Conf-940433--8

EXPERT SYSTEMS FOR THE TRANSPORTATION OF HAZARDOUS AND RADIOACTIVE MATERIALS

Craig E. Luce
James C. Clover
Juan J. Ferrada

ABSTRACT

Under the supervision of the Transportation Technologies Group which is in the Chemical Technology Division at Oak Ridge National Laboratory, an expert system prototype for the transportation and packaging of hazardous and radioactive materials has been designed and developed. The development of the expert system prototype focused on using the combination of hypermedia elements and the Visual Basic™ programming language. Hypermedia technology uses software that allows the user to interact with the computing environment through many formats: text, graphics, audio, and full-motion video. With the use of hypermedia, a user-friendly prototype has been developed to sort through numerous transportation regulations, thereby leading to the proper packaging for the materials. The expert system performs the analysis of regulations that an expert in shipping information would do; only the expert system performs the work more quickly. Currently, enhancements in a variety of categories are being made to the prototype . These include further expansion of non-radioactive materials, which includes any material that is hazardous but not radioactive; and the addition of full-motion video, which will depict regulations in terms that are easy to understand and which will show examples of how to handle the materials when packaging them.

1. INTRODUCTION

The ability of the computer to simulate and make decisions as a subject matter expert (expert system) has greatly expanded potential computer applications. With this type of computing power available in an increasingly user-friendly format (using object-oriented programming and hypermedia capabilities), the development of a transportation packaging expert system that is based on the hazardous materials regulations is possible.

The Transportation Technologies Group of the Engineering Coordination and Analysis Section which is in the Chemical Technology Division at Oak Ridge National Laboratory (ORNL), under the sponsorship of the U.S. Department of Energy's (DOE's) Office of Environmental Restoration and Waste Management, Transportation Management Division (EM 561), has developed an expert system prototype by undertaking the task to develop a computerized application of transportation regulations. One of the objectives of this task was to evaluate the feasibility of developing a computerized expert system to ensure straightforward, consistent, and error-free application of the hazardous materials regulations.

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

2. PROTOTYPE DEVELOPMENT

The feasibility of developing the full expert system for the transportation and packaging of hazardous and radioactive materials was initiated within the framework of three subtasks: (1) analysis of commercial packages related to regulation scanning, (2) analysis of computer languages to develop the expert system, and (3) development of the expert system prototype. The strategy to develop the latter subtask was to (a) develop modules to capture the knowledge of different areas of transportation and packaging and (b) analyze the feasibility of integrating these different modules into one final full package. The individual modules contemplate one prototype for transporting and packaging of radioactive materials and another for transporting hazardous chemical materials. This paper summarizes the efforts in developing the radioactive materials transportation expert system prototype.

An element that plays an interesting role in the prototype development is the possibility of using some existing Prolog (Programming in Logic) code. A Prolog code was previously developed to demonstrate the proof-of-principle that the regulations could be translated into a logic diagram and from them into a rule-based system. The ultimate decision mechanism should take this factor into consideration.

2.1 ANALYSIS OF COMMERCIAL PROGRAMS

The analysis of the commercially available software, RegScan™ and Environmental/Safety Library™, indicated that both packages, although very useful for navigating the pertinent regulations, are not particularly suitable for the determination of the type of packaging required for hazardous and radioactive material transportation. However, the regulations can be downloaded from these software systems and used as the source data base for the regulations that will be accessed by the expert system.

2.2 ANALYSIS OF LANGUAGES

Once the system requirements were analyzed, a crucial step in creating the expert system was to determine the environment in which the expert system would run. *Environment* in this case means 1) the type of computer the system will run on, 2) the type of operating system the system will run under, and 3) the tool that will be used to develop the system.

2.2.1 System Requirements

The basic premise of development was that the expert system would run on the PC platform. From Sect. 2.1, it is clear that one requirement of the expert system is the ability to access the regulations from the commercial programs that update the regulations on regular basis. The friendliness of the user interface is another important requirement together with the ability to navigate throughout the regulations and display graphics and full-motion video information on the screen. Above all, the program has to provide an environment wherein a rule base system that represents the application of regulations can be implemented.

