Work-centered Infomediary Layer (WIL): An architecture for adaptive interfaces

Wayne Zachary and Benjamin Bell

CHI Systems, Inc.
716 N. Bethlehem Pike, Lower Gwynedd, PA 19002
wzachary@chisystems.com; bbell@chisystems.com

Abstract

Active interfaces that act as infomediaries between underlying systems and their human users offer a rich and comprehensive solution to the pervasive problem of (lack of) usability in computer-based work environments. However, they also pose a substantial technological challenge. The paper describes a novel work-centered active interface architecture, called the Work-centered Infomediary Layer (WIL). This architecture allows interfaces to actively identify and offer highly customized work support to the system user without employing any intent inference mechanism. Instead, WIL relies on its ability to build and maintain an internal representation of the work context, and to offer support to the user based on their common relationship to the underlying work situation.

Introduction

A long-standing problem in information systems has been the difficulty in focusing the human-computer interface on the work needs of the human user instead of the functionality of the tools, databases, etc., of which the system is comprised. User interfaces that are system-centered have repeatedly been shown to be difficult to learn and use, and often act as impediments to effective utilization of the underlying system capabilities. Given the premise that the information system (in the broadest sense of the term) is there to facilitate human work, the user interface should instead be work centered. That is, it should be organized around the work that the person is trying to accomplish. While this organization can be passive (i.e., simply reflecting and conveying the work-domain concepts and principles, as suggested by the work of Vicente & Rasmussen, 1992), more extensive work-centering can be accomplished by allowing the interface to actively translate and transform the underlying capabilities of the information system into the user s work-centered terms and representations. This perspective sees the user interface as an active component of the whole system, not merely a porthole into a software black box.

We have been conducting research into this paradigm of active work-centered interfaces for a number of years. This line of research has resulted in the development of several different active interface systems, all designed to support work in different specific domains. Most recently, these various applications have been abstracted to create a novel work-centered interface reference architecture, called the Work-centered Infomediary Layer (WIL), along with an associated design methodology.

The central feature of the WIL concept is an explicit representation of context. The notion of context in WIL is broad, including the overall dynamic environment, the work setting, the underlying infosphere, what the WIL user is doing (as well as what other people in the work environment are doing (to the degree that information is accessible to the WIL), and even the WIL interface s own internal context (i.e., awareness of what it is doing, trying to do, has already done, etc.). Context is critical because work actions, decision-making, and information needs are strongly based on an internal representation of the current work context. This suggests that the work needs and goals of the user of WIL (or any interface) are frequently understandable only within the local work context in which they arise.

A key architectural issue in WIL is how this context-based understanding occurs. When people collaborate in a work context, they share much of their internal context representation, and use that shared representation to interpret each other s actions and behave collaboratively (see various papers in Robertson, et al., 1990; Eisenberg, 1990). Importantly, though, the sharing of these context representations is implicit rather than explicit; that is, people do not communicate about context, but rather use interpretation and assessment knowledge that is presumed to be shared in a tacit way to build generally compatible representations of the context. (An exception to this is when people become aware that they do not share a common understanding of the situation, and then often engage in communicative acts with the goal of repairing the inconsistent understanding). A corollary of this situation-based action viewpoint is that it explains how people are usually able to cooperate without complex explicit processes such as intent inferencing that have been the focus of much active interface research (e.g., Jones, 1988; Vaubel & Gettys, 1990; Bell & Santos, 2001). Rather, people use their shared relationship to a common context (and presumed shared understanding of it) to reason about what needs to be done (see Suchman, 1987) and cooperate on the basis of that. This is the principle on which a WIL interface operates, incorporating knowledge

about the work domain and using it actively to build and maintain an understanding of the work domain and the current context within the work environment (which includes the human user). The context is then used to construct various forms of work-centered support, which are customized to the current situation.

