GEORGE MASON UNIVERSITY INTELLIGENT EDUCATOR[1]

E. Oguejiofor, R. Kicinger, E. Popovici, T. Arciszewski, and K. De Jong

eoguejio@gmu.edu; rkicinge@gmu.edu; epopovic@gmu.edu; tarcisze@gmu.edu; kdejong@gmu.edu

School of Information Technology and Engineering, George Mason University, 4400 University Drive MS 4A6, Fairfax, VA, USA 22030

ABSTRACT

Engineering education is undergoing significant changes, mostly driven by the ongoing Information Technology Revolution. One of the most promising technologies becoming available to educators is that of intelligent software agents. Such agents can be used in building tutoring systems, which in this way become intelligent in terms of knowledge content, their ability to learn (acquire knowledge), and to adapt to the needs and learning preferences of their users.

The paper discusses the development of a prototype intelligent tutoring system, called "George Mason University Intelligent Educator." It has been built as a result of a complex process in which various software development tools have been used in a novel way, including ConceptMap, Protégé2000, Macromedia Flash and JRun. The reported research is part of a NASA-sponsored project in the area of engineering education called "Hierarchical Learning Network" and is coordinated by the Old Dominion University in Virginia. The primary objective of the research at George Mason University is to develop a methodology for building intelligent tutoring systems and to demonstrate its feasibility.

First, the paper provides a brief description of the process that was used for knowledge acquisition and its representation in the form of an ontology. The description includes examples from the area of personal air vehicles. Next, the development process for the proposed intelligent tutoring system is presented, along with a discussion of the integration of various software development tools. Finally, the initial research conclusions and plans for further research are discussed.

Keywords: Intelligent agent, intelligent tutoring system, knowledge base, ontology.

[1] *Citation:*
Oguejiofor, E., Kicinger, R., Popovici, E., Arciszewski, T., and De Jong, K. A. "George Mason University Intelligent Educator." Innovative Developments in Architecture, Engineering and Construction. Proceedings of the 2nd International Conference on Innovation in Architecture, Engineering and Construction, Loughborough University, UK, 25 - 27 June 2003, C. J. Anumba, ed., Millpress Science Publishers, Rotterdam, The Netherlands, 355-366.

INTRODUCTION

An agent generally refers to an entity that acts on behalf of another entity. An agent differs from a computer program in that, in addition to being able to accept tasks from an external source, it often has some measure of intelligence built into it and is thus able to take initiatives independent of external controls. "Intelligent agents" are autonomous or semi-autonomous entities that are capable of learning the preferences of the user and adapting to the environment (Skolicki and Kicinger 2002). Skolicki and Arciszewski (2003) distinguish a computer program from a software agent by describing the former as having a sequence of instructions while the latter possesses a constant loop of intelligent interactions with the environment.

Over the years, various agents have been developed with varying degrees of intelligence. The development process is being accelerated by the Information Technology Revolution and agents are being applied in a variety of engineering fields. Architects and designers have moved from the use of simple CAD systems and 3D modeling tools to the use of Virtual Design Studios and research efforts are geared towards the development of 3D Collaborative Dynamic Virtual Worlds that will utilize intelligent design agents (Maher and Gu 2002). Agents are also being used in the construction industry to organize and manage projects (Ugwu et al. 2002). These agents are typically integrated with other technology tools such as video conferencing, the Internet, the World Wide Web, etc.

The paper reports results obtained at George Mason University (GMU) as part of a NASA-sponsored project in the area of engineering education called "Hierarchical Learning Network," which is coordinated by the Old Dominion University in Virginia. The GMU effort is focused on developing intelligent agents that capture the knowledge of domain experts and present the knowledge to domain users in an effective and efficient manner. The primary objective of the research at George Mason University is to develop a methodology for building intelligent tutoring systems and to demonstrate its feasibility. The secondary objective of the reported research is to provide NASA engineers with an experimental tool called "George Mason University Intelligent Educator." The feasibility of the tool is being evaluated in the area of Personal Air Vehicles (PAVs). The knowledge to be used in the tool under development results from NASA's Small Aircraft Transportation System (SATS) program (NASA SATS homepage).

