THE DOT-LOOP ARCHITECTURE: A VIRTUAL REALITY-BASED SYSTEM FOR AIRCRAFT DESIGN, OPERATION, AND TRAINING

David Littman
Datamat Systems Research, Inc.
McLean, VA

B. K. Gogia
Datamat Systems Research, Inc.
McLean, VA

Douglas Anthony
Praxis Technologies Corporation
Woodbury, NJ

Introduction: Motivation and Goals

For many years, designing a user interface for operation of a complex aircraft has been a primarily human-intensive process in the sense that prototypes are laboriously developed by hand, subjected to human factors testing, and then revised. This process is costly in that iterations are expensive, must be limited in number and scope, the time between iterations can be significant. and the cost of undetected design errors and miscalculations can be extremely high.

Recently, a great deal of effort has been focused on developing computer-based tools that permit designers to prototype operator interfaces relatively quickly. Such tools run the gamut of function, from tools that support rapid design and sequencing of information and response windows, such as Microsoft's Visual series, to sophisticated tools that run on high-end graphics workstations and allow designers to develop a "look-alike" interface for an airplane, ship, or power plan control system, such as the tools developed by Virtual Prototyping. These tools thus provide various approaches for designers to build operating mockups of user interfaces, test them, and then modify them to improve performance. In essence, these tools are part of a cyclical design-build-test process.

For the most part, however, these tools still require substantial human effort to manage the process of design and to provide and utilize the knowledge required to interpret operator performance data and alter the design based on the performance data. In our view, it would be a useful advance if these tools could assist interface designers by providing knowledge based guidance in 1) designing highly realistic -- virtual reality based -- interfaces instrumented for operator testing and 2) interpreting the results of operator and trainee interaction with the goal

of improving the design to improve performance.

The type of knowledge based tools required to help designers will require significant investments of time and money to develop and may seem premature in view of currently available hardware and software technologies. However, there are three reasons why it is important to begin to consider developing knowledge-based tools to support the use of virtual reality-based prototyping tools:

• Necessity: Many government agencies that acquire systems which have large design and development costs are beginning to impose requirements for efficient use, coordination, and reuse of design knowledge.

• Feasibility: Several large, high-visibility projects funded by the military are concerned with the development of standards for knowledge based design environments that support all phases of design.

• Possibility: The coming generation of hardware will very soon make it possible to run extremely complex virtual reality simulations and knowledge based tools on what will amount to the next generation of low-end desktop workstations. In addition, the time significant generations is shrinking rapidly: Anyone who has seen a native graphics design program running on a low-end PowerPc machine will understand.

In this paper, we describe our efforts to design a knowledge-based architecture to support iterative design and modification of testable virtual reality prototypes of user interfaces for aircraft with the goal of optimizing them for operation and training. The primary goal of the Design, Operation, and Training Loop (the DOT-Loop) development architecture is to permit the construction and testing of a virtual reality, simulated, version of the operator interface for an aircraft before it is actually "built." The DOT Loop system will aid the interface designer in evaluating data from the pre-build simulation by a knowledge-based component of the design expert that 1) identifies aspects of the interface design that lead to less than optimal operation or training performance and 2) makes recommendations to the human designer about how to reconfigure the operator interface for improved operation or training.

Description of DOT-Loop Architecture System

In this section, we give a brief description of each of the components of the DOT-Loop architecture, describe some of the knowledge that resides in each, and give a brief example of how this component would operate in action.

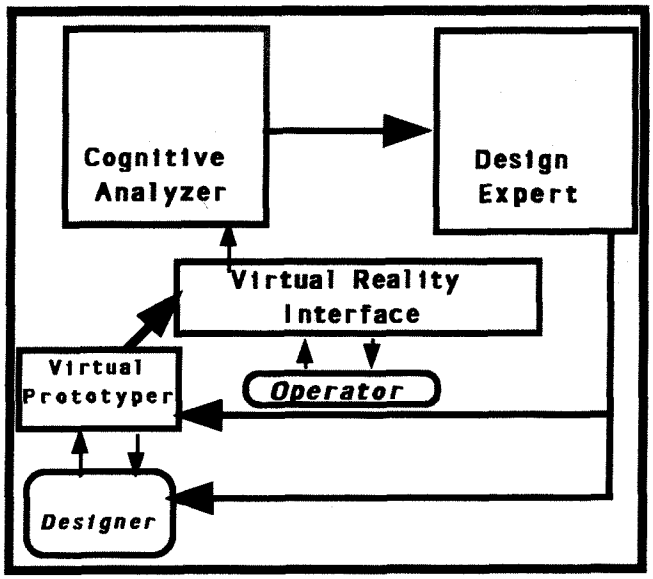

Figure 1: DOT-Loop Architecture

Figure 1 shows the overall design of the DOT-Loop architecture. The DOT-Loop architecture has three main knowledge based