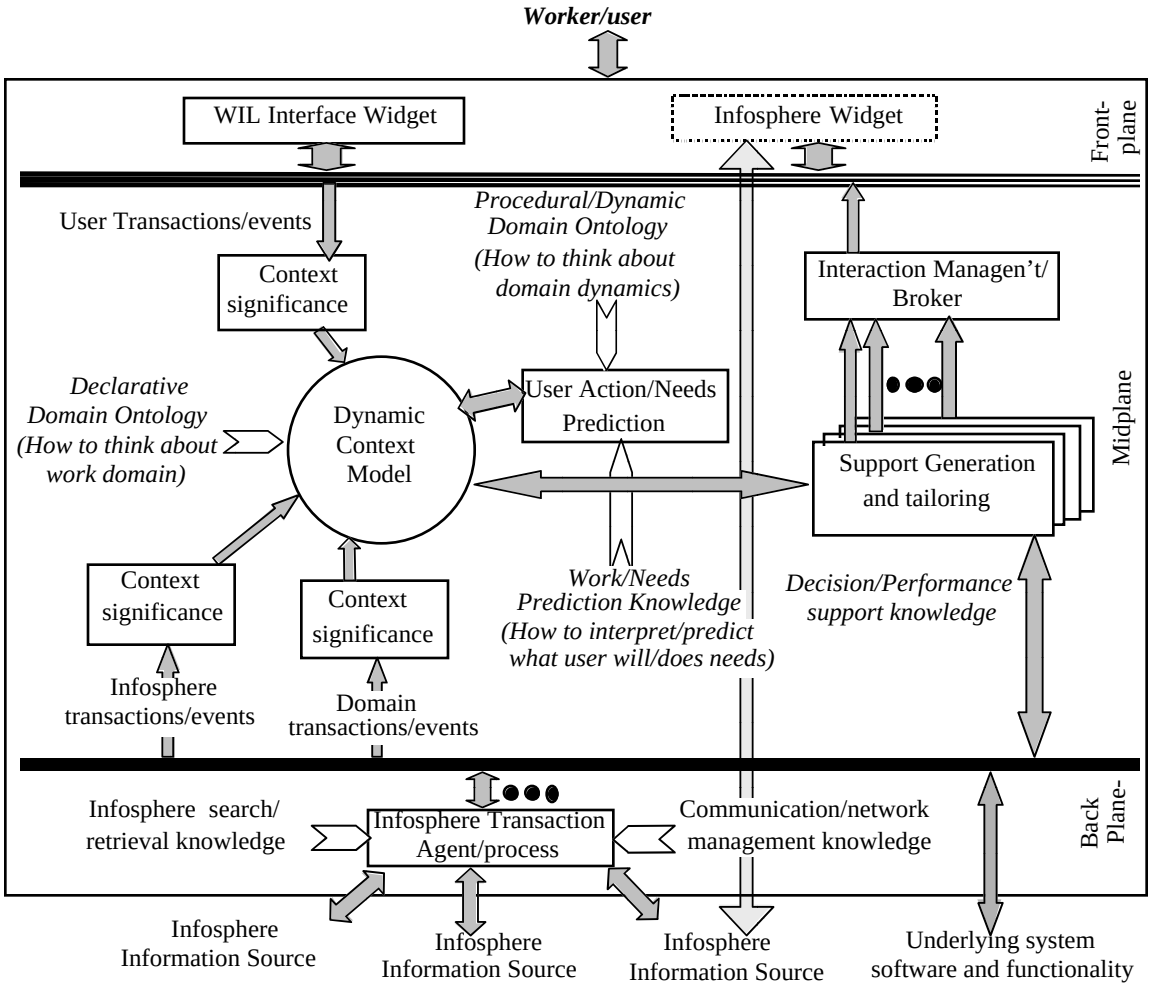

Figure 1. WIL Conceptual Architecture

Organization of computation in WIL

The conceptual WIL architecture is shown in Figure 1. At the highest level, the architecture is divided into front-plane, mid-plane, and back-plane components. A WIL interface interacts with the human user through a series of direct interaction objects (i.e., widgets), whose look and feel are tied both to the work domain and the interface functionality they present. These direct interaction objects are managed as a front-plane layer, which can either be integrated with the (legacy) information system displays and controls (in which they pass thru the remainder of the WIL) or serve as the complete user interface to the underlying information system.

