

Debugging Assistance for Novices: A Video Repository

Beth Simon
Computer Science and Engineering
University of California, San Diego
La Jolla, CA, USA 92093-0404
bsimon@cs.ucsd.edu

Sue Fitzgerald
Computer Science
Metropolitan State University
Saint Paul, MN, USA 55106
sue.fitzgerald@metrostate.edu

Renee McCauley
Computer Science
College of Charleston
Charleston, SC, USA 29424
mccauley@cs.cofc.edu

ABSTRACT

Debugging can be difficult and frustrating for novice programmers. We propose forming a working group to create a set of web-based debugging tutorials that use video clips along with step-by-step written instructions to teach students how to go about debugging common novice errors. Background and motivation for the project as well as the activities and outcomes of the proposed working group are described.

Categories and Subject Descriptors

K.3.2 [Computers and Education]: Computer and Information Science Education

General Terms

Human Factors

Keywords

Debugging, novice programmers, podcast

1. INTRODUCTION

When students debug, what do they do when they “get stuck”? When possible, they turn to their instructor, a lab assistant or perhaps a classmate if such human help is readily available. If it is not, they may email someone for help or search their textbook; no doubt they search the Internet. A Google search for "Java NullPointerException" yields hits for everything from Sun's API documentation to "CodeGuru Forums" and "Blog BugReports". Most of these have reasonable explanations of the error but are beyond a novice programmer's understanding and none really teaches a student how to go about debugging his or her code.

As a result, what could have been an excellent educational opportunity turns into a mis-educative one [4]. Instead of learning proper debugging techniques, the frustrated student learns that programming is difficult, confusing and lonely. This is particularly unfortunate because it is precisely in these "stuck" moments that students are motivated to learn how to debug; when their learning will be contextualized by the program they are

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '04, Month 1–2, 2004, City, State, Country.
Copyright 2004 ACM 1-58113-000-0/00/0004...\$5.00.

coding rather than presented as part of some contrived exercise [2]. It is also when student learning needs to be "scaffolded" by someone who has pedagogical expertise and experience teaching novice students and is likely to understand their common pitfalls and misunderstandings.

Of course, it is not possible to provide human intervention on a 24/7 basis. So, what is the next best thing to being there? We propose to create a video repository of debugging vignettes designed to walk students through the process of locating and fixing common novice programming bugs. When students have middle-of-the-night bug emergencies they will have somewhere to turn that is designed specifically to teach them not just what caused their bug but to show them how to go about understanding, locating, correcting and testing it [6]. Advantages of the videos include their likely appeal to visual learners and the ability to rewind, replay and refer back to them at anytime.

2. BACKGROUND

This project was inspired in part by a lively discussion about debugging at the ICER '06 workshop. Participants emphasized debugging as a key facet of learning to program and lamented that debugging instruction seems to have been largely squeezed out of today's jam packed introductory curriculum. The need for more research on how debugging is learned and how it should be taught, particularly within an object-oriented context, was stressed. It was also suggested that investigating the debugging processes of CS instructors, clearly experts at the unique process of debugging *novice* programs, could yield valuable insights in this area.

Research on novice debugging suggests that direct instruction improves the debugging skills of novice programmers and increases the likelihood they can transfer those skills to other situations [3]. Perkins reports on an approach to help students overcome “fragile knowledge” in which the experimenter first intervened with a general prompt, then gave a hint and finally offered specific advice [7].

There is an active community of researchers focused on using visualization to promote algorithm understanding [8] with newer research focused on web-based animations [5]. Such approaches are designed to appeal to today's highly visual learners.

Although there are some existing texts to support teaching debugging (e.g., [1]), easily accessible, multi-media, web-based materials are more likely to appeal to today's students.

We propose to incorporate a highly visual approach to giving direct instruction in debugging. This will not only provide a valuable resource for students but may also act as a catalyst for future research on debugging by (1) suggesting a set of desirable

novice debugging skills within an OO context and (2) making explicit instructors' processes of debugging novice programs.

3. WORKING GROUP PROJECT

The goal of this working group is to create a set of resources to facilitate the development of effective debugging processes among beginning students. Specifically we seek to:

- Identify and report on key debugging skills and processes that first year computing students should develop.
- Create a web-based set of debugging tutorials which can be used by students in a variety of settings.
- Explore a pair-programming approach to authoring reusable learning objects (i.e., the debugging tutorials).
- Develop preliminary on-line assessment materials such as student surveys and usage metrics to assess the impact of these tutorial materials.

