Chen, W. et al. (Eds.) (2016). Proceedings of the 24th International Conference on Computers in Education. India: Asia-Pacific Society for Computers in Education

How to teach troubleshooting skill to Computer Science undergraduates?

Kavya ALSE[a*]
[a]*Interdisciplinary program in Educational Technology, Indian Institute of Technology - Bombay, India*
*kavyaalse@iitb.ac.in

Abstract: Troubleshooting is an important ability to required by Computer Science undergraduates. The current research work tries to teach troubleshooting as a skill using a technologically enabled smart learning environment. This proposal gives an overview of the literature related to teaching troubleshooting, the instructional strategy of the learning environment and a plan to evaluate the learning environment.

Keywords: Troubleshooting skill, Computer Networks, Smart Learning Environments, Design Based Research

1. Introduction

Troubleshooting is an important ability required by professionals working in IT industry. A Linkedin* search for the keyword 'jobs requiring troubleshooting skill' in India showed 23 thousand result. Troubleshooting is a process which ranges from the identification of problem symptoms to determining and implementing the action required to fix that problem (Schaafstal, Schraagen, & Berlo, 2000). Troubleshooting is an ill-structured problem requiring human troubleshooters to involve in high level cognitive activities like analyzing the behavior of a system, generating multiple hypotheses which are plausible causes for the problem, keeping track of the troubleshooting process etc. (Jonassen, 2010). This proposal talks about the problem of teaching troubleshooting skills to Computer Science undergraduates in the domain of Computer Networks. Then an overview of the solution approach using educational design research methodology is given. The solution approach includes the sub-skills of troubleshooting skill and instructional strategy. A rudimentary idea of how to evaluate the solution and the contributions of the thesis is explained in the end.

1.1 Motivation

Troubleshooting is a part of daily activities of an IT professional. Be it a code developer or network administrator, he/she might have to troubleshoot some system ranging from embedded chips to data centers hosting exabytes of data. And for the novice professionals these are most probably complex systems already setup by others. This aspect of complex technology adds to the ill-structured nature of troubleshooting. My assumption is allowing undergraduate students to work with appropriate complex problems & essential scaffolding will alleviate some of the problems they face in professional lives.

1.2 Scoping the problem

The approach I am taking is to train students with troubleshooting skill, i.e., making them aware of cognitive processes involved in troubleshooting and allowing them to practice these processes in authentic troubleshooting environments. This requires that the students be familiar with concepts & techniques in the domain.

I have chosen 3rd year Computer Science Engineering undergraduate students and the domain of Computer Networks to setup the authentic troubleshooting scenarios. The complex nature of troubleshooting task and authentic problems in Computer Networks would require a technological environment for the students to work with. Also, scaffolds and affordances required by students can be easily provided in a technologically enabled learning environment. This leads to the research goal of teaching troubleshooting skill in the domain of Computer Networks with a smart learning environment.

*www.linkedin.com

The specific research questions are explained in next section.

2. Statement of Thesis/Problem

The broad research question that is being considered is "How to teach troubleshooting skill to computer science undergraduates in computer networks as the domain using a TEL environment?" This leads to more specific research questions like:

i. What does troubleshooting skill consist of? (its sub-skills/competencies)
    a. How to teach each of those sub-skills?
    b. How to evaluate these sub-skills of troubleshooting skill?
ii. What technological features are required to teach troubleshooting in a TEL environment?
iii. How does learning happen when the students interact with the TEL environment?

3. Related Work

Studies on teaching-learning of troubleshooting are reported in the domains of chemical, electrical and mechanical systems (Johnson, 1995; Ross & Orr, 2009; Woods, 2006). These studies report the abstract sub-skills of troubleshooting skill. Another thread of research has been the design and development of expert systems for troubleshooting where knowledge organization and representation models like "structure-behavior-function" and causal maps were developed (Chandrasekaran & Mittal, 1983). Some researchers are interested in the cognitive and meta-cognitive processes of experts and novices with respect to troubleshooting (Johnson, 1987; Reed, 1993; Yen, Wu, & Lin, 2012). The expert studies provide a starting point for designing assessments and learning outcomes. The novice studies help in understanding the learner needs and designing scaffolds.

