An NSF-Sponsored Summer Research Program for Undergraduates: Evolution and Refinement*

W. Eric Wong, Vidroha Debroy and Andrew Restrepo
Department of Computer Science
University of Texas at Dallas
Richardson, TX 75080, USA
{ewong,vxd024000,arestrep}@utdallas.edu

Abstract **-** This paper reports the organization of an undergraduate research summer program in 2010 at the University of Texas at Dallas, which is the second iteration of a three-year project sponsored by the National Science Foundation. The primary research subject is "Verification and Validation for Software Safety." A special focus is on the evolution and refinement using lessons learned from the previous program. In keeping with this trend, we also list our plans for future programs.

Index Terms - REU program, NSF, software safety

I. INTRODUCTION

One of the missions of the National Science Foundation (NSF) is to actively encourage and provide support for undergraduate students who are involved with (and directly participate in) hands-on research projects. This is critical as the sooner a student is exposed to research principles and practices, the better, and consequently the faster they will be able to grasp what research (and a research project) is really about. In keeping with this objective, the NSF has created a special program, REU (Research Experiences for Undergraduates) [4], to support active research participation by undergraduate students in any of the areas of research funded by NSF. We at the Department of Computer Science of the University of Texas at Dallas (UTD) have received a three-year funding (2009–2011) from NSF to run an REU program with a focus on "Verification and Validation for Software Safety" [6].

While software safety (defined as freedom from software-enabled accidents or losses) has always been considered to be important, it is more so today than ever before, as software is increasingly being used to monitor and control safety-critical devices and processes in areas such as medicine, energy, and transportation [1,8]. It is therefore vital for today's software engineering/ computer science students, who shall soon go on to become tomorrow's software engineers, to be not just familiar but well-versed in the area of software safety. This claim is supported by the fact that software safety has also been recommended as an independent curriculum module [1,2]. However, despite such recommendations and realizations, software safety is still not receiving the attention it deserves as an educational topic, and indeed it is not adequately covered (taught) by any undergraduate curriculum in Computer Science and/or Software Engineering. This is the primary motivation for our REU program having a focus on the software safety, and through such a program we hope to expose students to this important subject as early as possible, and instil in them a strong fundamental knowledge of the field.

This is in fact the second year that our REU program has been hosted at UTD. A total of ten students were selected nationwide from 7 Universities to participate in the eight-week program (June 1 to August 23, 2010). To guide the students, in addition to the professors in the Department of Computer Science at UTD (who acted as faculty mentors) and PhD students doing research on related subjects (acting as student mentors), we also had active participation from our Industry Advisory Board (IAB) with member experts in software safety from Lockheed Martin Aeronautics Company, Raytheon, and EDS/HP. Please refer to the website at http://paris.utdallas.edu /reu/people.html for more details.

An especially unique characteristic of our 2010 REU program is that it was held concurrently, and matched perfectly, with another ongoing project – "*Foundational Requirements for Competency in Software Safety*" which is supported in part by NSF funds for a Net-Centric Industry University Collaborative Research Center (I/UCRC) of which UTD is a member, and in part by industrial memberships of this center [3]. The I/UCRC project produces a set of instructional materials for an overall picture of how software safety can be achieved effectively. These materials can be (and have been) used to help REU students (no matter what their previous level of experience with software safety) quickly acquire a good understanding of existing software safety standards and practices, as well as an awareness of emerging trends and developments in the area.

This I/UCRC project and our NSF REU project thus mutually enrich each other, and are able to leverage each other well. While the instructional materials developed by virtue of the I/UCRC project will serve as the basis for a series of training seminars to prepare REU students prior to their research projects, at the same time, the feedback on the training will be used to help further improve the materials themselves. In fact, results of the three REU research projects described in Section III (exploring the impact of safety requirements on testing software, the role of software in catastrophic accidents, and techniques for a better software safety analysis) can be used to create enhanced instructional material for the I/UCRC project. Such a direct and symbiotic relationship between practice (our I/UCRC project) and research (our REU projects), and also the

*This work is supported by the National Science Foundation (NSF CCF-0851848). Professor W. Eric Wong is the faculty supervisor and the principal investigator of this research initiative. Mr. Vidroha Debroy is a student mentor.

close collaboration between industry (I/UCRC) and academia (REU), truly makes our REU program unique, and helps provide students with solid experience from both an industrial and an academic standpoint.