2.2.2 Rating the Candidate Tools

To decide what tool would be used to create the expert system, six candidate tools were rated using six criteria. These criteria are 1) multimedia capabilities, 2) rule-making capabilities, 3) flexibility of the environment, 4) user interface provided, 5) data-handling capabilities, and 6) ease of use of each tool. The following tools were rated: OWL^{TM} Industries Guide (a multimedia document presenter), the C/C++ programming language (a general programming language), the ProLog programming language (a logic-based DOS programming language), general expert system shells (tools used to create expert systems), and the Visual Basic programming language (a general Windows programming language).

2.2.2.1 Multimedia Capabilities

The criterion "multimedia capabilities" refers to the capability of the tool to create multimedia presentations. This means that the tool has the ability to present text, graphics, and full-motion video in a context-sensitive environment. To rate "very good," the multimedia presentations must be easy to create (by the developers) and must be easy to access (by the users). An "adequate" rating means that the tool can present some of the different media, but not all, or does not contain a good interface from either the designer's or user's perspectives. A "very poor" rating means that the rated tool has no multimedia capabilities. This rating is not exceedingly important because any Windows-based application can use Windows multimedia tools (which would rate "good").

2.2.2.2 Rule-Making Capabilities

The criterion "rule-making capabilities" refers to the capability of the tool to work with knowledge-based systems by creating "rules" or "if-these-criteria, then-do-that" statements. A "very good" rating means that the tool enables the designers to easily create and modify rules to reflect the logic in the logic diagram mentioned previously. An "adequate" rating means that the tool can manage rules of some type, but that the rules are harder to develop or use. A "very poor" rating means that the tool contains no innate capability to create and maintain "rules". This criterion is fairly important, because "rules" are the basic building blocks of the expert system.

2.2.2.3 Flexible Environment

The criterion "flexible environment" refers to the ease of implementing features that are not central to the tool. A "very good" rating means that the tool is highly flexible and contains the framework to accomplish virtually any possible feature within the environment. An "adequate" rating means that the tool is flexible enough to accomplish a variety of extra tasks but that it is limited in the depth of what it can accomplish or that it simply can not do certain tasks. A "very poor" rating means the tool is not very flexible at all and can be used to accomplish only what it was directly developed to do. This criterion is fairly important because a flexible tool can make up for deficiencies in the other criteria and provide space for further expansion.

2.2.2.4 User Interface

The criterion "user interface" refers to the variety of input and output methods provided by the tool. A "very good" rating means that the tool has many varied input and output methods such as buttons, list boxes, graphics, text sizes and fonts, menus, and tool bars and that these methods are easy to develop and use. An "adequate" rating means that the tool has a fair amount of input and output methods or that some of the methods are harder to use or develop. A "very poor" rating means that the tool contains only a few input and output methods. This criterion is very important because the user needs input and output that are 1) easy to understand and 2) that do not make the expert system cumbersome to use.

2.2.2.5 Data-Handling Capabilities

The criterion "data-handling capabilities" refers to the ability of the tool to create, maintain, and access data bases. A "very good" rating means that the tool can create, maintain, and access data bases with little effort by the designers and without long waiting periods for the users. An "adequate" rating means that the tool does not provide all these data-handling capabilities but that it does provide some capabilities that decrease the developer's work. A "very poor" rating means that the tool has no built-in data-handling capabilities. This criterion is important because this expert system relies heavily upon certain hazardous material data bases (i.e., tables from the regulations).

2.2.2.6 Ease of Use

The criterion "ease of use" refers to how difficult the tool is to use from the designer's point of view. A "very good" rating means that the tool is simple to use: it uses a well-designed interface that will increase the developer's efficiency. An "adequate" rating means that the tool is not too difficult to use, but that it lacks certain features that will enable the developers to work efficiently. A "very poor" rating means that the tool is very difficult to use and thus will significantly slow down the work of the developers. This criterion is very important because a powerful tool will not likely be used if it cannot be used easily.

2.2.2.7 Use of Existing Prolog Code

This criterion refers to the possibilities of using existing Prolog codes used in the proof-of-principle stage of the expert system prototype. A "very good" rating means that the tool can use the code without modification. An "adequate" rating means that the tool allows use of a total or partial code in the application, including some changes. A "very poor" rating means that the Prolog code cannot be used in the application.