We propose to gather interested CS educators with experience teaching CS1/2 with the goal of sharing community wisdom on commonly needed debugging skills in the first year of computing.

3.1 Pre-Workshop Preparation

Participants will be expected to read background materials pertaining to novice debugging.

Using the background literature review and directed group brainstorming, the WG will identify a set of bugs and debugging processes which are commonly needed by beginning programmers. Participants will meet prior to the conference via tele- or video-conferencing to facilitate this.

Participants will experiment with screen capture/screen recording software such as CamStudio (freeware) or Camtasia.

Participants will bring to the working group two buggy programs (two sorts of bugs) and sample step-by-step scripts for solving these bugs. The scripts will incorporate prompts and hints (as suggested in [7]) designed to scaffold students' learning.

3.2 On-Site Work

WG members must come equipped with laptops, headsets and microphones, and development and debugging environments commonly used by beginning programmers. Some participants, including the WG leaders, will purchase and install the Camtasia screen-shot recording software. At the WG meeting, paired teams will refine and record 2-7 minute screen-shot videos with explanatory voice-overs. These vignettes will demonstrate various debugging processes. The step-by-step written transcripts to accompany each of the recorded vignettes will be refined.

The video tutorials will expose expert thinking as an instructor works to find and fix a bug in a program. Multiple versions of each tutorial will be created using different common novice programming environments. Tips on handling emotions may also be desirable since students often find debugging frustrating.

A pair-programming approach will be used to develop the tutorials; WG participants will work in pairs to debug one another's programs and refine each other's scripts. Participants will also have the opportunity to change partners and get to know and work closely with many others in the group. We expect to learn more about teaching debugging processes from one another through these one-on-one interactions.

3.3 Outcomes

Videos and transcripts will be posted on the web in a debugging techniques repository hosted by the Educational Technology Research Group at the University of California, San Diego. Linking to a tutorial for a particular type of bug will take the user to the written debugging steps with the option of clicking to view the video for each step. This web-based approach will permit us to provide video examples of the same problem in a variety of development environments.

WG members will develop a survey-type assessment module which can be deployed in conjunction with the videos.

The WG paper will include a summary of the categories of bugs generated by the group. We will discuss the process of how we chose appropriate examples of student problems and selected approaches to debugging. The paper will also focus on how we went about collaboratively authoring reusable learning objects.

4. FUTURE WORK

Future work will be based on the assessment results. Existing tutorials may be expanded, refined or improved. WG members as well as others will be encouraged to submit new tutorials for inclusion. Possible enhancements to the site include a FAQ page and a wiki for student input.

5. REFERENCES

- [1] Bartlett, S.L., Ford, A.R., Teorey, T.J. and Tyson, G.S. *Practical Debugging in Java*. Pearson Education, Inc. Upper Saddle River, NJ, 2004.
- [2] Bransford, J.D., Brown, A.L., and Cocking, R.R. (Eds.). *How People Learn: Brain, Mind, Experience, and School*. National Academy Press, Washington, DC, 2000.
- [3] Carver, S. and Risinger, S. Improving Children's Debugging Skills, In *Empirical Studies of Programmers: Second Workshop*, 1987, 147-171.
- [4] Dewey, John. *Experience and Education*. Free Press, New York, 1997 (reprint edition).
- [5] Hundhausen, C. and Brown, J. Personalizing and discussing algorithms within CS1 studio experiences: An observational study. In *Proceedings of the 2005 International Workshop on Computing Education Research (ICER '05)* (Seattle, WA, Oct. 1-2, 2005). ACM Press, New York, NY, 2005, 45-56.
- [6] Katz, I. and Anderson, J. Debugging: An analysis of bug location strategies. *Human-Computer Interaction*, 3, 4 (1987), 351-399.
- [7] Perkins, D., Hancock, C., Hobbs, R., Martin, F., and Simmons, R. Conditions of Learning in Novice Programmers. In *Studying the Novice Programmer*. Soloway and Spohrer (Eds.) Lawrence Erlbaum Associates, 1989, 261-279.
- [8] Wilson, J., Robert Aiken, R., and Irvin Katz, I. Review of animation systems for algorithm understanding. In *Proceedings of the First Conference on Integrating Technology into Computer Science Education (ITiCSE '96)*, (Barcelona, Spain, June 2-6, 1996). ACM Press, New York, NY, 1996, 1-3.