The tools (gdb, wireshark etc.) which intend to help in program debugging or network troubleshooting are tied very much to the domain. They don't aim to teach the process of troubleshooting explicitly. They will be of more use when the learners understand when & where to use them.

There are very few systems which intend to teach the process of troubleshooting to students. Jonassen (Jonassen, 2010) talk about the architecture of one such system. However, there have been research studies to investigate the technological features that will help students in learning such ill-structured problem solving (Basu, Dickes, Kinnebrew, Sengupta, & Biswas, 2013; Jonassen, 2010). There are studies (Quintana et al., 2009; Xun & Land, 2004) that talk about providing different types of scaffolds for different tasks while teaching to solve an ill-structured problem like scientific inquiry or troubleshooting. Some of these scaffolds are representation that would enhance 'meaning-making', prompts and reflection for metacognitive processes etc.

4. Research Methodology

The research methodology I am using is educational design research (EDR) (McKenney & Reeves, 2014) along with the conceptual framework of TELoTS (Murthy, Iyer, & Mavinkurve). EDR is an iterative method consisting of phases – analysis and exploration of the problem, design and construction of solution, evaluating the solution to verify problem solving. An output of EDR apart from the solution is to produce theories related to solution development. EDR includes the participation of all stakeholders like the researcher, instructor, students etc. in the solution development.

TELoTS stands for technology enhanced learning of thinking skills. The TELoTS framework provides step by step guidelines for developing TEL environments for thinking skills like troubleshooting, considering EDR as a research methodology. The following diagram summarises the steps of TELoTS framework as mapped to EDR.

5. Solution approach

I have synthesized the following sub-skills of troubleshooting. The students are made to practice these sub-skills in different troubleshooting contexts.

5.1 Sub-skills of troubleshooting

The sub-skills of troubleshooting as synthesized from literature are as follows:

a. Problem Space Representation: where students have to represent the problem space in various levels like the structural composition of the system, the function of each component in the system and the connections between each of those components etc.
b. Hypothesis Generation: The students need to generate a number of plausible, testable reasons for the error before they actually go and test the reasons. These reasons are called hypotheses.
c. Hypothesis Prioritization: Once there are multiple hypotheses, the students need to prioritize those hypotheses according to some criterion (like easiest one to test, most probable hypothesis etc.). Then they select the most prioritized hypothesis to test.
d. Design and run test: The students will have to identify the testing means/instruments for the hypothesis that they have chosen. Then they have to predict the result of that test and compare it with the obtained result after the test is done. This comparison is intended to interpret the result and take further decisions. The iteration of generation, prioritization & testing continues till the reason for error is found.

5.2 Instructional Strategy

The system will have students trying to solve simple to complex problems. At the beginning, students will be given an overview of the troubleshooting process consisting of 4 sub-skills. Then students will have to complete tasks corresponding sub-skills of troubleshooting. The tasks have scaffolds related to i) domain concepts (Computer Networks), ii) the process of troubleshooting and iii) reflective prompts intended to aid in metacognitive processes.

The following diagram represents the part of work completed (the blocks with dark blue background) and the part of work (the blocks with light blue & white background):

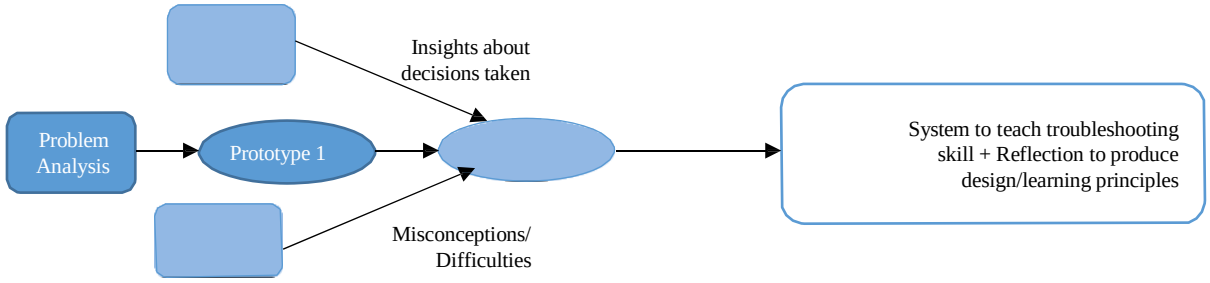

Figure 2. Overview of dissertation status

6. Evaluation Plan

The troubleshooting skill teaching system will be evaluated along the following dimensions:

i. Learning – What do the students learn & How do they learn using the system – using post-test, interview & analyzing student interactions in the system
ii. Evaluating the usability of the system – using perception survey reports from students & interview
iii. Evaluating the perception of learning – using survey reports from students & interview

