] T. Ellman, J. Keane, M. Schwabacher, and T. Murata. A transformation system for interactive reformulation of design optimization strategies. In *Proceedings of the Tenth Knowledge-Based Software Engineering Conference*, Boston, MA, 1995.

[Ellman *et al.*, 1996] T. Ellman, J. Keane, A. Banerjee, and G. Armhold. A transformation system for interactive reformulation of design optimization strategies. Accepted with revisions to *Research in Engineering Design*, 1996.

[Keane and Ellman, 1996] J. Keane and T. Ellman. Knowledge-based re-engineering of legacy programs for robustness in automated design. In *Proceedings of the Eleventh Knowledge-Based Software Engineering Conference*, Syracuse, NY, 1996.

[Keane, 1996] J. Keane. Knowledge-based management of legacy codes for automated design. Technical Report HPCD-TR-49, Rutgers University Department of Computer Science, 1996. Ph.D. Thesis.

REPORT OF INVENTIONS AND SUBCONTRACTS

(Pursuant to "Patent Rights" Contract Clause) (See Instructions on Reverse Side.)

Form Approved
OMB No. 0704-0297
Expires Jun 30, 1992

Public reporting burden for this collection of information is estimated to average 5 minutes per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0297), Washington, DC 20503.

1a. NAME OF CONTRACTOR / SUBCONTRACTOR	c. CONTRACT NUMBER	2a. NAME OF GOVERNMENT PRIME CONTRACTOR	c. CONTRACT NUMBER	3. TYPE OF REPORT (X one)
Rutgers. The State Univ. of NJ	NCC 2-302			a. INTERIM [ ] b. FINAL [X]
b. ADDRESS (Include ZIP Code) Div. of Grants and Contracts P.O. Box 1089 Piscataway, NJ 08855-1089	d. AWARD DATE (YYMMDD)	b. ADDRESS (Include ZIP Code)	d. AWARD DATE (YYMMDD)	4. REPORTING PERIOD (YYMMDD) a. FROM 7/1/93 b. TO 12/31/96

SECTION I – SUBJECT INVENTIONS

5. "SUBJECT INVENTIONS" REQUIRED TO BE REPORTED BY CONTRACTOR / SUBCONTRACTOR (If "None," so state)

a. NAME(S) OF INVENTOR(S) (Last, First, MI)	b. TITLE OF INVENTION(S)	c. DISCLOSURE NO., PATENT APPLICATION SERIAL NO. OR PATENT NO.	d. ELECTION TO FILE PATENT APPLICATIONS (1) United States (a) Yes	(1) United States (b) No	(2) Foreign (a) Yes	(2) Foreign (b) No	e. CONFIRMATORY INSTRUMENT OR ASSIGNMENT FORWARDED TO CONTRACTING OFFICER (1) Yes	(2) No
	none							

f. EMPLOYER OF INVENTOR(S) NOT EMPLOYED BY CONTRACTOR / SUBCONTRACTOR		g. ELECTED FOREIGN COUNTRIES IN WHICH A PATENT APPLICATION WILL BE FILED	
(1) (a) Name of Inventor (Last, First, MI)	(2) (a) Name of Inventor (Last, First, MI)	(1) Title of Invention	(2) Foreign Countries of Patent Application
(b) Name of Employer	(b) Name of Employer		
(c) Address of Employer (Include ZIP Code)	(c) Address of Employer (Include ZIP Code)		

SECTION II – SUBCONTRACTS (Containing a "Patent Rights" clause)

6. SUBCONTRACTS AWARDED BY CONTRACTOR / SUBCONTRACTOR (If "None," so state)

a. NAME OF SUBCONTRACTOR(S)	b. ADDRESS (Include ZIP Code)	c. SUBCONTRACT NO.(S)	d. DFAR "PATENT RIGHTS" (1) Clause Number	(2) Date (YYMM)	e. DESCRIPTION OF WORK TO BE PERFORMED UNDER SUBCONTRACT(S)	f. SUBCONTRACT DATES (YYMMDD) (1) Award	(2) Estimated Completion

SECTION III – CERTIFICATION

7. CERTIFICATION OF REPORT BY CONTRACTOR / SUBCONTRACTOR (Not required if [ ] Small Business or [ ] Non-Profit organization.) (X appropriate box)

a. NAME OF AUTHORIZED CONTRACTOR / SUBCONTRACTOR OFFICIAL (Last, First, MI)	c. I certify that the reporting party has procedures for prompt identification and timely disclosure of "Subject Inventions," that such procedures have been followed and that all "Subject Inventions" have been reported.	
b. TITLE	d. SIGNATURE	e. DATE SIGNED 5/2/97