The paper provides a brief description of how the knowledge to be used in the GMU Intelligent Educator is acquired and structured in the form of an ontology. The prototype intelligent tutoring system and its development process are then presented, including the discussion of the system integration process and of the tools used. Finally, the initial research results are presented and plans for further research are discussed.

KNOWLEDGE ACQUISITION AND STRUCTURING

The first issue addressed by the research team was how to acquire, represent, and ultimately use the domain knowledge in a meaningful way. After extensive studies, we decided to use an ontology as an optimal knowledge representation form for our purposes. An ontology in its simplest form consists of hierarchical knowledge structures. The science of ontology deals with the study of the general concepts and abstractions that make up the fundamental aspects of our world (Noy and McGuinness 2001). Gruber (1993) defines an ontology as an explicit formal specification of the terms in a domain and the relations among them. An ontology allows information providers, users and software agents to establish a shared understanding of the definitions and descriptions of common terms and vocabularies in a particular domain (Musen 1992). Furthermore, the structuring of knowledge in an ontology facilitates the reuse of domain knowledge and the explicit specification of domain assumptions. Ontologies provide a convenient means for separating domain knowledge from operational knowledge. Thus the operational knowledge may be encoded in a software agent that can be applied to different knowledge domains (McGuinness and Wright 1998; Rothenfluh et al. 1996). We have found this separation of knowledge important and useful from the development point of view, and we have implemented this separation in our George Mason University Intelligent Educator.

After investigating a number of knowledge management packages, the ontology editor, Protégé2000 (Noy et al. 2000), developed at Stanford University, was chosen and used to capture and structure the knowledge base. This system comes with an easy to use set of tools for building, visualizing and editing ontologies, as well as capabilities for importing from and exporting to different ontology formats. Its open-source Java API allows dynamic ontology access. Also, its architecture is extensible, and various new plug-ins are continuously being developed by a large and active user community.

Prior to the choice of Protégé2000, the ConceptMap software (Coffey et al. 2002), developed at the Institute for Human and Machine Cognition, University of West Florida, was used to lay out ideas and concepts pictorially and show how these connect with one another. For example, Figure 1 shows a concept map that was developed in the process of structuring the domain knowledge. This map depicts the taxonomic hierarchical arrangement of the classes and subclasses for a part of the ontology. Multimedia resources (text files, HTML files, video clips and graphic files) were used as values for some of the slots. However, unlike Protégé2000, the concept maps cannot be used for dynamic access to the knowledge or for reasoning.

GEORGE MASON UNIVERSITY INTELLIGENT EDUCATOR

The research presented in this paper has two major interrelated components. The first is the development of a novel methodology for building intelligent tutoring systems for engineering education; this component is being accomplished by integrating state-of-the-art computer science techniques. The second component is the development of an actual tutoring system, George Mason University Intelligent Educator, to be built using this methodology in order to prove its feasibility. As previously mentioned, the application area used was that of personal air vehicles (PAVs).

To derive such a methodology, we began with an extensive research effort to analyze existing tools and technologies to find the best ones for tackling the following tasks: representing knowledge; acquiring, visualizing and updating knowledge; integrating knowledge; constructing a graphical user interface that is user friendly, intuitive and incorporates multimedia for a more effective learning experience; and finally for building intelligent tutoring systems. The methodology was developed keeping in mind goals such as platform independence, modularity and Web-based access.