2.2.2.8 Additional effort to finalize the prototype

This criterion refers to the possibility of completing the prototype using the existing Prolog code. A "very good" rating means that the tool can use the existing code and that very little additional code is necessary to complete the prototype. An "adequate" rating means that a fair amount of code must be added to complete the prototype. A "poor" rating means that changes must to be made to the existing code and that extensive additional coding is required.

2.2.3 Ratings Given to Each Environment

Guide™—This tool ranked "very good" in multimedia capabilities and ease of use. This may be a less important category because Windows can provide the needed multimedia capabilities. Guide rated "good" in the categories of flexible environment and user interface. Guide does not provide a good mechanism to build rules. Guide does not handle diverse types of data easily, and distribution is not very feasible (purchase of Guide is necessary to run the program, run-time version does not follow the software updates). Guide rates "good" in additional codes needed to complete prototype.

C/C++—With good flexibility and an adequate interface for the user, as well as good data handling, C/C++ possesses all the power necessary to create the expert system. The main problem is that C/C++ is very difficult to use as a development environment. Only a highly experienced software developer should consider attempting to create the expert system in C/C++. This dismisses C/C++ as a possible choice for the expert system.

Prolog—The application produced with Prolog was not a true Windows application. Thus, the "poor" rating in multimedia capabilities is a serious shortcoming. Prolog is exceptionally suited for rule-based systems and requires no additional fee for distribution. In addition to this advantage, Prolog is very easy to use for the proof-of-concept stage of development. However, Prolog simply has too few features to be an appropriate multimedia development environment. If future versions of Prolog are true Windows applications with better user interfaces and data-handling capabilities, those versions might be good contenders for developing the expert system.

Expert System Shells—Generally speaking, expert system shells are not true Windows applications. This means that the "very poor" multimedia rating is going to be hard to overcome, especially without good flexibility. Also, the only adequate data-handling capabilities and user interfaces make the expert system shells bad choices for the expert system using multimedia features. VP-Expert™, and Nexpert Object™ were evaluated.

Visual Basic—The poor rating in "use of existing Prolog code" and the "adequate" rating in multimedia for Visual Basic is overcome by the fact that it is a true Windows application, allowing the expert system user to use Windows multimedia to accomplish this task. The adequate-rule making capabilities of Visual Basic can be overcome by the flexibility of the tool. Applications developed in Visual Basic do not require any additional fee for distribution.

From these ratings, it is evident that Visual Basic is the best environment to create the expert system. However, one element that weighed heavily in the decision process was the fact that a great deal of code in Prolog had already been created during the proof-of-concept stage. It was clear that putting a sole Prolog version together required little additional effort. Now, putting together Guide and Prolog and thus taking advantage of the rule-making capabilities of Prolog and the multimedia capabilities of Guide seemed to be very attractive. Nevertheless, any combination including Visual Basic represented the best alternative. Based on this analysis it was decided that the development of the expert system prototype would consist of three steps: (1) a DOS version using Prolog, mainly because it was simple to complete the prototype, (2) a Windows version using Guide and Prolog, and (3) a Windows version using GUIDE and Visual Basic.

Table 1 presents the results of the ratings.

Table 1. Results of software ratings.

	Guide	C++	Prolog	Expert Shells	Visual Basic
Multimedia capability	Very good	Good	Poor	Poor	Adequate
Rule-Making capability	Poor	Good	Very good	Very good	Adequate
Flexible environment	Good	Good	Adequate	Poor	Very good
User interface	Very good	Adequate	Adequate	Good	Very good
Data-Handling capability	Poor	Very good	Poor	Adequate	Very good
Ease of use	Very good	Poor	Adequate	Good	Very good
Distribution	Poor	Very good	Very good	Poor	Very good
Use of existing Prolog code	Very good	Poor	Very good	Poor	Poor
Additional code to complete prototype	Good	Poor	Very good	Poor	Poor

2.3 DEVELOPMENT OF THE EXPERT SYSTEM PROTOTYPE

Once the decision was made to test three developmental environments for the development of the expert system prototype, the work was focused on completing the knowledge acquisition needed to capture the process of selecting packages for transporting radioactive materials. The general strategy for developing the prototype was to (a) develop a logic diagram of the decision-making process involved in packaging radioactive materials for transport and (b) put this expertise in computer code.