A WIL application also needs to interact with various algorithmic components, databases, data streams, etc. that make up the external infosphere. The functionality that allows WIL to interact with this infosphere make up the back-plane. This allows the WIL interface to avoid being the end-point to a stovepipe , connected to only a single data source or underlying system. More importantly, the breadth of the backplane gives the interface access to the information it needs to develop and maintain its context model (and allows it to do so on either an active or a passive basis).

In between the front-plane and back-plane of Figure 1 lies the mid-plane, which is the main locus of the active processing within WIL as an active interface architecture. A WIL interface simultaneously and continuously performs three general functions within the mid-plane:

- determining what the current work context is (*the understanding function*)

- identifying what work needs to be done given the current context (*the work/need assessment function*); and
- support those users work needs in light of the current work context (*the tailored support function*).

The mid-plane is consists of a set of three major infrastrucural processes that implement these functions. These processes:

- automate the understanding function by representing work context explicitly and maintaining a dynamic understanding of the current work context;
- automate the work/need assessment function by using context-driven interpretation techniques; and
- automate the tailored support function by dynamically adapting a set of generic work-support processes, using the content of the dynamic context representation.

Clearly, the results of the tailored support function are the only aspect of mid-plane processing that the human user working with the WIL interface sees. The understanding functions and actions/need assessment functions are the enabling capabilities that allow the WIL interface to be helpful to the user without an explicit intent inferencing process.

Mid-plane Processing Structure

As stated above and shown in Figure 1, the mid-plane of a WIL interface is organized into three interacting processes, corresponding to the three functions listed above. Each is discussed in additional detail below, although space limitations preclude an in-depth analysis. The discussion for each process is framed first in general terms, and then in terms of the types of domain-specific knowledge that must be built into the WIL interface to apply that process in a specific application domain.

The understanding function: building and maintaining work context. Cognitive research has consistently shown that human decision-making and information processing are organized around an internal representation of context (see, for example, theoretical discussions by Barwise and Perry, 1983, Pylyshyn, 1984, Klein ,1989, Kolodner, 1991; Endsley, 2000). The WIL architecture takes a similar approach, representing context as an explicit internal body of knowledge about the domain, about current and evolving objects (abstract or concrete) and relationships among them, and about the set of intelligent entities within the work environment (including the WIL itself). The context representation also may include the projected (in time and space) significance, effects, and/or behavior of objects or relationships that are part of the context.

WIL represents context as a complex body of declarative knowledge organized into an ontology that is usually highly specific to the work domain and work role being supported by the WIL application. Importantly, the body of information in the context model is selective rather than exhaustive. This is also cognitively rooted, reflecting the tendency of human experts to learn the types of information that are relevant and seek (only) those, ignoring the much larger body of irrelevant data as the cognitive equivalent of noise. Thus, building an internal context representation requires not just knowing how to 'think about the domain' (i.e., possessing the structure of the appropriate ontology and ability to instantiate it), but an additional set of cognitive and perceptual processes that can selectively acquire this relevant information and relate it to the context ontology.

These processes themselves fall into two categories -- those that deal with information in the environment outside the system, and those that seek information *inside* the system. The former (externally-oriented processes) are the analog of human perceptual processes. They interpret information received from events and processes outside the system (i.e., from outside the WIL) and internalize it in terms of the domain ontology that structures the internal context model. The actual acquisition of external world information for these perceptual processes occurs through the analog of sensory processes in the back-plane or the front plane, depending whether the information concerns the human user of the WIL (on whom information is acquired through the front-plane), or concerns the environment, other people in the system, or the infosphere (about which information is acquired through the back-plane). These perceptual processes require the domain-specific perceptual knowledge needed to interpret the 'raw' data acquired through the front- or back-plane and to translate that data into internal knowledge that becomes part of the context model. In the case of infosphere retrieval, the sensory-like processes may also require extensive procedural knowledge concerning how to locate and interrogate the information source itself.

