Interrupted Cognition in an Undergraduate Programming Course

Kay Wijekumar, Ph.D.
Assistant Professor (Corresponding Author)
School of Information Sc & Technology
The Pennsylvania State U. Beaver
Monaca, PA 15061
kxw190@psu.edu

Paul Meidinger
School of Information Sciences and Technology
Penn State Beaver
Monaca, PA 15061

Abstract

Computer and Information Science courses are frequently taught in networked computer laboratories where students have access to advanced development environments, connectivity, communications, and multi-media capabilities. Advanced connectivity and communications can have positive or negative effects on learning. One cause for concern can be interruptions from emails, cell phones, and instant messages. This paper presents a theoretical foundation for Technology Interrupted Cognitive processes with examples of how learners' cognitive schemas maybe positively or negatively affected by the interruptions specifically from instant messages. The paper also presents findings from a research study to document the frequency and prevalence of instant messaging use by students. Results document the approaches that the students participating in the study use and how their cognitive structures are affected by instant messaging interruptions. Finally the paper also recommends guidelines for reducing the negative effects of interruptions on cognitive processes.

Introduction

"As I conduct a Java programming lab, I notice most students are concentrating on writing and debugging their code. Some students appear to be moving their heads constantly between windows on their computer desktop. They appear to be having a conversation with a friend on instant messaging (IM) while working on their program."
– Author

Interruptions are everywhere today. In the workplace, in the classroom, and even while you are driving, cell phones, email, instant messaging, helpful hint pop-up agents, and spam compete for your attention and disrupt your plans. Companies have documented how disruptive and costly interruptions are in the workplace .(Dahms, 1998) Even though there has been research documenting the costs, time (Speier et al., 2003), and social effects of interruptions, there has been no research on the possible effects of interruptions on human cognitive structures. This paper starts by proposing a theoretical framework for "Technology-Related Interrupted Cognition" (TIC) and presents results from a research study designed to identify the outcomes of instant messaging interruptions in a programming course. Finally, recommendations for managing TIC in learning environments are suggested.

Technology Interrupted Cognition

When the student in the java class is concentrating on debugging a program (a complex problem) she is looking at the errors and the source program by shifting her visual attention between those two windows. When she receives an instant message from her friend in a third window, her visual attention is split again. This attention split requires the student to "save" her thinking on the debugging process and move to read the social message. After reading and responding to her friend the student has to return to the debugging process. Here are some possible outcomes of this scenario:

- The interruption may require that the student retrace her debugging steps and that may help her learn more (create more routes to her existing schema).
- The interruption may relieve some cognitive load that had built up due to stressing out about the program errors. She can now start with a cleared working memory!
- The student may be now processing the complex task of debugging with the social conversation, reducing the available working memory to concentrate on the debugging task.
- The interruption may cause the student to take more time to solve the problem.
- The long term memory structure for this debugging experience may be fractured or not complete. Specially, if the student's prior knowledge was very low, this can cause significant breakdowns in the creation of the schemas.

TIC can have positive or negative side-effects in learning environments. Positive side-effects of interrupts like a friend/colleague giving a hint to solve the problem can help re-organize information in long-term memory or create a new cue to existing information (Nardi et al., 2000). Many people seem to remember tasks that were interrupted more than tasks that were not interrupted (Zegarnik, 1927). Students may also be learning to multi-task because of the availability of multiple windows on the desktop. Negative side-effects can be missed sensory signals, fragmented knowledge structures, and poor organization in long-term memory.

The TIC theory is framed by synthesizing interruption theory (Speier et al, 2003), disruption theory, schemas, task variables, and learner variables.

Interruption & Distraction Theories

An interruption is an "externally-generated, randomly occurring, discrete event that breaks continuity of cognitive focus on a primary task" (Corragio 1990.p. 19) in (Speier et al., 2003). Interruptions are usually caused by an external actor and causes the person performing a task to divert their attention for some matter of time to the interrupt. Interruptions usually use the same sensory channels as those used in the primary task and therefore break the concentration on the task. The interruptions can also cause overloading of memory capacity.

Jett & George (2003) suggest that intrusions are unexpected encounters initiated by another person that results in a loss of time and sometimes affects the flow of processing, motivation, and engagement on task. Finally, discrepancy is an internally generated or externally pointed unexpected interruption which causes the person to stop and think, re-arrange their plans and actions, and re-organize their schema.

Based on the theories of interruption and distraction the following are the critical elements of the programming class:

- Length of Interrupt
- Type of Interrupt (Social or Professional)
- Sensory Channel used for Interrupt

IM can cause interruptions of varying length, can be social or professional, and cause overloading of the visual channel. IM is also gaining popularity among college students.

