[Bio follows Proposal]

Bob Balzer

Grand Research Challenges Conference

Computer usage has evolved from small special purpose applications to large Commercial Off The Shelf Software (COTS) products that dominate the landscape. These COTS products present major challenges for our traditional software assistance paradigm.

User's conduct a great deal of activity within such products. To assist them, that activity must be observed and understood and the assistance delivered non-obtrusively through the interface being used by the user.

Respective examples of good and bad assistance within the same COTS product -- Microsoft Word -- are Autocorrect (which is unobtrusive and almost always correct) and Office Assistant (which is obtrusive and often wrong).

The challenge for the research community is finding generic ways to observe and understand user behavior within COTS tools, ways of inferring what the user is trying to accomplish, determining how to accomplish those goals within the COTS tool(s), and either offering suggestions to the user or automating some of the required steps for the user.

A big step forward towards the feasibility of this challenge has been the recent real-time availability to third-party "plug-in" modules of application-level user action histories. These user action histories provide real-time visibility into the user's behavior within a COTS tool and are produced by either an event system provided by the tool vendor (e.g. Visio and Rational Rose) or by a "GUI Monitor" that wraps the user interface to identify low-level gestures within the user interface, translates those into application level actions, and makes them visible by issuing pseudoevents on behalf of the COTS vendor (e.g. Microsoft Word and PowerPoint, and Mathworks Matlab).

Intelligent Personal Digital Assistant

A closely related challenge is assisting the user at a higher level by

integrating the capabilities of several COTS tools. The Intelligent Personal Digital Assistant (IPDA) must extract and fuse information from several different COTS tools to determine what the user is trying to accomplish and what their priorities and deadlines are.

It must be able to assemble, summarize, and present the information that the user needs for those tasks. To do this, it must not only have generic knowledge of how to gather, integrate, and present information, but also domain-specific knowledge of the particular domain(s) in which the user works so that it can process the information being assembled and organize it in ways that are familiar and helpful to those working in the domain.

The second major task for the IPDA is to automate the actions taken by the user in response to the presented information so that the user merely needs to indicate what response needs to be taken and IPDA performs all the necessary actions on behalf of the user (including the supervision required to make sure those actions are completed as planned).

Examples include formatting information for several different databases or systems and propagating it to them to maintain information consistency across those systems, routing documents, gathering signatures or votes, and invoking tools (e.g. to place orders or cancel a trip).

It should be noted that this IPDA automation task is the modern equivalent of the 1980's automatic programming problem. Today's users manually invoke operations in an individual COTS tool through that tool's graphic user interface. Almost all of those COTS tools have programmatic interfaces (i.e. APIs) through which those same operations could be invoked. What is required is a universal "process" language for assembling these COTS operations into programs for the user and an intelligent assistant that will develop, invoke, monitor, and manage those "process programs" for the user.

ROBERT BALZER - BIO

Dr. Robert Balzer received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from the Carnegie Institute of Technology, Pittsburgh, Pennsylvania, in 1964, 1965, and 1966, respectively.

After several years at the Rand Corporation, he left to help form the University of Southern California's Information Sciences Institute (USC-ISI) where he served as Director of ISI's Software Sciences Division and Professor of Computer Science at USC. In 2000 he joined Teknowledge Corporation as their Chief Technical Officer and Director of their Distributed Systems Unit. The Distributed Systems Unit combines artificial intelligence, database, and software engineering techniques to automate the software development process. Current research includes wrapping COTS products to provide safe and secure execution environments, extend their functionality, and integrate them together; instrumenting software architectures; and generating systems from domain specific specifications.

He has served as chairman of SIGART (The ACM Special Interest Group on Artificial Intelligence); as program chairman for the First National Conference of the American Association for Artificial Intelligence; chairman of the Los Angeles Organizing Committee for the 1985 International Joint Conference on Artificial Intelligence (IJCAI); program co-chair for the 9th International Conference on Software Engineering, held in the Spring of 1987 in Monterey, California, and whose theme was “Formalizing and Automating the Software Process,'' and co-chair for the 1st Intelligent Information Systems Workshop, held at Niagara-on-the-Lake, Ontario, Canada, April 1991. Elected AAAI (American Association for Artificial Intelligence) Fellow, 1993.