2.3.1 Knowledge Acquisition

The U.S. Department of Transportation (DOT) and the U.S. Environmental Protection Agency (EPA) have stringent regulations regarding the packaging of hazardous materials before these materials are shipped. These regulations specify types of packages (i.e., steel drum, plastic-lined wooden box, etc.) in which each hazardous material may be shipped; the shipper decides which one to use. However, selecting one of the allowed packages is not a simple task. To achieve this goal, the shipper must sift through hundreds of regulations, search large tables, and perform calculations, depending on the applicable regulations. Only when all the applicable information has been determined, depending on the regulations and the tables, can the shipper choose a package type. These regulations are extremely complicated. Transportation engineers have the expertise to know which regulations to apply to a given shipment. It is this knowledge—that of the correct path through the regulations—that this expert system will contain.

Figure 1 illustrates the five different stages necessary to determine the type of package required for transporting a given radioactive material. Stage 1 of the logic diagram simply represents the data

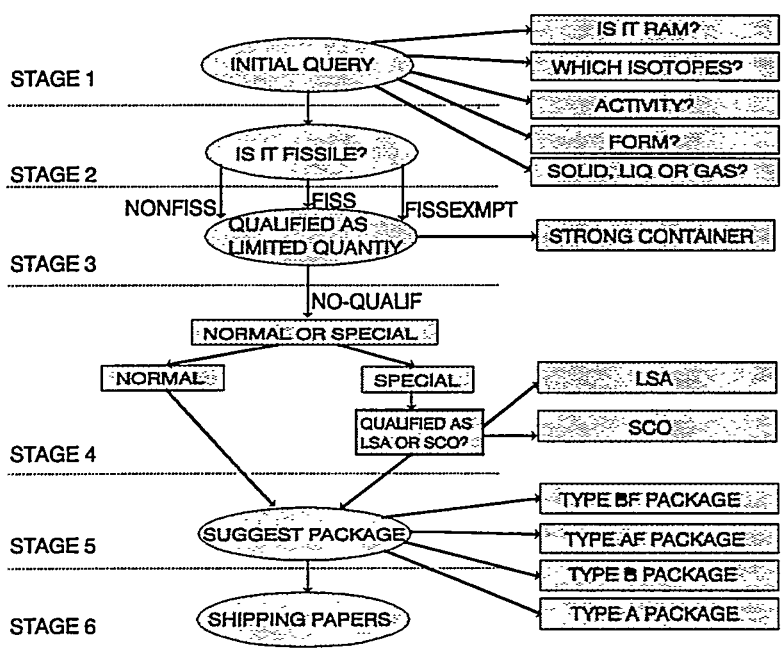

Fig. 1. Six stages of the expert system.

input required from the user. At this point, the user must indicate whether the material conforms to the definition of radioactive material. In addition, the user must supply information about the isotopes that comprise the radioactive material [i.e., how much activity per isotope, the form of the material (special or normal), and the physical state of the material (solid, liquid, or gas)]. Along with this information, stage 1 of the diagram contains enough data for the program to calculate the fraction A_1 (if material is special form) or the fraction A_2 (if material is normal form).

The boxes of stage 1 and stage 4 on the logic diagram represent decision points where the system requires information from the user. Hence, enough information has been provided by the user or enough facts have been matched with the rule system to reach a decision.

Stage 2 of the logic diagram determines whether the material to be transported is fissile, nonfissile, or fissile exempted. Stage 3 of the logic diagram determines if the material qualifies as limited quantity (package requirements vary dramatically depending whether it qualifies as limited quantity or not).

Stage 4 determines, for those materials that have not qualified as limited quantity and have been declared as normal, whether the material can be shipped as low specific activity (LSA) or surface contaminated object (SCO).

Stage 5 of the logic diagram determines the type of package based on information provided by steps 1 to 4. Package types can be Type A, Type A fissile, Type B, and Type B fissile at this point. Stage 6 prepares the shipping order.

2.3.2 Coding the Prototype

2.3.2.1 Coding the DOS Prolog Version

The logic diagram was first developed as a rule-based system using Prolog. PDC Prolog™ is produced by Prolog Development Center. PDC Prolog is a declarative language, which means that, given the necessary facts and rules, PDC Prolog will use deductive reasoning to solve programming problems. PDC Prolog has several features that make it very attractive for expert system development. The rules can be built using conventional **IF, THEN** clauses. In order to prove a particular hypothesis, the system will backtrack to its original conditions, prove them, and then make the hypothesis true thus activating the actions that evolve from the hypothesis.