The rest of the paper is organized as follows. Section II discusses some of the salient features (including the research projects) of the 2009 REU program, followed by Section III which discusses, in detail, the 2010 REU program (along with how we took our experience from the 2009 program into account). Section IV goes on to list some of our intended enhancements with respect to the 2011 REU program, thereby demonstrating our commitment for continual evolution of the REU program. Finally, we present our conclusions and acknowledgements.

II. HIGHLIGHTS OF THE 2009 REU PROGRAM

The 2009 REU program at UTD involved three carefully designed research projects for the students to work on:

(1) ***An Evaluation of Software Standards***

There are hundreds of overlapping software safety standards today, that interact with an even larger number of additional standards, thereby making it challenging for safety engineers to select a safety standard well suited to their project. This project aimed to perform an evaluation of safety standards across 15 criteria proposed by us.

(2) ***The Role of Software in Catastrophic Accidents***:

In the past, unsafe practices in the implementations of several safety-critical software systems have led to catastrophic accidents. The goal of this project was for students to study these accidents and partially (if not completely) summarize their root causes and the lessons that can be learned from them so that such mistakes can be avoided in the future.

(3) ***Software Safety – An Integrative Approach***

The safety analysis process used today is not based on an integrated model that considers both functional and safety requiremetns simultaneously. It does not give a thorough analysis of all the scenarios that may lead to possible failures. The objective of this research was to integrate safety analysis methods with standard functional requirements to reduce failures inherent in performing the two independently. The proposed way of accomplishing this was by integrating the system's functional behavioural model (represented by UML state machines) with fault tree diagrams derived from the safety hazard analysis, which would give a clearer picture of the possible system failures. The identified failure-causing scenarios could then be used in guiding subsequent validation to ensure the safe operation of the system.

In lieu of space limitations and to maintain our focus on the 2010 REU program, we do not go into further detail on these projects. For readers interested in specifics of the above projects, as per the 2009 REU program, we refer them to [7]. To help the REU students with no background in software safety, guest lectures and tutorials were provided. In addition to the research at UTD, students had field trips to locations such as Lockheed Martin Aeronautics Company for a tour that walked them through the assembly lines of the F16, F22, and F35 fighters. This helped them better understand how software safety was actually applied in practice. Also, fun activities (such as a 4th of July barbeque party) were hosted for the REU students and their friends giving them a break from research. The program concluded with the students being given the opportunity to present their research and receive feedback from their peers, as well as representatives from the industry partners and faculty at UTD.

III. A REFINED 2010 REU PROGRAM BASED ON THE 2009 EXPERIENCE

The 2010 program was run in a similar way as the 2009 program including (1) a kick-off orientation at the beginning where students learned about the upcoming projects and the resources available to them, and received a brief overview of UTD, and (2) special training courses and tutorials by university professors as well as experts from our industry partners to help students improve their background in *software safety*. The ten students who participated in the 2010 program (from 7 universities) were divided into three groups such that each had a well-balanced distribution of skills. However, the objective was not to create new research projects from scratch, but rather to extend and further develop the projects that were already underway. Thus, three projects were carefully selected for the students:

(1) ***Project 1***: Accounting for the effects of safety requirements on the verification and validation of software (linked to Project 1 of the 2009 program as listed in Section II).

The students focused on answering questions such as: does it cost more to test safety-critical software than software with less stringent safety requirements? Do existing safety standards (such as DO-178B, MIL-STD-882D, DEF-STAN-0055 & 00-56, and FAA Systems Safety Handbook) provide appropriate guidelines and sufficient details on how to test software with high safety requirements? Why could projects such as the Lockheed Martin's C-130J conduct their testing process by spending less than a fifth of normal industry costs, even though its software was developed to reach Level A (the highest level) of the DO-178B software standard?

(2) ***Project 2***: Characterizing the culpability of software in accidents resulting in death, injury, property damage, and significant financial loss (linked to Project 2 of the 2009 program as listed in Section II).

Our previous study [8], partially based on the results of the 2009 REU summer program, analyzed how an error in

software or an error in its use contributed to 14 catastrophic accidents. In addition, so as to analyse a more recent cross-section of accidents from aeronautics, astronautics, medicine, nuclear power generation, transportation, finance, military, etc., the students of the 2010 REU program also focused on what lessons could be learned from these accidents. In particular their goal was to answer the following two questions: What is the context of each lesson and under what circumstances is each lesson (singularly or collectively) applicable? Also, can we draw any meaningful conclusions that can then be used to prevent future accidents?