The internally-oriented processes also include two sub-types, which are the analogs of interpretive and metacognitive processes in people. Interpretive processes are reasoning processes that operate on information already in the context model and infer more new, often more abstract, information. For example, externally oriented processes may acquire information about aircraft tracks from a geographical plot (such as an air traffic control display), and internalize these tracks symbolically as part of the context model. Internally oriented reasoning processes might operate, for example, on the individual track knowledge currently in the context model to seek out and identify tracks that are traveling as a group or formation, and make the formation itself part of the context model as well. These interpretive processes are activated by the presence of actionable information in the context model, and require both the procedural knowledge needed to carry out the interpretive context, and the control knowledge needed to define the activation conditions. The metacognitive processes are reasoning processes that operate on the state/actions of the cognitive process itself. These are the processes by which information on the internal operation of the WIL mid-plane is transformed into a symbolic representation and internalized as part of the context model itself.

The work/need assessment function: identifying contextually-appropriate work needs. The context-based approach allowed the interface to identify appropriate things that needed to be done, and to offer full, partial, or supportive automation in performing them. Rather than determining what the user is doing, or trying to do, or wants to do, the WIL architecture instead takes an approach much more commonly used by humans beings who are trying to collaborate. It analyzes the context model and tries to determine what kinds of work activities need to be done, what kinds of information is needed (to do them), and/or what changes in viewpoint, level of abstraction, or frame of reference might be useful *at the current moment.* As the various processes carrying out the understanding function (see above) change the context model, the set of current work needs may also correspondingly change. Some activities that were relevant a moment earlier may now be in appropriate, just as new information needs, etc. may arise.

In computational terms, each of the three broad categories of work needs -- work activities, information needs, viewpoint/abstraction frames -- will be associated with a set of spontaneous computation units. Each of these units can recognize, for a specific content element (e.g., specific work activity or specific information need), the context cues which define several different attributes or state of that category of work need. These attributes or states include such things as:

- activation -- conditions that recognize that it is now relevant,
- sustainment -- conditions that keep it activated,
- accomplishment -- conditions that recognize that it has been completed or satisfied, and
- deactivation -- conditions that recognize that it is no longer relevant.

As these attributes are computed, they themselves become part of the context model, via internally-directed metacognitive processes.

The tailored support function: generating context-based end-user support. As the WIL application identifies specific elements or aspects of work that need to be done, these too become part of the work context, as shown on Figure 1 above. From these work needs, then, the interface can identify specific things that it could do to help the WIL user meet these work needs. Although these work-support functions could be developed in a purely domain-specific manner for each new WIL application, a set of general (in the sense of commonly-occurring) work support strategies have been defined from analysis of a set of past WIL applications (see Zachary et al., 2001). Table 1 defines the set of nine general work support functions from that analysis. By incorporating generalized processes which can perform each of these functions into a WIL application, the process of providing each type of support reduces to one of tailoring each generic process, according to the needs of the specific application and current context. (Of course, other work-support functions could also be incorporated as needed on a domain-specific basis.)

Table 1. WIL Work-Centered Support Functions

Function	Description
Activity Management	helping the person determine what work tasks or elements need to be done at the current point in time, given the current work context
Work Process Structuring	helping the person determine how a given work task or element ought to be approached or broken down, given the current context of the work environment
Customized Performance Assistance	performing, at the request or approval of the person, a given work task or sub-element
Explanation/Elaboration	providing explanations of system, sensor, infosphere events or processes in terms of their impact or effect on the user s own work and work needs
Work-Environment Representation	constructing and displaying multiple abstracted representations of the structure and processes in the work domain or environment
Infosphere data retrieval	going out into the information environment, outside the interface itself (i.e., into data streams, databases, etc.), to collect and make available pieces of information
What if	allowing the user to create a hypothetical work view in which possible future actions are defined and explored
Situation Awareness	providing information at multiple levels of abstraction on the external system or process (e.g., the battlespace) with which the user s work is involved
Work-flow Prioritization	helping the person determine which, among many work elements or tasks that need to be done at a given time, have priority in the current work context and why

For example, activity management could be offered whenever the process of identifying work needs determines that multiple work activities needed to be performed simultaneously, thus signaling a case where the human user might benefit form activity management support. The tailoring process, in this case, simply involves associating the linking content from the context (i.e., work activities that need to be done) with the abstracted sub-process that manages activity management and its interactions with the user via the front-plane. Analogous approaches work for the other general work support functions.