components and a reconfigurable suite of input and output devices to support the operator in virtual reality-based interaction with the system under development.

The operator interface designer uses the virtual prototyper to construct an operator interface for, e.g., a helicopter. The operator's interaction with the virtual prototype is managed by the Virtual Reality Interface. The operator interacts with the Virtual Reality Interface, producing performance data. These data are transmitted to the knowledge-based Cognitive Analyzer component, which identifies sources of performance errors in the design. Information about the design problems is transmitted to the knowledge-based Design Expert, which interacts with the human designer to modify the design of the operator interface in ways that are intended to reduce the sources of performance errors.

The Design Expert. Figure 2 illustrates the three main components of the Design Expert of the DOT-Loop. The Design Expert has three main types of knowledge. The application domain knowledge base contains information about the specific domain of application, such as the helicopter domain.

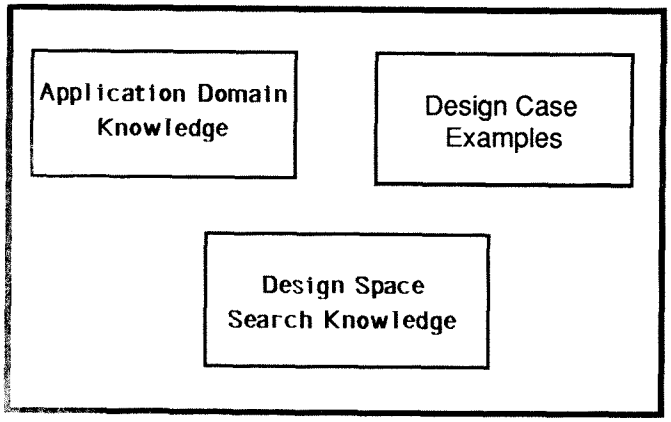

Figure 2: Design Expert

For example, the application domain knowledge base would have information about the tasks that operators must perform. This would include, among many others, the types of information the operator must process, the decisions he or she must make using the information, and the time and information constraints on the operator's decision making.

The knowledge base of design case examples contains examples of successful and unsuccessful operator interface designs for the task structures represented in the application domain knowledge bases. For example, "war stories" about interface design strategies and concepts that seem good on paper -- and thus lead designers unfamiliar with them down a very expensive garden path -- can be represented in the case example knowledge base for use by the design expert.

The Design Expert's design space search knowledge consists of methods for exploring the design space of operator interfaces. The design search knowledge base contains two major types of information about searching the design space.

First, the design search knowledge base contains specific algorithms for searching the design space. For example, we have investigated the use of genetic algorithms to converge on coordination structures for distributed cooperative decision support systems; we have also applied more standard knowledge intensive, rule-based heuristic search methods.

Second, the design search knowledge base contains information that guides the selection of specific search algorithms, such as genetic algorithms, to use to search the design space. This *search meta-knowledge* encodes the conditions under which e.g., genetic search algorithms, or simulated annealing methods, would be most fruitfully applied in a

particular design problem requiring search in the design space.