(3) ***Project 3***: Integrating fault trees and UML state machine diagrams to improve software safety analysis (linked to Project 3 of the 2009 program as listed in Section II).

Poorly designed software systems are one of the main causes of accidents in safety-critical systems, and thus the importance of safety analysis for software has greatly increased over the recent years. Software safety can be improved by analyzing both its desired and undesired behaviors, and this in turn requires expressive power such that both can be modeled. However, there is a considerable gap between modeling methods for desired and undesired behaviors. This project focuses on methodology which can bridge the gap between fault trees (for undesired behaviors) and UML state machine diagrams (for desired behaviors). More specifically, the rules and algorithms that facilitate the transformation of a hazard (in the context of fault trees) to a UML state machine diagram are studied. Such a transformation can help engineers identify how the hazards may occur, thereby allowing them to prevent their occurrences.

Throughout the program, weekly group meetings were held to discuss the research outcome and monitor the intermediate progress. Feedback from faculty and graduate student mentors, and members of IAB were also provided. At the end of the program, each group was required to finalize their research and present the results to other students as well as invited guests. Similar to 2009, field trips to Lockheed Martin Aeronautics in Fort Worth, Texas and HP/EDS in Plano, Texas were organized to expose students to the inner workings and experiences of industry practitioners as a contrast to the knowledge learned in the classroom. Additionally, we encouraged students to take a break from their research and also enjoy their summer by organizing a trip to the Six Flags amusement park.

While the 2010 REU program shared some similarities with the 2009 program in terms of organization and timeline, one of our major objectives was to better execute the 2010 program based on our experience with the 2009 program. Two important lessons we learned from 2009 are: "undergraduate students may be capable of writing, but not always technical writing" and "some do not have an acceptable level of communication skills" [7]. To address these concerns we specifically organized:

a. Lectures on technical writing

We provided students with special lectures on the subject of technical writing, emphasizing particularly the skills required by researchers and professionals in the field of computer science and software engineering. Sample templates and additional resources such as the Chicago Manual of Style [5] were provided as references. Important techniques were addressed such as the identification of the target audience and the organization of key points. Students were trained to convey written information in a concise and effective way. Such skills are critical whether the goal is to create a brief technical report or a full research paper. Furthermore, while students that pursue a career as a scientist or engineer will benefit most from these lectures, they are also broadly applicable to a variety of alternate career paths that require writing ability.

b. Videotaped presentations

Project presentations given by the students were videotaped and later reviewed to identify potential improvements. These reviews not only focused on the content of the presentation, but also emphasized the presenter's ability to successfully communicate with the audience. Techniques such as how to develop a script and speak effectively were discussed. This allowed students to sharpen their skills in preparing and delivering a good technical presentation.

Proficiency in both written and oral communication is a valuable complement to technical ability for both students and industry professionals – a competent scholar or engineer should be able to not only obtain results, but also understandably and clearly explain how these results were achieved, as well as the observations and conclusions drawn from these results.

In addition, the 2010 Summer Program also included strong:

c. Interactions between REU and UTD students

Special lectures by guest speakers from Lockheed Martin Aeronautics Company, Kennedy Space Center of NASA (The National Aeronautics and Space Administration), and the IEEE Reliability Society were organized such that they included REU students in our 2010 class, as well as UTD undergraduate students enrolled in CS/SE 4367 (Software Testing and Quality Assurance) and graduate students in CS/SE 6367 (Software Testing, Validation & Verification) during the summer semester. The in-class exercises were presented to mixed groups so that REU students could cooperate with UTD students who might have a richer background in software testing. The varied knowledge and experience of the UTD students were leveraged to enrich the discussions and significantly benefit the REU participants. This is notable as very few other REU programs have the necessary environment to enable such interaction – UTD

offers a B.S., M.S., and PhD in software engineering, and software testing is a core course in these programs.

d. Interactions with Faculty and Student Mentors

Outside of the normal meeting schedule, the faculty mentors observed an open-door policy such that REU students could stop by for guidance at any time. Graduate student mentors connected to each of the selected research projects were similarly made available. This provided students with an avenue for a more detailed one-on-one discussion that was not possible in the setting of the regular group session. Furthermore, the availability of mentors ensured that students did not struggle too long with a difficult problem before receiving advice or clarification.

