Research Interests and Directions

Spencer Rugaber

January 1, 2004

Research Area

My primary research area is software engineering, the construction of complex software systems by teams of people using well-defined engineering methods and procedures. Within this area, my research concerns the topics of program comprehension and software design representation. Program comprehension includes manual and automated techniques for increasing knowledge of software artifacts. Software design representation concerns the expression and analysis of design information in support of activities including critiquing and generation of implementations. In addition to these software engineering topics, my work overlaps with research in the areas of human-computer interfaces and programming languages. The overarching theme of the work is the management of complexity by the application of abstraction and specification.

Accomplishments

During my tenure at Georgia Tech, I have published extensively in my research area. These publications have usually taken the form of refereed conference publications for the *International Conference on Software Maintenance*, the *Working Conference on Reverse Engineering,* and the *International Workshop on Program Comprehension*. One of the conference publications was selected as best paper, and several of these articles have been reprinted in books. I have also had journal articles appear in *IEEE Software, Automated Software Engineering*, and the *Journal of Management Information Systems.* Moreover, I am the author of the article on "Program Comprehension" in the *Encyclopedia of Computer Science and Technology.*

My contributions to the field include a validated reverse engineering method (Synchronized Refinement), a new approach to design recovery based on domain analysis, the recognition and exploration of a programming phenomenon exacerbating software maintenance (*interleaving*), and an approach to the migration of user interfaces.

For five years I was Vice Chairman of the IEEE Computer Society Technical Committee on Reverse Engineering and led the effort in that group to organize a research infrastructure for the reverse engineering research community. We constructed an automated repository of bibliographic information, test data, and tools. I have served as a Steering Committee member and Program Co-chair for the *International Workshop on Program Comprehension*. I was Program Co-chair for the *Working Conference on Reverse Engineering*, hosted by Georgia Tech in the fall of 1998. I have participated on the program committees of all three of the conferences mentioned on numerous occasions. In addition, I have been a member of the program committees of the *International Conference on Software Engineering.* and the *Automated Software Engineering Conference.* In recognition of my contributions to the research community, I was awarded the first *Outstanding Contribution over Time* award at the 1997 *Working Conference on Reverse Engineering.* More recently, for the period of November 2001 until August 2002, I served as Program Director for the Software Engineering and Languages program at the National Science Foundation.

My research program has been funded both by the U.S. government and by industry. I have

received grants and contracts from DARPA, NSF, and the Army Research Laboratory (ARL). Also, for nearly ten years, I received gifts in support of research from Nortel (formerly BNR). Other sources of industrial funding include DEC, SUN, and Spectra Research.

Current Research Projects

I am currently engaged in several funded research projects. The largest project with which I am associated is the DYNAMO project (*Dynamic Assembly from Models*) funded by DARPA as part of its Dynamic Assembly for System Adaptability, Dependability, and Assurance program. DYNAMO is concerned with the specification of assemblies of software components in such a way that formally expressed assembly properties can by guaranteed by the automatic generation of wrapper code. This work is joint with Kurt Stirewalt of Michigan State University.

I am also the principle investigator of the NSF-funded *Ectropic Design Project: Intelligent Collaboration Spaces,* jointly with Mark Guzdial of Georgia Tech. The Ectropic project explores ways in which the conceptual integrity of software systems can be maintained or improved over time as it undergoes maintenance and enhancement. Two ways in which this is accomplished are by tightly coupling design information with code and by providing intelligent design critics that note when design diverges from code.

In addition to the above, I have completed preliminary work and written a funding proposal for a research project on the formal foundations of reverse engineering. Reverse engineering is concerned with the extraction and representation of design information from existing code. I am developing a theoretical basis for work in this area that will enable objective evaluations to be made of reverse engineering representations thereby supporting managerial prediction and control of reverse engineering projects.

I am also engaged in two research projects with undergraduates at Georgia Tech. The Plexicon project has developed a hypertext dictionary of programming concepts in support of students in a sophomore-level *Languages and Translation* course. The TAJ (Transformational Abstraction in Java) project has developed a program documentation tool for supporting reverse engineering efforts.

Research Vision

As computers have become more widespread, the demand for software has increased dramatically. Increased demand can lead to reduced quality unless steps are taken to improve software development processes and tools. The major culprit is complexity, and the major weapon to deal with complexity is abstraction. The work described above is concerned with the cataloging, representation, application, and theoretical foundations of abstraction on existing and to-be-built software.

In my opinion, our understanding of the software development process and the abstractions that underlie it are still immature. We need to develop a better understanding of abstractions, provide a firm foundation, and apply our ideas to existing and new software systems.

Acknowledgments

Although the text above is written in the first person, it actually describes the work of many people with whom I have collaborated, both faculty colleagues and students, including Ph.D., Masters, and undergraduate research participants.