Other features of PDC Prolog include simple data base handling commands. Data bases written in PDC Prolog can be stored in expanded memory, in conventional memory, or on a hard disk. The last option of keeping the data base on a hard disk may represent one of the outstanding features of PDC Prolog, although this type of storage does delay the retrieval of information for a few moments. Translating the advantage of this feature into practical terms, the amount of information that may be held in the data base is limited only by the size of available disk space.

PDC Prolog can also access text files that may hold massive amounts of information. These text files can also reside in the hard disk. A word or word group search routine can be easily set up to configure the text files. PDC Prolog works with a graphical environment and can handle graphics produced by Borland Graphics Interface (BGI) commands or bit images with GIM format.

Currently, one disadvantage of PDC Prolog is its 640K working-memory ceiling, which is typical for a DOS application. Another current disadvantage of PDC Prolog is the type of graphics that can be directly accessed by the program. The GIM format, in which it accesses bit images, is not extensively used by other software packages or hardware systems (such as a scanner). However, a program that takes images directly from a scanner to be later translated into Prolog commands, has been written as part of this project. Navigation through the regulations using hypertext capability is a feature that could not be implemented using Prolog. Video clips can be accessed by PDC Prolog using VideoLogic multimedia interactive control (MIC™) software, but cannot be displayed in a resizable window. The final executable was compiled by combining the individual Prolog programs representing each one stage of the logic diagram and the C program required to run the graphics files.

2.3.2.2 Coding the Guide-Prolog Version

GUIDE™ is multimedia software created by OWL International that is implemented in Windows 3.0. GUIDE software allows the user to navigate through large amounts of material to locate answers quickly. Interacting with information through associative links, users can follow their trains of thought and view any level of detail they desire. By activating a pointing device (such as a mouse) on specific objects, users access cross-referenced materials, footnotes, reports, articles, graphs, full-color illustrations, spreadsheets, video and audio messages, etc.

A built-in programming language, LOGiiX, gives the user control of all GUIDE functions, thus allowing for the development of "smarter" documents that will simplify reader interaction. These features include comprehensive document formatting; graphical user interface; flexible link structures; text, graphics, video, and sound versatility; and search and object management tools.

LOGiiX commands allow for the execution of the Prolog rule-based programs developed during the proof-of-concept stages. This is a great advantage because of the fair amount of code required for the reasoning mechanism. In addition, Guide uses a hypertext mechanism. This is a novel approach that is becoming very popular for new text-search mechanisms. This mechanism consists of creating a special area surrounding a text (e.g., a word, a sentence, or a paragraph). It is mouse- driven; so by clicking on a "hyperarea" that represents a regulation, for example, the user is presented with the contents of that regulation. These hyperareas not only connect the text with other pieces of text, but also they can connect the user to different types of actions, such as running other programs or bringing other types of information to the screen. By clicking on a button, a text file containing a particular regulation may be brought onto the screen. As noted previously, a regulation text file may contain reference to another regulation. If the user clicks on the hyper area text that represents another regulation, the text corresponding to that regulation is brought onto the screen in another window for the user to navigate through.

Access to video clips is possible by using Guide, VideoLogic MIC, and Guide VIDEO TOOLKIT™ (GTV). The use of MIC was needed when full-motion video was added to the expert systems. The system used a frame-addressable video laser-disc player. This type of device allowed instant access to a specific motion sequence using GTV. To implement the solution to the problem of video images GTV was purchased. GVT, is a software package that allows the developer easily to add full-motion video to the documents. Video sequences are developed inside GVT and then saved in a file that is accessed by the Guide document when the video command is called. These sequences are easily developed by giving a starting and ending frame. This is accomplished using a screen, as illustrated in Fig. 2.