7. Expected Contributions

I intend that my thesis would result in the following contributions:

i. Sub-skills as applicable for troubleshooting in computer networks and assessment rubrics for

the same

ii. A system to develop troubleshooting skill to computer science undergraduates in the domain of computer networks

iii. A theory which explains how learning happens when students interact with the system.

References

Basu, S., Dickes, A., Kinnebrew, J. S., Sengupta, P., & Biswas, G. (2013). CTSiM: A Computational Thinking Environment for Learning Science through Simulation and Modeling. *The 5th International Conference on Computer Supported Education*.

Chandrasekaran, B., & Mittal, S. (1983). Deep versus compiled knowledge approaches to diagnostic problem-solvingt, 425–436.

Johnson, S. D. (1987). Knowledge and skill differences between expert and novice service technicians on technical troubleshooting tasks. Retrieved from http://eric.ed.gov/?id=ED290043

Johnson, S. D. (1995). Understanding Troubleshooting Styles To Improve Training Methods. Retrieved from http://eric.ed.gov/?id=ED389948

Jonassen, D. H. (2010). *Learning to solve problems: A handbook for designing problem-solving learning environments*.

McKenney, S., & Reeves, T. . (2014). Educational Design Research. In *Handbook of Research on Educational Communications and Technology* (Vol. Springer N, pp. 131–140). http://doi.org/10.1007/978-1-4614-3185-5

Murthy, S., Iyer, S., & Mavinkurve, M. (n.d.). Pedagogical Framework for Developing Thinking Skills using Smart Learning Environments. *(Under Review), IDP-ET, IIT-B*.

Quintana, C., Reiser, B. J., Davis, E. a., Krajcik, J., Fretz, E., Duncan, R. G., … Soloway, E. (2009). A scaffolding design framework for software to support science inquiry. *Journal of the Learning Sciences*, *13*(May 2014), 37–41. http://doi.org/10.1207/s15327809jls1303

Reed, N. E. (1993). Analysis of expert reasoning in hardware diagnosis. *International Journal of Man-Machine Studies*. http://doi.org/10.1006/imms.1993.1012

Ross, C., & Orr, R. R. (2009). Teaching structured troubleshooting: integrating a standard methodology into an information technology program. *Educational Technology Research and Development*, *57*(2), 251–265. http://doi.org/10.1007/s11423-007-9047-4

Schaafstal, A., Schraagen, J. M., & Berlo, M. Van. (2000). Cognitive Task Analysis and Innovation of Training : The Case of Structured Troubleshooting. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *42*(1), 75–86.

Woods, D. D. R. (2006). *Successful trouble shooting for process engineers: a complete course in case studies*. John Wiley & Sons. Retrieved from https://books.google.co.in/books?hl=en&lr=&id=jBGWytm2aFwC&oi=fnd&pg=PR5&dq=Successful+trouble+shooting+for+process+engineers:+a+complete+course+in+case+studies&ots=xXyG4cmNJm&sig=g 6l-XuSVO37JT7BkOVysR3gtBrQ

Xun, G., & Land, S. M. (2004). A conceptual framework for scaffolding III-structured problem-solving processes using question prompts and peer interactions. *Educational Technology Research and Development*, *52*(2), 5–22. http://doi.org/10.1007/BF02504836

Yen, C.-Z., Wu, P.-H., & Lin, C.-F. (2012). Analysis of Experts' and Novices' Thinking Process in Program Debugging. In K. C. Li, F. L. Wang, K. S. Yuen, S. K. S. Cheung, & R. Kwan (Eds.), *Engaging Learners Through Emerging Technologies* (Vol. 302, pp. 122–134). Berlin, Heidelberg: Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-642-31398-1