

The Integrated User-Support Environment (IN-USE) Group at USC / ISI

Robert Neches,
Peter Aberg, David Benjamin, Brian Harp, Liyi Hu, Ping Luo, Roberto Moriyón, Pedro Szekely

USC / Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292 USA
Email: Neches@isi.edu

INTRODUCTION

Integrated user support environments are individual and cooperative-work systems which allow their users to perform a large quantity of their daily work on-line, and which do so by providing access to a comprehensive set of tools that interact smoothly with each other and present a uniform interface to the users. The INtegrated User-Support Environments (IN-USE) Group is developing a framework for facilitating construction of such systems. The framework is oriented toward assisting users who must timeshare between multiple, highly information-intensive data analysis and problem solving tasks. Our fundamental goals are to help developers quickly assemble support environments that offer reasonable default appearance and behavior, and to make it easy to then customize those environments as needed.

To this end, our research focuses on developing a set of principled facilities that can operate upon a shared declarative model. Application developers work by extending the model to describe their domain. This allows them to get initial versions of their system by inheriting default versions of tools, and lets them specialize the system by refinement rather than by programming "from scratch." Our current efforts to develop the framework focus on four key facilities:

- HUMANOID: a user interface development environment
- BACKBORD: a browsing and clarification aid
- Scenarios/Agendas: a multi-agent activity manager
- TINT: a tool for handling semistructured information

HUMANOID and MASTERMIND (Szekely, Luo, Moriyón, Neches)

HUMANOID's approach to interface design lets designers express abstract conceptualizations in an executable form, allowing designers to experiment with interactive behavior

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

even before the system model is completely stated [7]. This lets designers get an executable version of their design quickly, experiment with it in action, and then repeat the process after adding only the details needed to extend it along the particular dimension currently of interest to them.

HUMANOID is a model-based system: interfaces are specified by constructing a declarative model of how the interface should look and behave. A run-time support module allows applications to execute the model, constructing displays and interpreting input according to the information in the model. HUMANOID provides a declarative modeling language that models the functional capabilities of an application as a set of objects and operations; it further partitions the model of the style and requirements of the interface into four additional semi-independent dimensions: *Presentation, Behavior, Dialogue Sequencing, and Action Side-effects*.

In the spirit of the DARPA Knowledge Sharing Effort approach to sharing and reuse of knowledge-based systems [5], we have been collaborating with Jim Foley's group at Georgia Tech on developing a joint model of interface design. This shared model, called MASTERMIND [6], will enable integration of HUMANOID's design environment with UIDE's design critics, as well as integrating HUMANOID's run-time capability for context-sensitive presentation with UIDE's animated help facilities.

BACKBORD: A specification by reformulation shell (Aberg, Neches)

The BACKBORD system (Browsing Aid Complementing Knowledge Bases OR Databases) is our implementation of the specification by reformulation paradigm [8]. By this, we mean a paradigm of human/computer interaction in which the user develops, by successive approximations, a specification of the objects a system is to manipulate and/or the behavior it is to evince. In this style of interaction, the system provides an environment which facilitates the refinement of the specification, largely by generating feedback for the user about the specification in its current form and by providing guidance about means for modifying that specification. Our research focuses on packaging this paradigm in a library of reusable components to facilitate instantiating this paradigm in a wide range of applications.

Scenarios/Agendas (Benjamin, Szekely, Neches)

Scenarios [4] are program-like descriptions of the sub-tasks that compose extended tasks. They have been used extensively in our logistics applications, both to perform many tasks automatically and to interactively guide users through those activities not amenable to automation.

Scenarios are like very high-level procedures, in that they describe a sequence of steps to be performed. Unlike a procedure, a scenario attempts to capture the processes or sequences of tasks that are related by user-level concerns in performing an activity. Also, a scenario imposes only orderings between steps that are necessitated by dependencies between them. Scenarios operate by attaching the sub-tasks to assorted Agendas, each of which represents the completed and pending tasks of some agent that participates in the task. Scenarios and Agendas provide external memory for "unfinished business," reducing burdens on users' memories.

