Self Assessment Statement

Teodor Rus

April 16, 2008

1 Introduction

This document contains the self-assessment of my research, teaching, and service over the last 5 years and my plans for the next five years research, teaching, and service. These assessments and plans need to be regarded in the context of current computer-based problem solving methodology which is characterized by an “one-size-fits-all” problem solving pattern. That is, current computer-based problem solving process unifies all problem-domains and problems, irrespective of their conceptual logic, into the unique pattern of program development. While this problem solving process proved itself very successful over the short life of today computers, it also revealed difficulties that threaten computer technology itself. This statement is supported by human-cognition spiral which states that: computer-use increases human-cognition power which in turns requires more computer use. The major consequences are: the number and the complexity of software tools supporting program development and execution increases with the number of problem domains and problems to solves; difficulty of computer use (programming) increases with software-tool complexity; software-tool diversity threatens computer education (from hundreds of languages and tools what should we teach?); software maintenance threatens computer technology itself (IBM program on autonomic computers is based on an analysis that shows that education is unable to produce enough CS experts to maintain future computer systems). The only viable solution to the problem raised by growing software complexity seems to be the development of a domain dedicated problem-solving methodology where: (1) problem domain experts are allowed to use the natural language of the domain to develop solution algorithms (this is conceptual computation) and (2) algorithm execution is performed directly by interpretation (no programming as usual is needed) by virtual machines dedicated to application domains. Hence, I see my current research and teaching activity as an effort to raise computer science awareness on the consequences of continuously increasing software tool complexity and proportionally increasing difficulties faced by computer user during computer-based problem solving process and to pursue our effort for the development of software tools that offer a viable solution to these problems.

2 Assessing my research

My research during the last 5 years has been dedicated to the problems specified in the introduction and evolved as follows:

1. In 2003 I started the process of transforming the software tools generated by TICS program into tools supporting a domain-oriented problem solving environment. See paper “Domain-Oriented Component-Based Automatic Program Generation”.

1
2. Due to sickness-problems, during 2004–2005 my research activity has slow-down. However, during this period together with Don Curtis (one of my PhD students) and James Hunsaker (one of my Masters students) we set the foundation of the Application Driven Software (ADS) development which is currently supported by SADL language and SADL interpreter. During these years through three NSF grant proposals: A Process Based Application Development System (March 2004), Integrating Computer Education with Ontology Engineering (June 2005), An Application Driven Problem Solving Methodology (October 2006), neither funded, I introduced the concept of Computational Emancipation of an Application Domain (CEAD). Using this concept I have shown that in order to develop a viable solution to the problems raised by software tool complexity the problem-solving process must be split among the domain experts and computer experts such that they should be able to work concurrently and independent of one-another on the same problem, bridging the conceptual gap. This splitting moves the problem solving difficulty towards computational emancipation of the problem domain and thus provides a logical path to fight software complexity and its consequences. This idea is currently promoted by NSF using the concept of “computational thinking” and is supported by the new NSF program named “Cyber-Enabled Discovery and Innovation” (CDI). Unfortunately I failed to submit my proposal CDI-Type II: Computational Emancipation of Problem Domains (February 2008). However, I will submit this proposal during September 2008 at the second edition of CDI call for proposals.

3. In 2006, with the paper “Application Driven Software Development” (by T. Rus and D. Curtis) presented at the International Conference on Software Engineering Advances, ICSEA 2006, we have launched the idea of developing software tools to support a domain-driven problem solving methodology which is semantically based.

4. In 2007, with the paper “Towards an Application Driven Software Technology” (by T.Rus and D. Curtis) presented at the 2007 International Conference on Software Engineering Research & Practice the ADS software tools (CEAD, SADL, and SADL Interpreter) have been demonstrated on problems in Linear Algebra.

5. In 2008, with the grant proposal project ”Integrating Research and Teaching through Cyber-infrastructure” and the papers “Application Driven Software for Chemistry” by D. Curtis and T.Rus (accepted to be presented at the IEEE International Conference on Electro/Information Technology, EIT-2008, “Natural Language of Application Domains versus Domain Specific Languages” by C.Bui, D. Curtis, and T. Rus (submitted to AMAST 2008) and ”Liberate Computer User from Programming” by T. Rus (invited talk to Students in Applied Math) we are really raising the awareness on software complexity and show the viable solutions for preventing it.

6. For the next 5 years I plan to computationally emancipate such problem domains as Programming Language Design and Implementation (through the PhD thesis developed by D. Curtis), Natural Language Processing (through the PhD Thesis developed by C. Bui), and System Software (through the book I am currently developing as a new edition of the book “System Software and Software Systems” Vol 1: Systems Methodology for Software, by T.Rus and D. Rus.

3 Assessing my teaching

During the last 5 years I did teach the following courses from CS curricula:
1. 22C:132 *Parallel Programming* in Fall 2003. Unfortunately this course designed to teach our students how to program modern-parallel computers is on its way of disappearing due to lack of appropriate equipment.

2. 22C:96/01 Introduction to XML in Fall 2003. This course is very much alive through the Ken Sloneger offerings.

3. 22C:186 Compiler Construction in Spring 2005, and Spring 2006. This is another course that needs support as graduate and undergraduate student education on compiler design and implementation is in much demand. I plan on teaching this course again in the future using the tools we currently develop for computational emancipation of the programming language design and implementation.


5. 22C:135 Theory of Computation (Fall 2004, Fall 2005, Fall 2006, Fall 2007).


During the last five years I also kept alive by “overload teaching” the *Seminar on Language Processing Technology* which I developed as a mechanism of keeping my PhD students in touch with the research progress on our domain of expertise. The offerings of this seminar are:

1. 22C:398 Language Processing Technology (Fall 2003 and Spring 2004).
2. 22C:398 Domain-Oriented Ontology-Based Software Development (Fall 2004).
3. 22C:398 Component Based Software Development (Spring 2005).
4. 22C:398 From Discourse to Logic (Spring 2008).

4 Assessing my service

During the last 5 years my service to the profession has decreased in intensity. I continue to review journal papers which present topics in my domain of interest and publish these reviews in Computing Reviews. But this activity has decreased (2 such reviews during the last 5 years). I also have slowed down with my participation in grant-reviewing process and PC membership of various internal and international conferences. However, I do continue to give invited talks. My last invited talk has been given on 4 April 2008 to Students in Applied Mathematics and Computational Science, University of Iowa. My next invited talk will be at the AMAST 2008 conference to be held on July 28-31 at the University of Illinois at Urbana.

As far as committee membership in Computer Science Department is concerned, my activity is really slow. I fail to volunteer for various activities and people rarely solicit my expertise. Though I am still responsible with CS library.