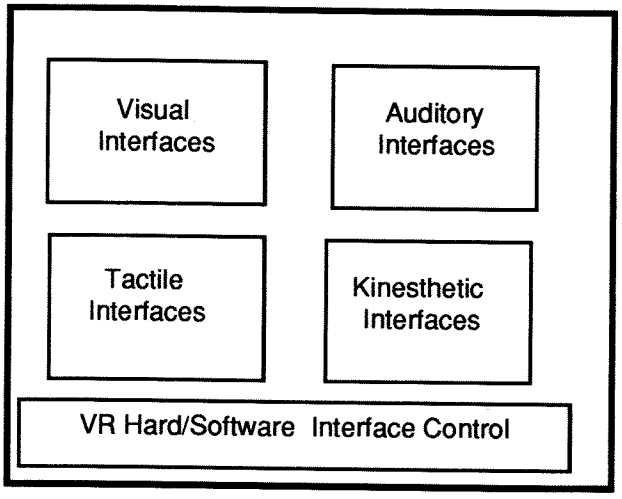

Figure 3: Virtual Reality Interface

The Virtual Reality Interface. Figure 3 shows the main components of the Virtual Reality Interface used by the operator or trainee to exercise the virtual prototype of the operator interface. The Virtual Reality Interface consists of components that let the trainee or operator interact with the virtual prototype as if it were a real, physical artifact. For example, visual input devices provide the operator with exactly the same information that would be available in a real, physical version of the interface.

Tactile input/output devices, such as bladdered gloves, give the operator the same information that would be available from the switches, levers, buttons, and dials in the real physical version of the interface. Kinesthetic input/output devices would do the same for movement cues and auditory output devices the same in the sound domain.

The VR Hard/Software Interface Control organizes the interaction of the operator or trainee with the four types of interfaces. The interface control has the responsibility for 1) maintaining the low-level communications of the interfaces with the virtual prototype program and to maintain the "flow" of the virtual experience for the operator or trainee.

The Virtual Prototyper. The Virtual Prototyper, shown in Figure 4, provides the interface designer with a set of tools to construct a virtual reality-based implementation of an aircraft operator's interface.

The Virtual Prototyper consists of four main components. First, the Virtual Prototyper contains design space search knowledge, similar in content and role to the Design Expert's search knowledge. In addition, the Virtual Prototyper also contains case examples of virtual prototypes, again in parallel with the case examples of the Design Expert.

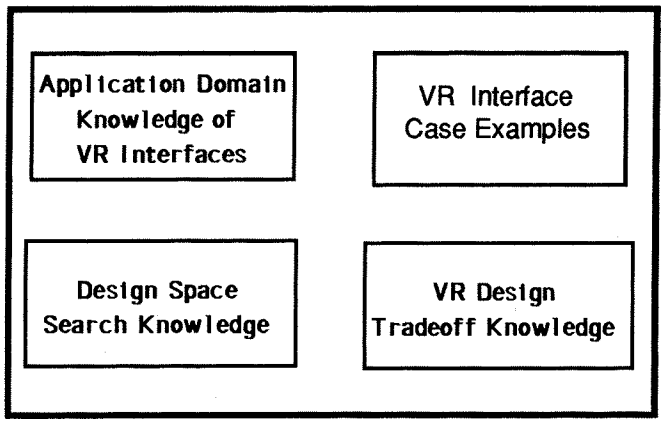

Figure 4: Virtual Prototyper

Finally, the Virtual Prototyper has knowledge about the application domain of virtual reality interfaces and knowledge about the tradeoffs that must be considered in deciding which of several alternative VR prototype interfaces would best serve the application and the system evaluation requirements.

In essence, the Virtual Prototyper provides a knowledge-based environment with which it is possible to construct virtual reality-based

prototypes that can be exercised with the Virtual Reality Interface.

The Cognitive Analyzer. The Cognitive Analyzer, shown in Figure 5, is a key component of the DOT-Loop architecture. The Cognitive Analyzer has knowledge about the perceptual and cognitive decision making processes of operators of aircraft using various equipment configurations to execute alternative task structures.

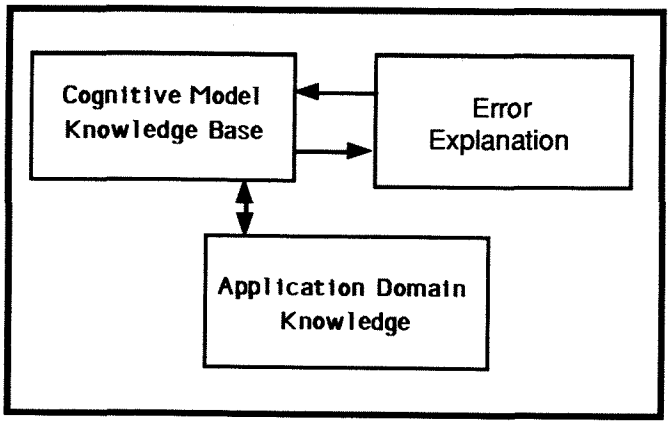

Figure 5: Cognitive Analyzer

For example, the Cognitive Analyzer has knowledge that permits it to reason about how a particular decision making task that requires several sources of input would be affected according to how the input is divided between auditory and visual input channels.

