PPIG-7

The Seventh Annual Workshop of the Psychology of Programming Interest Group

Diane H. Sonnenwald

Introduction

The Psychology of Programming Interest Group (PPIG) is a multi-disciplinary network of researchers and practitioners interested in understanding cognitive, social, and educational aspects of programming and design. The group meets yearly to discuss their research progress and issues in an informal, supportive environment. This year's workshop, PPIG-7, was held at the University of Edinburgh, 4-6 January 1995.

The organizers of PPIG-7 were Helen Pain (local organizer), Thomas Green, and Ben du Boulay; students at the University of Edinburgh assisted during the workshop. PPIG-7 was sponsored by the Engineering and Physical Science Research Council.

Work from a variety of countries and disciplines was discussed at the workshop. Presentations were divided into the following themes: Tools and Techniques, Design Strategies, Competence, Knowledge and Learning, Using Diagrams and Graphical Programming Languages, and Perspectives. Following is an overview of the talks presented in each theme. Affiliations and email addresses of authors are provided so readers can contact the authors directly for further information.

Tools and techniques

The first session of the workshop focused on tools and techniques. The session began with Robin Johnson (University of Technology, Papua New Guinea; johnson@maths.unitech.ac.png) reporting on courseware design support. Observing courseware designers select and adapt cases from a software case library, Robin concluded that designers appeared to need support in evaluating the suitability and completeness of their adaptations. John Domingue (Open University, UK; J.B.Domingue@open.uk.ac) spoke about using software visualization technology, e.g., filmcraft, cartoon animations, and graphic design techniques, to display data structures, programs, and algorithms, and the need to support interactions among domain, design task and code visualizations. Margaret Burnett (Oregon State University, USA; burnett@research.cs.orst.edu) demonstrated the visual programming language, Forms/3, that she is developing. Forms/3 is based on a spreadsheet model; in an integrated environment, programmers attach declarative statements to abstract graphical forms, these statements are executed, and changes to the forms displayed.

The tools and technique session continued with a report on maintenance in object-oriented systems. Jos van Hillegersberg, K. Kumar (Erasmus University, the Netherlands; jhillegersberg@fac.fbk.eur.nl), and R. Welke (Georgia State University, USA) found that subjects with 5-10 years experience in structured programming and six months experience in object-oriented programming had more trouble, i.e., were less productive, maintaining object-oriented systems than structured systems. It is important to understand these implications when industry switches from structured to object-oriented techniques. Continuing his work in Prolog, Paul Mulholland (Open University, UK; p.mulholland@open.ac.uk) reported on his most recent study that evaluated the effectiveness of debugging Prolog using a non Byrd Box model, e.g., a choice-point model, of execution for students. Continuing to discuss Prolog, Tom Ormerod (Lancaster University, UK; T.Ormerod@lancaster.ac.uk) and Linden Ball (Derby University, UK) presented a study that evaluated TED, a techniques (standard algorithm) editor, for teaching Prolog. Their results indicated that students performed better when using TED for easy programs but not for hard problems. It appears that the techniques editor is good for coding but perhaps not as good for design.

Design Strategies

The theme of the workshop session that began the following day was "Design Strategies." Willemien Visser (INRIA, France; visser@nuri.inria.fr) began the session with an invited talk titled 'Use of episodic knowledge and information in design problem solving.' Analyzing a video tape of an individual designer working on a design problem, Visser observed that the use of episodic knowledge based on the designer's and other people's experiences played an important role. An open research question is whether designers need assistance in use of, and access to, episodic data. Diane Sonnenwald (Riso National Laboratory, Denmark; dhs@risoe.dk) was the second speaker in this session. Sonnenwald's talk, 'Knowledge exploration in design: Communicating across boundaries' discussed communication roles and strategies that help design team members explore and integrate knowledge during the design process. These roles and strategies are based on her field studies of four design teams in the USA and Europe. The roles span internal and external organizational boundaries, as well as task, discipline, personal and multiple boundaries, and help filter and translate information among design participants from different domains as they come to an understanding of how the artifact will support the user. Next, Steve Lang (Loughborough University, UK) and Tom Ormerod (Lancaster University, UK; T.Ormerod@

lancaster.ac.uk) talked about 'Control strategies used by expert program designers.' Lang and Ormerod studied four expert Prolog programmers as they designed and coded solutions to an enlarged version of the 'signals' problem. They observed that the programmers adopted a 'children-first' approach, maximizing the advantages of breadth- and depth-first approaches.

