Research on Partner Technology

Kevin McGee and Mikael Kindborg
Partner Technologies Group
Department of Computer and Information Science
Linköping University
581 83 Linköping, Sweden
kevmc@ida.liu.se mikki@ida.liu.se
www.ida.liu.se/labs/aslab/partech/

Keywords

Adaptive user-interfaces, intelligent user-interfaces, AI, cognitive AI, cognitive modelling, agents, HCI, end-user programming, programming-language design, programmable interfaces

ABSTRACT

The Partner Technology Group of Linköping University's Human-Centered Systems division focuses on *empowering end-users with intelligent partner technologies*; that is, technology that works in active partnership with people. In this paper we briefly describe our research orientation and summarize some of our work to date.

1. PARTNER TECHNOLOGY RESEARCH

Much of the work in applied AI aims to "make life easier" by reducing or removing the need for human effort and participation. In some ways, our aim is to "make life more difficult": developing intelligent technologies that increase the number and kind of challenging activities people find most appealing and important – and that also empower people to more actively engage and participate in "the good, hard stuff." Such activities can include playing games, creating works of art, developing computer programs, maintaining social contacts, or conducting scientific research. Thus, our group works to develop different partner technologies for entertainment, learning, and research.

It is difficult to say precisely what makes "good partners" – but most of us have had the experience of working with them. When we work with good partners, both the product and the process are more effective and enjoyable. Our goal with partner technologies is to develop systems that are similarly empowering: that actively help people engage in and enjoy the activities they find interesting, entertaining, or otherwise important.

Although partners may take on different roles or tasks at different times, in good partnerships all the partners contribute equally in some significant sense. Thus, the notion of *partnership* signifies something more than simply delegating some aspect(s) of a task to a computational system – or even having that system "tell users what to do." This issue is obviously complex, but the example of *end-user programming* highlights some of our concerns. Much of the work on end-user programming attempts to reconceive different activities so that non-programmers are able to use the power and expressiveness of programming within those activities. Thus, participating in some activity can become a form of interactive programming, broadly defined. One aspect of this is the nature of the *development tools* or environments – and the degree to which they can be supportive, even *pro-active*, in collaborating with users on different tasks and goals. It is in this sense that developers of certain AI-based design and development systems seriously suggest that these systems are co-designers or co-authors of works created with them. Of course, such claims are controversial, and it is not our purpose to argue for (or against) them; rather, we merely wish to indicate a class of system where the term "partner" seems more appropriate than "servant," "tutor," or "advisor."

Our work on partner technologies involves a combination of research in cognitive science, AI, and the design of advanced interfaces, games/entertainment, and programming languages. In order to develop partner technologies, we study examples of good human partnerships, develop computational models of partnership, implement prototype partner technologies, and study their usage – which in turn suggests further insights into cognitive aspects of partnership. Thus, our work on cognitive models of partnership lead to implementations of those models as well as partner applications – and our studies of the models and use of the applications lead to further cognitive questions and insights.

The remainder of this paper briefly highlights some of our work on these topics.

1.1 Partner Applications

In our work to date, we have developed a number of prototype applications.

These include work on *concurrent comics* [4] – a programming environment for interactive comics that extends various ideas from constraint-based programming and graphical rewrite systems. One result of this work is *ComiKit*, a programming tool for children being developed as a commercial product; a preview release is planned for late spring, 2005.

Ongoing work in this area includes a masters' project on *ActionCards*, an experimental visual programming system based on behaviors that is a variation on the comic-strip style of programming. We are also planning to develop visual programming techniques that will allow children and other users to program their own artificial life characters, develop and control avatars for online worlds, or model animal behavior in an ecological system. We have a collaboration on programming tools for children with Ken Kahn at University of London, Institute of Education.

Related to this is our work to understand some of the

SAIS-SSLS 2005
Artificial Intelligence and Learning Systems
12-14 April 2005, Västerås (Sweden)
.

ways different programming metaphors/paradigms inform both the implementation and the use of programming languages. One masters' thesis on *Schemorphic* [10] explored the relationship between an interactive language (Scheme) and *batch*-oriented models/implementations of graphics libraries. The project implemented a graphics library for Scheme, based on the Morphic model originated in Self; the result is an interactive development environment that allows graphical runtime programming and debugging, very much in the same spirit as the interactive programming and debugging of non-graphical programs in interpreted environments.

Two earlier masters theses – on *CrawLogo* [9] and on *DocPlayer* [1] – explored related issues. *CrawLogo* is an end-user programming environment, inspired by the classic Logo Turtle, for creating web-enabled games and applications. *DocPlayer* involved the development of an alternative to the standard "desktop" interface, where the file-system is treated as a database – and the interface is analogous to a media-player, making it easy to create groupings and views ("playlists") for documents and other files. One current project involves the development of an interactive language for the development of *interactive fiction* that will allow readers/players of such fiction to have more direct *programmatic* control over different aspects of the stories they read/play.

