



Supporting Collaborative Design With Representations for Mutual Understanding

Jonathan Ostwald

Department of Computer Science and Institute of Cognitive Science
University of Colorado
Boulder, CO USA 80309-0430
E-mail: ostwald@cs.colorado.edu

© [ACM](#)

Abstract

This paper describes a research effort to investigate cross-cultural collaboration in software development. The work is based on a model of collaborative design that calls for stakeholders (including developers and end-users) to iteratively construct an understanding of design problems and potential solutions through the construction and refinement of design representations. The Evolving Artifact (EVA) software design environment has been implemented to support this process. EVA has been used in a development project in a regional telephone company. A case study of this project will be analyzed to yield guidelines and design principles for constructing *representations for mutual understanding*.

Keywords:

collaborative design, workplace cultures, work-oriented design, software development, EVA, functional objects, prototyping

Introduction

My research views development of work-oriented [3] software as a collaborative design task between software developers and clients (i.e., end-users). Developers and end-users belong to different workplace cultures [2], each of which depend on shared representations, norms, rules, and languages. While shared conventions of meaning permit a conceptual coordination within a given culture, they also form boundaries that can hamper communication across different cultures. The essential challenges for cross-cultural collaborative software development are to transcend the respective cultural boundaries and to achieve a growing understanding. The two main questions addressed by my research are:

- 1) What kind of design representations can support cross-cultural collaborative design of work-oriented systems?
- 2) How can a growing understanding be represented in a software artifact?

RELATED APPROACHES

Techniques and methods for large-scale, contract-based software development are not oriented toward user involvement [4], and therefore are not appropriate to support the design of highly interactive, work-oriented systems. These approaches assume that analysts or other user representatives can produce requirements documents that precisely and completely reflect the needs of end-users. However, this is an unreasonable assumption because end-users do not always know their needs, and because their task knowledge is often tacit [8]. Furthermore, end-users are not always aware of the available technological options [5].

Participatory design approaches aim to increase usability of systems through collaboration between developers and end-users [1]. My work shares this goal. In particular, I focus on *tool support* to ground collaboration with computational design representations, to drive an evolutionary design process with the communication that takes place around the representations, and to capture the communication in an evolving artifact along with the representations themselves.

MODELING COLLABORATIVE DESIGN

The action-reflection-critique model [6] is proposed as a model for collaborative design. The model synthesizes design theory with theories about how people collaborate to solve problems. Design theory suggests that designers solve problems incrementally by creating explicit design representations which "talk back" to the designer [9], revealing new understandings of the problem. Human-human collaborative problem solving theory suggests that people construct and maintain an understanding through dialog, in which meanings are accrued incrementally, along with evidence of what has been understood so far. The action-reflection-critique model calls for stakeholders to incrementally develop an understanding of design problems and potential solutions through construction of external design representations that ground communication of design intent and design critiques. Mutual understanding is constructed through creation, discussion and modification of design representations.

REPRESENTATIONS FOR MUTUAL UNDERSTANDING

Communication of design ideas through external design representations is difficult when the communication partners come from different workplace cultures. Bødker and Pedersen [2] describe workplace cultures as systems of meanings "hidden behind or in the various artifacts, symbols, work routines, and established patterns of cooperation." Importantly, meanings are not an intrinsic property of artifacts and symbols. Instead, they are ascribed by the people who share the culture. Design representations must therefore be meaningful across cultural boundaries to talk back to all stakeholders and to foster a growing understanding of the design issues.

System documentation is often not meaningful to end-users because it is abstract and tends to be written in the developer's language. Prototypes, on the other hand, can be meaningful across cultural boundaries because they are concrete design representations that are interpretable by both developers and end-users. When prototypes and documentation are combined, the context for understanding each is increased. Prototypes clarify and instantiate documentation, while documentation provides context for interpreting prototypes.

THE EVA ENVIRONMENT

The EVA (for Evolving Artifact) development environment [7] has been implemented as a part of this research to support the construction, integration, and evolution of representations for mutual understanding. EVA supports the creation of a novel type of software artifact that integrates hypertext-based documentation (including system requirements, design rationale, usage scenarios, user critiques) with prototypes. Hypertext is characterized by mouse-sensitive links that allow navigation from one piece of text to another. The EVA environment provides an additional type of link that, when selected, causes a prototype to appear on the screen (in effect, navigating to the prototype). When the user is finished interacting with the prototype, it disappears from the screen and the user is once again in the hypertext.

Using EVA, all stakeholders can create hypertext-based documentation describing the application domain and system requirements. Developers create prototypes and embed them within the context of appropriate documentation. End-users can then access and interact with the prototypes to experience what the final system might be like, and record their comments and critiques in the hypertext. Documentation and prototypes are then modified accordingly and the iterative process continues. The system design process is thus driven by communication between developers and end-users, and grounded by the evolving software artifact.

FUTURE WORK

EVA has been used to develop an application prototype in a regional telephone company. This case study will be analyzed to yield guidelines and design principles for constructing design representations for mutual understanding. Of particular interest are:

- What are the limitations and challenges of an evolutionary approach to collaborative software development?
- What tool support is necessary to help end-users and developers to better deal with these challenges?

References

1. Communications of the ACM. Special Issue on Participatory Design, 36, 4 (June 1993).
2. K. Bødker, J. Pedersen. Workplace Cultures: Looking at Artifacts, Symbols and Practices. In J. Greenbaum, M. Kyng (Eds.), *Design at Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1991, Chap. 6, pp. 121-136.
3. P. Ehn. *Work-Oriented Design of Computer Artifacts*. Almquist & Wiksell International, Stockholm, Sweden, 1988.
4. J. Grudin. Interactive Systems: Bridging the Gaps Between Developers and Users. *Computer* 24, 4 (April 1991), 59-69.
5. F. Kensing, A. Munk-Madsen. PD: Structure in the Toolbox. *Communications of the ACM* 36, 4 (June 1993), 78-85.
6. K. Nakakoji, G. Fischer. Intertwining Knowledge Delivery and Elicitation: A Process Model for Human-Computer Collaboration in Design. *Knowledge-Based Systems Journal* (1995). (forthcoming).
7. J. Ostwald, B. Burns, A. Morch. The Evolving Artifact Approach to System Building. Working Notes of the AAAI 1992 Workshop on Design Rationale Capture and Use, AAAI, San Jose, CA, July, 1992, pp. 207-214.
8. M. Polanyi. *The Tacit Dimension*. Doubleday, Garden City, NY, 1966.
9. D. Schoen. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, New York, 1983.