After a short break, the session continued with a talk by J. Siddiqi (Sheffield Halam University, UK; J.I.Siddiqi@ shu.ac.uk), B. Khazaei (University of Wolverhampton, UK), R. Osborn and C. Roast (both from Sheffield Halam University) titled 'An investigation into strategies employed in solving a programming task using Prolog.' Their work discussed the effects of changing from a procedural to a declarative approach in programming. Studying 32 undergraduate computer science students make the transition to Prolog programming after two years of procedural programming training, they concluded that the choice of data representation and decomposition strategies used appear to be more important than the paradigm used.

Next, David Budgen (db@cs.keele.ac.uk) and Mitch Thompson (both from Keele University, UK) talked about 'Software design using G.O.O.S.E.' G.O.O.S.E. is a generalized object oriented support environment intended to support large-scale design activities; the current prototype has three diagram editors (for functional, behavioral and structural properties), text and audio note editors, a consistency checker, a design 'execution' tool with a 'scenario' editor, and a logging facility. Copies of the prototype that run on a Sun Sparc 2 workstation are available. The talk described plans to conduct experimental studies with G.O.O.S.E. in order to investigate how designers use such a tool and whether it meets its aim of being "method-free" and non-intrusive. The session concluded with David Gilmore (University of Nottingham, UK; dg@psyc.nott.ac.uk) providing a demo of KidSim, a graphical programming/simulation environment developed by Apple Computer, Inc., for 10-13 year old children. It was great fun to see the 'worlds' the children constructed using the tool. It was clear that the children enjoyed programming with KidSim, although, as Gilmore explained in a paper co-authored with Karen Pheasey, Jean Underwood, and Geoffrey Underwood, it's less clear which generalized programming constructs and thinking skills children acquired. For example, few children debugged the rules that they programmed, instead they wrote new rules.

Competence, Knowledge and Learning

After a nutritious lunch, we enjoyed an afternoon that focused on competence, knowledge and learning. To begin, we heard an invited talk by Brian Reiser (Northwestern University, USA; reiser@ils.nwu.edu) titled 'Facilitating the acquisition of mental models of programming with GIL: An integrated planning and debugging learning environment.' GIL (Graphical Instruction in LISP) is an interactive learning environment designed to be reasoning-congruent, i.e., provide a fit with the structure of students' solution plans, makes the structure of solutions more visible, and provide access to invisible program behavior. Reiser also reported on several empirical studies examining the effectiveness of versions of GIL. Next, Linda Carswell (Upper Bann Institute, UK; l.carswell@ulster.ac.uk) presented 'A pilot study on novice Pascal programmers on vocational courses in further education in Northern Ireland.' This study begins to question students' attitudes to programming and the perceived necessity and relationship to mathematical competence, and possible reasons for the gender imbalance in vocational computing courses. The third talk by Jean-Francois Rouet (INRIA, France; rouet@isis.imag.fr), Catherine Deleuze-Dordon and Andre Bisseret (both from INRIA) was titled 'Documentation skills in novice and expert programmers: An empirical comparison.' As part of a broader project to study the use or natural language documentation in design activities, the authors presented their current study that investigated the role of expertise in the production of software program comments. They found several categories of documentation: paraphrases, syntactic and semantic explanations, meta-comments and inferences from labels. Paraphrases were more frequently written by novices, and experts issued more explanations.

The fourth speaker in this session was Lindsey Ford (University of Exeter, UK; lindsey@dcs.exeter.ac.uk). In a talk titled 'A model of programming', Ford reviewed visual programming research and suggested a process plant metaphor may increase programmer's comprehension of large scale programs. The last talk in this session was presented by Jose Canas (delagado@ugr.es), Maria Bajo, Raquel Navarro and Pilar Gonzalvo (all from the University of Granada, Spain). Their work focuses on 'Mental representation and computer use.' When people learn to use a computer, they acquire a mental model of the system; Canas *et al.* are investigating techniques for facilitating the acquisition of effective mental models by novices.

Using Diagrams and Graphical Programming Languages

The session that began the last day of the workshop focused on diagrams and graphical programming languages. Pertti Saariluoma (University of Helsinki, Finland; psa@utu.fi) and Jorma Sajaniemi (University of Joensuu, Finland) opened the session presenting their studies of transforming verbal descriptions into mathematical formulas in spreadsheet calculations. In their studies, they identified three models subjects appeared to be using to transform verbal descriptions into math formulas: direct translation, imagery, and propositional. Judith Good, Richard Cox (University of Edinburgh, UK; judithg@aisb.ed.ac.uk) and Paul Brna (Lancaster University, UK) presented their work on diagrams, asking the question, do diagrams make us smarter? Their work that investigates factors which determine the effectiveness of external representations in visual programming environments is in its early stages and we look forward to their conclusions.