In summary, we were very careful to take into account our experiences from last year (2009) in organizing the REU program this year (2010). Throughout this paper, we have purposefully refrained from using expressions such as 'the 2010 program improved upon the 2009 program', so as to be fair to last year's program. However, it is safe to state that the REU program this year built upon, and refined, last year's program. It is out intent to continue to evolve (for the better) and thus, we list some of our initial goals for the REU program of 2011 in the following section.

IV. FUTURE PLANS FOR THE 2011 REU PROGRAM

Our further refinements for next year's (2011) REU program include, but are not limited to, the following:

- **Fostering even more industry participation**

We have observed the industry participation that we have brought into our REU program to be successful and well received for both the 2009 and 2010 classes. It is therefore our intent to continue this trend, and in fact try to strengthen it, with regards to the 2011 program. The increased industry participation shall serve to promote more research/experuence exchanges between students and industry practitioners.

- **Making use of a repository of materials**

Having the REU program twice already (2009 and 2010) we hope to be able to organize all of the developed materials and artifacts into a central repository. This shall prove invaluable to the REU students from next year (as well as other parties such as engineers affiliated with the sponsors of our I/UCRC project) who can use the materials to quickly learn about software safety and transition into the projects. It is our intent to eventually make this repository publicly and freely available, subject to legal constraints, etc.

- **Providing more guest lectures and tutorials**

The guest lectures and tutorials that were provided to the students (for example, the seminar on software safety by Dr. Michael Siok from Lockheed Martin Aeronautics Company) were very well received, not just this year, but also the last. Since some of these tutorials are to be placed in our repository (as discussed above), we are going to organize even more tutorial and lecture sessions from faculty at UTD and subject experts of our industry partners. We also intend to try and organize a re-visit (if possible) by some of the REU students from previous years to address, share their experiences with, and provide advice to the students from the next year.

V. CONCLUSIONS

We summarize our experience of organizing an NSF-sponsored REU summer research program in 2010 at UTD. The implementation of the program along with the three research projects, related activities, and the refinements based on the lessons learned from the previous program in 2009 are discussed. We also outline our future plans for the 2011 program to emphasize continual evolution and refinement.

ACKNOWLEDGMENT

We wish to thank the National Science Foundation as well as the School of Engineering and the Computer Science Department at the University of Texas at Dallas, without which this program would not have been possible. We also extend thanks to our industry partners (especially Michael F. Siok from Lockheed Martin Aeronautics Company, David Struble from Raytheon and Thomas Hill from HP/EDS) for their support which greatly contributed to the knowledge that the students gained from the program. Last but certainly not least, we are extremely grateful to the ten undergraduate students whose unwavering enthusiasm and eagerness to excel made this program a wonderful, yet learning, experience.

REFERENCES

1. D. Budgen and J. E. Toamyko, "The SEI curriculum modules and their influence: Norm Giibs' legacy to software engineering education", in The Journal of Systems and Software 75(2005): 55-62, July 2004.
2. N. G. Leveson, "Software Safety", SEI Curriculum Module SEI-CM-6-1.1 (http://www.sei.cmu.edu/reports/87cm006.pdf)
3. The Net-Centric Software & Systems Center: An NSF Industry/ University Cooperative Research Center.
4. The NSF REU Program (http://www.nsf.gov/crssprgm/reu/)
5. The Chicago Manual of Style, 16th edition, 2010
6. The REU Program at the University of Texas at Dallas – Verification and Validation for Software Safety, (http://paris.utdallas.edu/reu)
7. W. E. Wong and V. Debroy, "An Undergraduate Summer Research Program in Software Safety," in *Proceedings of the 23rd Annual IEEE-CS Conference on Software Engineering Education and Training* (CSEE&T), pp. 157-164, Pittsburgh, Pennsylvania, USA, March 2010
8. W. E. Wong**,** V. Debroy, A. Surampudi, H. Kim and M. F. Siok, "Recent Catastrophic Accidents: Investigating How Software Was Responsible," in *Proceedings of the 4th IEEE International Conference on Secure Software Integration and Reliability Improvement* (SSIRI), pp. 14-22, Singapore, June 2010