Position Paper on Research Infrastructure for Reengineering

Spencer Rugaber and Linda M. Wills
Georgia Institute of Technology
{spencer@cc, linda@ee}.gatech.edu

Abstract

This paper proposes a joint effort to build a research infrastructure for the area of reengineering. It is an update and condensation of a proposal made last fall at the Third Working Conference on Reverse Engineering held in Monterey. An elaboration of the points made here as well as a full set of references can be found in that paper.

Keywords: reengineering, research, infrastructure

1. State of reengineering research

Reengineering research has had notably little effect on actual software reengineering practice. The typical software developer on a reengineering project is still using code reading and test runs as the primary ways of learning about an existing system, where an editor and compiler are the primary tools used to change it. We see several reasons for the research area's lack of impact.

- We have not done a good job in communicating the costs and benefits of reengineering in the software lifecycle.
- We have not been able to develop an effective decision procedure to help a manager judge alternatives such as starting over, reengineering, continuing maintenance, or abandoning a product.
- There is no validated reengineering process that a manager can use to guide a project.
- The diversity and complexity of real programming languages, compilers, and hardware platforms has prevented tools from effectively penetrating throughout the marketplace.

2. A research infrastructure

This position paper describes a process by which the reengineering research area might become more productive. The main idea is to better share and build upon each other's results.

Here are some steps that we as a community can take to address these issues and to leverage our efforts. Some of these are already underway, some are understood but need to be implemented, and some are themselves research questions that need to be examined. All of them need contributions from volunteer participants.

Taxonomy: IEEE Reengineering Taxonomy project. Initiated by Elliot Chikofsky and James Cross, this effort has resulted in a published description of our field's terminology.

Common portable and interoperable intermediate representations: Although there have been several efforts within the larger computer science community to establish standard forms for intermediate representations (IRs), their penetration into the commercial marketplace has not been pervasive. Consequently, researchers have had to divert effort into building their own or retrofitting their tools on a case-by-case basis.

Reengineering resources: The advent of the World Wide Web has encouraged individual research groups to construct home pages containing references to resources related to the field of reengineering. Such pages list conferences, vendor and tool descriptions, and bibliographies describing relevant literature.

Educational and training materials: Technology transfer is often hindered because reengineering techniques and tools are complex and difficult to understand. Developing background reengineering materials would be valuable in communicating what reengineering has to offer to the software industry.

Reengineering task descriptions: Such descriptions include definitions of specific activities, their inputs and outputs, their costs and benefits, and any other collected wisdom the field can offer to practitioners.

Graduated series of milestones/challenges: Another step we can take is to define a series of specific research challenges to strive for.

Standard data sets/benchmarks: Coordinated and measurable progress in our field depends critically on establishing standard data sets and benchmarking tasks that can be used to quantitatively evaluate and compare reengineering tools and techniques. The standardization of these data sets, in conjunction with efforts to establish clear reengineering task descriptions, will provide the basis for benchmarking task scenarios representative of important, common reengineering activities.

3. Repository

To address the problems of Section 1 and to coordinate the efforts of Section 2, we suggest the development of a moderated research repository. The repository would include such components as the following:

Data sets: specific source code and test data which researchers can target to compare results.

Scripts and other code: such as Refine language programs for specific analyses.

Libraries of cliches and transformation rules: in support of program recognition and transformational analysis.

Grammars: machine processable programming language grammars.

Experiments: descriptions of repeatable program comprehension experiments and the materials to replicate them.

Case studies: a systematic collection of case studies, uniformly documented so as to provide guidance to development managers. This would serve as a first step to a cost model, and for reengineering process management.

Public domain utilities: research tools and basic utilities (such as parsers, visualization tools, analyzers, graph layout and editing facilities), including documentation supporting automatic applicability qualification.

Literature references: up-to-date references and literature in our field such as texts, handbooks, and papers.

There are many difficulties, including technical, legal, logistical, and even theoretical issues that make constructing this repository problematic. For example, for the repository to be truly useful, it should be systematically documented and indexed—which requires a substantial effort. But we do not think that the problems are insurmountable, and the value of such a repository, both to practitioners and to researchers, warrants pursuing the possibility. Steps have begun toward the collection of existing resources.

4. Infrastructure implementation

Many of the discussions leading to this paper have taken place in the context of the Committee on Reengineering (CORE) of the IEEE Computer Society's Technical Council on Software Engineering. CORE has created a web site (http://www.tcse.org/revengr/), where a repository has been established to collect reengineering resources, the taxonomy of terminology, pointers to relevant conferences and literature, and the many products of our research.

We suggest that a next step is for individuals to volunteer to organize the various components by presenting an organizational plan including a time and effort estimate, resources required, issues to be resolved, etc. On-line discussion would raise awareness and hopefully stimulate a synthesis.

Unfortunately, even an outpouring of volunteer effort from the reengineering research community will not suffice to implement the proposal. We need external input from several other research and industrial areas including test and evaluation, compilers, and domain analysis.

We also propose the establishment of an advisory board to which we would report progress and from which we would solicit advice and contacts. Among the participants would be industrial practitioners, government funding agents, an IEEE representative, academic and industrial researchers, and tool vendors (both compilers and CASE tools).

5. Update

The infrastructure effort made progress on several fronts during the Third Working Conference on Reverse Engineering held in Monterey, California this past November.

- The Taxonomy Subcommittee (Chair: James Cross, Auburn University) is reviewing and recommending changes to terms and descriptions in three existing glossaries (linked to the Taxonomy Project web page at http://www.tcse.org/revengr/). This subcommittee’s goals extend further than defining glossary terms—to imposing a taxonomic structure on the concepts of the field.
- The Benchmarking Subcommittee (Chair: Bryce Ragland, STSC) plans to provide standard data sets, benchmarking tasks, and concrete milestones for evaluating progress in our field.
- Chikofsky’s Challenge—During the WCRE Plenary session, Elliot Chikofsky announced his plans to make available a legacy system that can be used as a common target for reengineering tool developers and researchers. He has recently announced that the Reverse Engineering Demonstration Project is ready to go. More details can be found by looking at URL http://www.worldpath.com/reproject/.
- The Outreach Subcommittee (Chair: Michael Blaha, OMT Associates) plans to organize conference tutorials that bring reengineering concepts to practitioners outside the reengineering community. In addition, tuto-

rials are planned for those within the community on advanced theory and techniques fundamental to reengineering.

- The Resource Repository Subcommittee (Chair: Spencer Rugaber, Georgia Tech) is enlisting the help of many volunteers to collect a wide range of on-line resources, including tools, data sets, contact information, bibliographies and case studies. The goals of this subcommittee are to enable researchers and practitioners to share and leverage their results and to promote effective technology transfer of research. As part of this effort, Rainer Koschke from the University of Stuttgart has recently made available his reverse engineering bibliography. Currently the bibliography provides a search mechanism generating references in BibTex format. Plans include other formats as well as a U.S. mirror site at Georgia Tech. The bibliography is available at URL http://www.informatik.uni-stuttgart.de/ifi/ps/reengineering/

6. Summary

We have proposed several steps that we believe will encourage maturation of the research efforts in the area of software reengineering. These include increased interaction with industry, the development of a repository of research artifacts, and convergence of intermediate representations. The paper is a proposal that requires considerable discussion and consensus before it can be realized. We encourage your thoughts and your participation.