Schemas

“Schemas are organized knowledge structures representing concepts such as situations, objects, events, and actions at various levels of abstractedness. The central functions of schemas are (a) to enable the comprehension of current input and (b) to predict future events” IRumelhart & Ortony, 1977; Schützwohl, 1998). Schemas and scripts created carefully and abstracted to chunks make the information easily retrievable when needed and affects the learner’s ability to use this knowledge in all their activities. When schemas are not retrievable, poorly structured, or not abstracted they cannot be used easily and flexibly in different situations, the learning has not been effective in organizing the information with cues and retrieval paths.

Task Variables

The tasks performed by students in undergraduate programming classes are usually complex and ill-structured. These require the students to process information from multiple sources (algorithm for the program, data structures to be used, syntax of the programming language, interfaces, editors, and errors). As students’ progress from the creation of the algorithm to programming the solution and debugging it they are creating problem schemas and refining their approach. These skills are what the learning environment is focused on. Ideally the learner who has practiced this cycle of problem solving can apply it (the schema) in new situations beyond the classroom.

Learner Variables

Learners exhibit a variety of individual differences. Some that are frequently used in the context of interruptions include prior knowledge, cognitive capacity, and metacognitive skills. High prior knowledge allows learners to generate more pathways to their schemas and also plan and execute their problem solving process more efficiently. These students will be less affected by interruptions in their learning environments. Similarly, students with high memory capacity are able to manage processing multiple requests for their memory resources and be less affected by interruptions. Finally, metacognitive skills allow the learner to plan, monitor, and manage their learning resulting in fewer interruptions to their learning.

Research Questions

1. Do instant messaging interrupts cause students with low prior knowledge to miss important concepts or fragment their schemas?
2. Do instant messaging interrupts cause students with low metacognitive skills to miss important concepts or fragment their schemas?
3. What strategies do students use to overcome interruptions via instant messaging in the learning environment?

Research Design

The purpose of this research was to compare students' performance on programming tasks when they were allowed to use Instant Messaging on their desktop and when they were not. We also wanted to study whether individual differences like prior knowledge and metacognitive skills affected the learning with and without IM interruptions.

Participants & Methods

Twenty eight juniors and seniors enrolled in two sections of a distributed programming course (using Java) were recruited. The participants were given a pre-test on their knowledge of java programming and also completed a questionnaire about their metacognitive skills. The students' use of metacognitive skills was verified through observations of their programming tasks. The average age of the participants was 21 and there were eighteen males and ten females.

Description of the Project

An undergraduate level course on .advanced Java programming was the setting for the study. Students had to have completed a prerequisite course on Java and also at least 3 programming related courses prior to enrolling in this class.

To assess the impact of interruptions IM was disabled on the laboratory computers for the first 5 weeks of the semester (condition 1). Students were informed about this at the beginning of the semester. IM was enabled and students were informed for the second five weeks of the semester (condition 2). Finally, IM was disabled again for the last 5 weeks of the semester (condition 3). At the conclusion of each five week session a post-test and interview were conducted.

The prior knowledge pre-test consisted of a written test and programming task. Students were given specific hints by the instructor during each week of the lecture and laboratory on the questions that will be on the unit tests.

Assessment of learning outcomes was conducted using three short and long answer tests, and interviews with students. The students completed three tests at the 5th, 10th, and 15th weeks of the semester. They also completed a programming project during each of the three phases. The first project was to design a servlet based web interface for a store. The second project was to create a similar solution using enterprise java beans. The last project used an RMI based solution for a similar task. Students worked individually on these projects and used a Linux Operating System, Tomcat Application Server, and Ant Framework. The three exams were similar in format with short and long answer questions. The questions ranged from simple definitions, evaluating and comparing solutions, and filling in lines of code on programs. The exams focused on factual knowledge and problem solving skills.

Observations were conducted in the computer laboratory and interviews were conducted with participants selected based on their scores. Two participants with very high test scores (95 and higher), two participants with very low test scores (< 30) and two participants with scores between 40-60 were interviewed. The interview was designed to gather information about how they managed their IM use.

Results

Standard statistical analysis procedures were used to compare differences among the three conditions. Data analysis was conducted on 24 participants who had completed all projects and tests for the semester.

62.5% of the participants scored less than 60% on the pre-test and were classified as having low prior knowledge. 75% of the participants showed little use of metacognitive skills. In questions like planning and managing their learning, these students did not suggest any concerted effort. They were classified as having low metacognitive skill level. Results from ANOVA with metacognitive skill as the independent variable and the three exams as the dependent variables showed no significant differences in exam performance between the groups. High prior knowledge (pre-test scores) showed significantly higher test scores for the first two exams (exam 1, $p=.000$; exam 2, $p = .001$). The significance for exam 3 was $p = .080$.