The construction of an expert system using Guide may require additional programming skills from the developer. Distribution of the final application will require the purchase of the Guide Reader, the

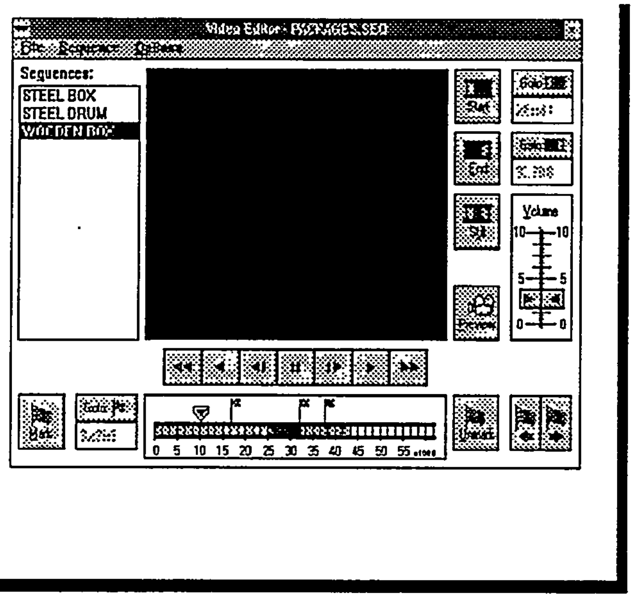

Fig. 2. Video sequencer.

price of which varies depending upon the number of copies desired. If the application includes video images, the system owned by the final user must have video and graphic interface hardware similar to that of the computer where the application was developed. Since Guide is a Windows application, every time it accesses the DOS Prolog executable, a noticeable flickering on the computer screen occurs which can cause concern and/or annoyance in the user.

2.3.2.3 Coding the Guide-Visual Basic version

DOS Prolog code is called up by a LOGiiX command. However, the Prolog executable is a DOS application rather than a Windows application. The transition between DOS and Windows is not a smooth action, and flickering is observed during the process. Microsoft Visual Basic is a Windows-based package that allows the programmer to write codes using the BASIC™ programming language. An executable file is created which is also called by a LOGiiX command. However, this executable is already a Windows application, and the transition between one Window screen and another is invisible to the user.

The major disadvantage of producing this version was the need to recode the reasoning mechanism in Visual Basic. The same logic was used, but the rule-based system had to be translated into Visual Basic. Translation of the code into Visual Basic was a lengthy process, but it allowed access to Windows features with very little additional programming effort.

2.4 VALIDATION OF THE PROTOTYPE

Validation of the prototype began with experts evaluating the model against the regulations. Any discrepancies were resolved, and the logic model was modified, as necessary. The second part of the validation consisted of a number of tests which compared model output with the results of manual selections.

3. ANALYSIS

The feasibility analysis of the radioactive materials transportation expert system prototype produced positive results in that it was concluded that the pertinent regulations can be translated into a logic diagram and that this logic diagram can be translated into a computer language. In addition, it was concluded that for presentation purposes, better memory utilization, and a larger portfolio of computer features, the radioactive materials modules could be best be developed completely in Windows 3.1 applications.

A QA plan was developed and implemented for the validation and verification of the expert systems. The purpose of this verification plan is to establish specific responsibilities and methods for the validation and verification of the packaging expert program (RaMTES) in order to do quality work to support the Office of Environmental Restoration and Waste Management, Transportation Management Division (EM-561). Several cases were analyzed using the expert system. Cases in which the answers were not correct were useful for checking the isotopes data base values. The data base has 390 different isotopes, and corrections were made for the errors found during the validation process. Other errors found were related to an incomplete set of rules by which the program makes decisions. Rules were completed accordingly.

4. FUTURE WORK

The feasibility analysis of the hazardous materials prototype is the next step in developing a global expert system to determine appropriate packaging for the transport of radioactive and hazardous materials. Based on results of this study, it was decided that it would be best to develop the radioactive and hazardous modules in one specific computer language. This would eliminate the need for additional fees for run-time versions and improve the programmer's control on the available features. Because Visual Basic offers these elements, it will be used for development purposes.

Preliminary work for this project using Visual Basic indicates that the radioactive and the hazardous materials modules may be linked successfully. The linking process is not as straightforward as initially thought; however, in a few instances it is imperative to move back and forth between the radioactive module and the hazardous materials module. This points out a need to develop some intermediate linking locations on both modules. This is another reason why it would be better to develop the global expert system using only one computer language.

Work is presently underway to define an appropriate architecture where both the radioactive and hazardous materials modules can reside. Linking locations are being defined at this moment. Appropriate pull-down menus that bring additional information to the user are being designed. Additional information includes access to hypertex, graphic, and video files.