TINT: Aids for Integrating Informal and Formal Knowledge Representations (Harp, Hu, Neches)

TINT (The Intelligent Note-Taker) acts as a user interface which allows users to enter information into the computer in the form of "semi-structured" notes that are attached to concepts in a knowledge base [2]. Both user and system can create and retrieve notes; they can use notes to exchange information and to store it for their own use. Even though the computer cannot fully understand some user-supplied notes, it can still operate with at least partial understanding, because the notes have some structure and are associated with knowledge base entries that the computer can be programmed to understand. This gives the computer a way of knowing about things it can't handle and asking for help. This reduces the brittleness of conventional expert systems, pointing the way towards a more collaborative style of human/computer interaction. TINT also lets developers do "incremental knowledge acquisition," using user notes to set targets for adding additional automation to a system.

APPLICATIONS

These tools have been put to practical use in a number of prototype application systems. These applications have ranged from logistical analysis domains (BEAMER and DRAMA) to collaborative environments for team development of knowledge bases (SHELTER), under funding from the Defense Logistics Agency, Air Force Logistics Command, and DARPA. DRAMA and SHELTER are both good examples of ISI's tradition of building useful tools and applying them to solve real problems.

DRAMA

DRAMA [1] is an intelligent data review and monitoring system for the Defense Logistics Agency. DRAMA monitors evolving databases containing weapon system design information, coordinating that design information with support requests from the services and knowledge about status of supply and procurement activities within DLA. Its function is to save DLA substantial amounts of money by

ensuring that purchasing and stock management plans rapidly adapt to changes.

SHELTER

The SHELTER development environment for ontologies and knowledge bases [3] is another key application of the IN-USE framework. SHELTER provides an integrated solution to problems in managing knowledge-base size and complexity, and in managing the size and complexity of the activities required to correctly use, modify, or extend such knowledge bases. It facilitates sharing and reuse by helping developers find candidate material to reuse, and ensuring that it is used properly. There are two key ideas in SHELTER. One is an interaction paradigm that encourages reuse of specifications, embodied in a set of browsing and retrieval tools provided by BACKBORD. The other is a set of methods for helping knowledge-base system builders record design rationale metaknowledge, using structured notes from TINT which the system can interpret to assist developers in ensuring the appropriateness of later modifications.

SELECTED REFERENCES

- [1] B. Harp, P. Aberg, D. Benjamin, R. Neches, P. Szekely. *DRAMA: an Application of a Logistics Shell*. In *Proceedings of the Annual Conference on AI and Logistics*, Williamsburg, VA, March, 1992.
- [2] B. Harp and R. Neches, *NOTECARDS: An Everyday Tool for Aiding in Complex Tasks*. ISI Research Report RS-88-204, March, 1988.
- [3] R. Neches, *Cognitive Issues in the SHELTER Knowledge Base Development Environment*. AAAI Spring Symposium on Cognitive Issues in Knowledge Acquisition, March, 1992.
- [4] R. Neches, D. Benjamin, J. Granacki, B. Harp, and P. Szekely. *Scenarios/Agendas: A Reusable, Customizable Approach to User-System Collaboration in Complex Activities*. ISI Working Paper, 1991.
- [5] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W.R. Swartout. *Enabling Technology for Knowledge Sharing*. *AI Magazine*, Vol. 12, No. 3, 1991, pp.36-56.
- [6] R. Neches, J. Foley, P. Szekely, P. Sukaviriya, P. Luo, S. Kovacevic, and S. Hudson. *Knowledgeable Development Environments Using Shared Design Models*. In *Proceedings of the 1993 International Workshop on Intelligent User Interfaces*, Jan., 1993.
- [7] P. Szekely, P. Luo, and R. Neches. *Facilitating the Exploration of Interface Design Alternatives: The HUMANOID Model of Interface Design*. In *Proceedings of CHI'92*, May 1992, pp. 507-515.
- [8] J. Yen, R. Neches, M. DeBellis, P. Szekely, and P. Aberg. *BACKBORD: An Implementation of Specification by Reformulation*. In J.S. Sullivan and S.W. Tyler (Eds.), *Intelligent User Interfaces*, pp. 421-444. ACM Press, 1991.