Perspectives

The last session of the workshop was titled 'Perspectives.' Comparing program comprehension in different cultures and different representations is the aim of a collaborative project involving Marion Petre, Blaine Price (Open University, UK; B.A.Price@open.ac.uk), Vikki Fix (University of South Dakota, USA), Jean Scholtz (Portland State University, USA), Susan Wiedenbeck (University of Nebraska, USA), Igor Netesin and Sergey Yershov (Technosoft, Ukraine). This group is comparing differences between 'Western' and 'Eastern' programmers as they use a data flow language, a textual language (C), and a control flow language called R-Technology which was developed in the former Soviet Union in the 70's. This is work in progress and it will be interesting to see future reports on the qualitative, quantitative, and methodological results of the project. The final talk in this session was a report on the psychology of programming in the former Soviet Union presented by Igor Netesin and Sergey Yershov (Technosoft International Software Technology Research Center, Ukraine; wyse.kiev.ua!sergey@aviion.ts.kiev.ua). They mentioned work going on at the Glushkov Institute of Cybernetics of the Ukrainian National Academy of Sciences (Kiev), Technosoft (Kiev), Ukrainian Scientific Center, Lyov University, as well as Moscow State University, Institute of Psychology of Russian Academy of Sciences, the Tver State Software Testing Center, and at branches of the Russian Academy of Sciences in Novosibirsk and Vladivostok.

In conclusion, impressions from the workshop were provided by Malcolm Peltu, a communication consultant who covered the workshop for Computing, a weekly professional newspaper in the UK. (Peltu's impressions may be found in the 9 February 1995 issue, page 39.) In addition, a number of small gifts were awarded to the children who attended and endured the workshop with their parents, the local workshop organizer, winners of word contests announced at the start of the workshop (e.g., the most successful use of the words red and green in a presentation), and attendees from outside the UK. Colorful postcards and maps from the Ukraine were also graciously distributed by Igor Netesin and Sergey Yershov.

Social Activities

Informal discussions inspired by the presentations took place during social activities at the workshop. The setting for these discussions included a local vegetarian cafe where we ate lunch each day, and a local pub and hotel where we ate dinner. The first evening dinner was augmented by skittles, a Scottish version of bowling played with bowling balls without holes. Two local boys picked up the pins we knocked over (which at times was a very small number), and they successfully dodged the balls that were thrown haphazardly! Highlights from the second evening included a whiskey tasting and a wee ceilidh. In addition to delicious food at the ceilidh, we enjoyed a bagpipe performance and did folk dancing, doing our best to follow step-by-step instructions in time to music provided by the talented musicians among us (without analyzing cognitive aspects of our failures and successes in remembering dance step sequences).

Other PPIG Activities

PPIG maintains an electronic mailing list for posting short announcements, requesting references, and occasionally for discussion. The list is currently maintained by Judith Segal (University of Surrey, UK). The group also distributes a usually-quarterly newsletter that contains conference and workshop announcements, conference reports, research notes, and abstracts. To join the list and to receive or submit material for inclusion in the newsletter, send mail to ppig-request@mcs.surrey.ac.uk. PPIG web pages have recently been started by Paola Kathuria and are accessible at http://www.u-net.com/ppig/.

Summary

Several research trends appeared to emerge from the presentations and discussions during the workshop. These include (a) increasing international interest in psychology of programming, (b) development of computer-based tools that support programming activities based on cognitive learning styles, (c) the importance of investigating social, cultural and cognitive aspects of programming, and (d) extension of research studies to include investigation of non-traditional students and children. These trends illustrate the continued relevance and multi-disciplinary nature of this workshop and research area.

I would like to thank the speakers, co-authors, workshop organizers, attendees, and sponsor for facilitating another workshop that spawned many interesting discussions. My apologies in advance for any mistakes I made representing their work in the summary. Thanks to Annelise Mark Pejtersen who provided useful comments on a draft of this report.

About the Author

Diane H. Sonnenwald is a NSF-NATO Postdoc Fellow/NSF International Postdoc Fellow at Riso National Laboratory in Roskilde, Denmark while on leave from the University of North Carolina at Chapel Hill. Her research interests include communication, knowledge exploration and information seeking behavior in design; email: dhs@risoe.dk