Every week students were given hints on possible exam questions. For example, the instructor would state, “a comparison on CGI scripts and Servlets will be on the next exam”. The data for these specific questions on the test showed that all students remembered more of the specific questions on the test when IM was not used in weeks 1-4 and then weeks 10 – 14. A significantly lower ($p = .001$) number of these questions were recalled when IM was used in weeks 5-9.

Students with high prior knowledge were less affected by the IM (during weeks 5-9) use than those whose pre-test scores were lower. Additionally, half the students in the high prior knowledge and high metacognitive skills category chose not to use the IM on their desktop. 50% of the lower metacognitive skill group chose to use the IM on their desktop.

Interviews with students resulted in very interesting findings. First, students with high metacognitive skills said they always turned off the sounds on their IM programs and usually waited till they thought they could take a break to acknowledge an interruption. They also suggested that they maintain two groups of IM buddies, the first for social reasons (that they do not talk to in the class) and a second that could provide them technical assistance with their tasks. This method appears to actually help them solve problems faster and reduce their frustration levels. One student also suggested that he gets to bounce ideas off his technical counterparts and get interesting new ideas through these interactions.

Second, students with lower metacognitive skills uniformly suggested that they break their concentration to attend to conversations on IM. This was also observed during laboratory exercises in programming and debugging. They also suggested that their interruptions did not affect their learning. However, their high metacognitive skilled counterparts were very conscious of the potential interruptions of IM.

Third, during interviews, the two students (with lower metacognitive skills) could not recall the errors they encountered or debugged. On the other hand, their high metacognitive skilled counterparts appeared to thrive on their learning from the errors. They recalled the specific errors, recalled when they had encountered those previously and how they would avoid them in the future.

Fourth, students with low metacognitive skill spent lots of time retracing their steps but did not perform well. When their IM was turned off, the same students remembered more questions

Implications for IM in Learning Environments

Computer Learning environments have taken quantum leaps in speed, connectivity, and tools. While they have afforded positive outcomes, some like instant messaging on the desktop can contribute to disruptions of schemas and learning.

On the positive side, students who manage their IM use by organizing their buddies into social and professional contacts appear to be positively impacted by interruptions on IM. They however, choose when to attend to the external interrupts and initiate their own interrupts to aid in problem solving.

On the negative side, students with low prior knowledge and lower metacognitive skills do not organize their interruptions, and show poorer test performances on specific questions mentioned during class.

Suggestions for Further Inquiry

The study is currently continuing with extensive observations of IM users, IM transcript reviews, and surveys of over 500 undergraduate students on their IM use patterns.

Recommendations for Teachers and Learners

a. Organize contacts in IM so that you can manage your conversations and timing
b. Avoid interacting on IM during class unless it is directly related to your learning
c. If you choose to use IM during class or when concentrating on some task follow your thought process until you can take a break to attend to the IM interruption.
d. Avoid using the sound interrupts on IM to minimize the interruption caused by a message.

Conclusion

The results reported here suggest that IM use can be useful when managed effectively by the learner. Training students to manage their IM use is a critical step that needs significant investment of time and effort by teachers and those concerned about learning.

References

Dahms, A.R. (1988). Time Management and the Knowledge Worker. Industrial Engineering, March – April, 1988, p. 27-29.

Jett, Q.R. & George, J.M. (2003). Work Interrupted: A Closer Look at the Role of Interruptions in Organizational Life. Academy of Management Review. (25 pages).

Nardi, B.A., Whittaker, S. & Bradner, E. (2000). Interaction and Outeraction: Instant Messaging in Action. *Proceedings of the conference on Computer Supported Cooperative Work*, 2000, Philadelphia, PA

Rumelhart, D.E. & Ortony, A. (1977). The representation of knowledge in memory. In R.C. Anderson, R.J. Spiro, & W.E. Montague (Eds.), *School and the acquisition of knowledge* (pp. 99-135). Hillsdale, NJ: Erlbaum.

Schützwohl, A. (1998). Surprise and Schema Strength. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 24, No. 5, 1182-1199.

Speier, C., Vessey, I & Valacich, J.S. (2003). The Effects of Task Interruptions, Task Complexity, and Information Presentation on Computer-Supported Decision-Making Performance. *Decision Sciences*. Volume 34 Issue 4 Page 771. November 2003

Zeigarnik, B. (1927). Das behalten erledigter und underledigter handlungen. Psychologische Forschung, 9, 1-85.