We have also developed some small prototype applications to explore aspects of "co-improvisation" in different kinds of partnerships, such as "collaborative painting," pilot/plane control systems, and the like [7]. There is also a current project in this spirit, developing a system for musical composition based on concepts from artificial life.

1.2 Partner Modeling

Part of our work to date includes highly idealized simulations of partnership, inspired by different concepts from the life sciences, such as *symbiosis*, *structural-coupling*, *co-evolution*, and *co-adaptation*. So, for example, we are currently experimenting with some partner-simulations loosely inspired by earlier work on Braitenberg Vehicles. Another project, being done in cooperation with Dr. Kumiko Tanaka-Ishii at the Language Informatics Laboratory of Tokyo University, is exploring aspects of partnership in terms of cellular dynamics (and certain aspects of that project may also form the basis of an end-user application). See also [12] for one example of extending this kind of work to the domain of human-in-the-loop systems.

Another area of interest is computational models of cognitive empowerment. That is, there are many well-known mental heuristics, techniques, and strategies that help people perform different tasks (mnemonics to improve memory, for example). But what are some computational cognitive explanations for *how* they help people? We are currently initiating two small projects to study this issue, with a focus on individual empowerment. One longer-term goal of this work is to develop better models and theories about how *partners* – human and otherwise – can empower each other.

1.3 Partner Studies

Our studies of cognitive issues related to partnership include studies of children using the visual language for programming concurrent comics [3, 2], a study of avatar characteristics that influence their appeal in computer-based role-playing games [11], and brief studies as part of developing *CrawLogo*, *DocPlayer*, and the like.

Similarly, we have done some studies on alternative interfaces [6, 5], looking at how people use physical "contact expressions" in their daily lives – and various ways these can be leveraged in the development of devices with contact-expressive interfaces. This initial work has been used to initiate a study by Sweden's *Hjälpmedelsinstitutet* (Swedish Handicap Institute) on possible uses and improvements of robotic pets for people with various forms of debilitating dementia.

Finally, we have also done some work to understand partnership-in-the-large, so to speak, by studying aspects of the *gifting* phenomenon in online communities [8].

As part of the work on the *ActionCards* system, in the spring of 2005 we will study how various kinds of conditional-behaviors are understood by users – as well as issues of partner dynamics in group programming projects.

References

[1] Jody Foo. Docplayer: Design insights from applying the non-hierarchical media-player model to document management. Masters thesis, Department of Computer Science, Linköping University, Sweden, Linköping, Sweden, December 2003.

[2] Mikael Kindborg. How children understand concurrent comics: experiences from lo-fi and hi-fi prototypes. Stresa, Italy, September 5–7 2001.

[3] Mikael Kindborg. Comics, programming, children, and narratives. In *Proceedings of Interaction Design and Children*, Eindhoven, Holland, August 28-29 2002.

[4] Mikael Kindborg. *Concurrent Comics: programming of social agents by children*. Phd thesis, Department of Computer Science, Linköping University, Linkoping, Sweden, May 2004.

[5] Kevin McGee. A touch of the future: contact-expressive devices. *IEEE MultiMedia*, 11(1), January-March 2004.

[6] Kevin McGee and Annika Harup. Contact expressions for touching technologies. In *COSIGN 2003: Procedings of the 3rd Conference on Computational Semiotics in Games and New Media*, pages 68–78, 10–12 September 2003.

[7] Kevin McGee and Johan Hedborg. Partner technologies. In *COSIGN 2004: Procedings of the 4th Conference on Computational Semiotics in Games and New Media*, pages 87–96, 14–16 September 2004.

[8] Kevin McGee and Jörgen Skågeby. Gifting technologies. *First Monday*, 9(12), December 2004.

[9] Johan Nilsson. Crawlogo: An experiment in end-user programming for web-enabled applications. Masters thesis, Department of Computer Science, Linköping University, Linköping, Sweden, December 2003.

[10] Tord Svensson. Schemorphic: A morphic-like graphics system for scheme. Masters thesis, Department of Computer Science, Linköping University, Linkoping, Sweden, May 2004.

[11] Sara Säfvenberg. Socially intelligent agents: designing for appeal and relationships. Masters thesis, Department of Computer Science, Linköping University, Linkoping, Sweden, May 2004.

[12] Rogier Woltjer and Kevin McGee. A joint subsumption architecture as a framework to address human machine function allocation. In *Proceedings of SIMSAFE 2004: Improving Public Safety through Modelling and Simulation*, Karlskoga, Sweden, June 15–17 2004.