The cognitive model knowledge base represents knowledge about the different perceptual and cognitive processes that humans can bring to bear the application domain problem solving tasks represented in the application domain knowledge base.

For example, a psychological "rule of thumb", often used by interface designers, is that a task based on a division of information sources into auditory and visual channels does not cause the degree of performance decrement caused by single channel information sources. However, what is less widely known is that auditory channels that carry information used for localization tasks -- such as a stereophonic sound progression representing the spatial position of an enemy aircraft -- *does* interfere significantly with visual processing. This type of knowledge can be critical to explaining why a particular division of information into auditory, visual, and kinesthetic channels produces an observed pattern of errors and correct responses.

As in the Design Expert, the application domain knowledge base has knowledge about e.g., crew communication tasks; weapons assessment tasks; and so forth.

The job of the error explanation knowledge base is to produce an account of why operators or trainees produced the observed performance data. The error explanation takes as input the performance data produced by the operator or trainee with the Virtual Reality Interface and uses the knowledge in the cognitive knowledge base to explain how the operator or trainee produced the observed performance data.

The error explanation mechanism uses a model based reasoning algorithm. This mechanism identifies the alternative configurations of cognitive models and their deployment during task performance that are likely to have been used by the operator and, therefore, to have caused the observed errors and correct performances. The explanation contains a profile of how and when the information produced by each of the components of the virtual prototype interface was used, misused, or undetected by the operator. Errors in performance are associated with specific uses or failures of perceptual and cognitive components to use information.

The information produced by the Cognitive Analyzer is routed to the Design Expert, which assists the interface designer to search the design space to find design alternatives that will avert the problems caused by the current configuration.

Conclusions and Future Directions

We have described the DOT-Loop architecture for knowledge-based environments to assist designers of operator interfaces for aircraft or, indeed, any other systems requiring complex interactions with people. The purpose of the design environment is to allow developers of operator interfaces to rapidly prototype and test highly realistic -- virtual reality-based -- versions of alternative interfaces.

The DOT-Loop architecture is heavily knowledge based and an architecture does not a system make.

Acknowledging that we may be slightly ahead of the virtual reality technology curve, in our view the most serious work to be done lies in the identification, representation, and encoding of the oceans of knowledge associated with each of the knowledge bases required by the DOT-Loop architecture. Many design environment architectures have foundered on these shoals and we believe that it is critical to pursue this activity in preference to the development of other aspects of the implementation.

We believe that a potentially significant application of design environments based on the DOT-Loop lies in moving the assessment of alternative operator interfaces and training procedures further and further back into the system development process. Ultimately it may be possible to consider methods for acquiring and representing enough knowledge in a DOT-Loop system to automatically generate training and operation procedures from design specifications. Constraints on training and operation procedures are candidates could be part of the knowledge guiding the search through the space of alternative interfaces as part of the search through the artifact design space.

Finally, a logical question is "Why not use the virtual reality interface from the DOT Loop system as the *real* interface for aircraft -- or any other complex human machine system?" Why not indeed?

We are intrigued by the possibility of beginning to consider "doing it all in virtual reality and software." While this may seem farfetched, there are several reasons for considering this as an implementation strategy for operator interfaces.

First, and most obvious, is that the design of the interface with the prototyping system could produce architecture-based, immediately runable operational software which could be maintained with the DOT-Loop design system.

Second, with this approach, the knowledge in design systems based on the DOT-Loop architecture could be used to tune *real* operator interfaces dynamically. For example, it may be possible to use a DOT-Loop design system during the development phase to dynamically seek to improve an operator's performance of tasks by continuously altering the interface and the task structure until a desired set of performance criteria are met.

This process amounts to improving the software that defines the operator interface during design. Once it has been improved to an acceptable level of performance, the software can be migrated to the actual operational equipment, which would use the

same software and virtual reality-based operator interface technology. As changes are required, the same architecture-driven activities performed during initial design could be carried out to make changes.