Conclusion

The main idea advanced here is that WIL interfaces are able to provide functions that are typically associated with active interface or associative systems, but without any intent inferencing whatsoever. Instead, the approach relies on the shared relationship to the overall situation as the basis for determining what type of support is to be offered to the user, and when.

The WIL concept and architecture was developed through an analysis of and abstraction from a series of prior active interface applications, all of which used the context-based approach described here. Those prior applications were all coded essentially by hand, and required substantial time and effort to implement (even that development process employed a high-level cognitive modeling toolkit that reduced the effort). A main motivation of the definition of the WIL architecture was to create an opportunity to further automate, streamline and structure the development process. The authors are currently working on implementing the WIL as a generalized software framework along with a set of tools to customize it to specific application contexts. These tools both help to capture domain-specific knowledge and to integrate this domain-specific knowledge with the general WIL software structures to yield a domain-specific WIL application. This WIL-Application Toolkit (WAT) can be used to develop, customize, and maintain specific WIL applications.

References

Barwise, J., & Perry, J. 1983. *Situations and Attitudes.* Cambridge, MA: MIT Press.

Bell, B. and Santos, E. (Eds.) 2001. *Intent Inference for Collaborative Tasks: Papers from the 2001 Fall Symposium*. Technical Report FS-01-05, American Association for Artificial Intelligence.

Eisenberg, E. M. 1990. Jamming: Transcendence Through Organizing. *Communication Research,* Vol. 17, No. 2, Sage Publications.

Endsley, M.R. 2000. Theoretical Underpinnings of Situation Awareness: A Critical Review. In Endsley, M.R. and Garland, D.J. (Eds.), *Situation Awareness Analysis and Measurement.* Mahwah, NJ: Lawrence Erlbaum Associates.

Jones, P., Mitchell, C., & Rubin, K. 1988. *Intent Inferencing with a Model-Based Operator's Associated* (Rep. No. 88-2). Atl, GA: G. Inst. Tech., School of Industrial and Systems Engineering.

Klein, G. 1989. Recognition-primed decisions. In W. Rouse (Ed.), *Advances in man-machine systems research*. (pp. 47-92). Greenwich: JAI Press.

Kolodner, J. 1991. Improving Human Decision Making Through Case-Based Decision Aiding. *AI Magazine, 12*(2)

Pylyshyn, Z.W**.** 1984. *Computation and Cognition Toward a Foundation for Cognitive Science*. Cambridge: The MIT Press.

Robertson, S., Zachary, W., and Black, J., Eds. 1990. *Cognition, Computation and Cooperation: An Interdisciplinary Study of Multi-Agent Systems*. Norwood: Ablex Press.

Suchman, L. 1987. *Plans and situated actions: The problem of human-machine communication*. Cambridge, England: Cambridge University Press.

Vaubel, K. P., & Gettys, C. F. 1990. Inferring User Expertise for Adaptive Interfaces, *Human Computer Interaction, Vol. 5*, pp. 95-117, Lawrence Erlbaum.

Vicente, K. J. and J. Rasmussen 1992. Ecological interface design: theoretical foundations. *IEEE Transactions on Systems Man Cybernetics **22**(4):* pp. 589-606.

Zachary, W. Schremmer, S., & Donmoyer, J. 2001. *Design Analysis for A Work-centered Infomediary Application Toolkit.* CHI Systems Technical Report 011214.01011, Lower Gwynedd, PA: CHI Systems Inc.