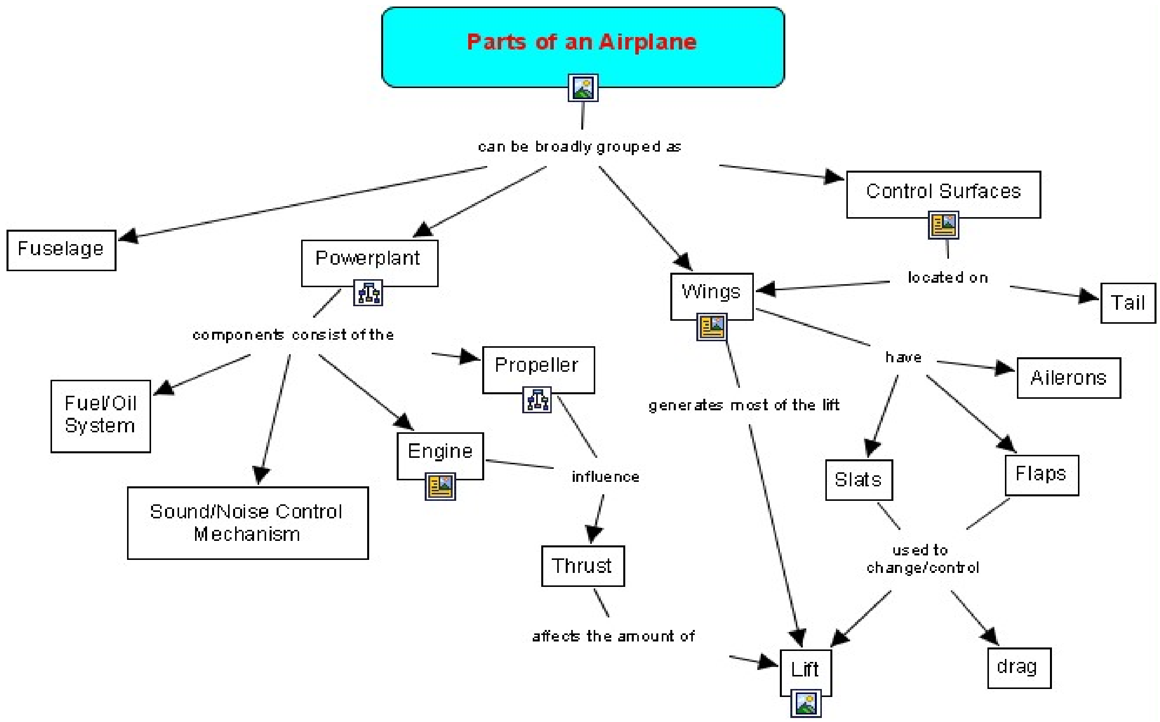

Figure 1. Structure of Part of the Knowledge Base

While the visualization capabilities of Protégé may be sufficient for a knowledge engineer, they are not suitable for an end-user trying to learn a domain. This led to the need to develop a user-friendly graphical user interface that would seamlessly present the domain knowledge to the end-user. Subsequently, the development of such an interface required the integration of the knowledge base produced by Protégé with visualization software. This task has been accomplished by using a set of Macromedia products for the task of integrating the knowledge base and various visualization means. Ultimately, Macromedia Flash turned out to be an excellent tool for developing the graphical user interface (GUI) front end. Also, the Macromedia JRun application server has been used as the middleware that connects the GUI with the back-end knowledge base. In this case, the Flash graphical interface dynamically invokes the middleware through another Macromedia technology called Flash Remoting. Flash Remoting is basically an Action Script (Flash's Scripting language) API for transparent, remote method invocation (RMI). On the other side, the JRun server accesses the knowledge base by forwarding the calls to a Java object built on top of the Protégé API. Finally, the JRun Server was chosen for its smooth integration with both Action Script and Java.

The graphical user interface in the form of a Flash movie has been embedded in an HTML Web page and deployed on a Web server from which it can be served to anyone with access to the Internet. As the user interacts with it, calls are made through the JRun server and information is extracted from the knowledge base and sent back to the Flash movie which dynamically displays it in the form of text, images and movies. Additionally, support for Microsoft Agent technology has been added for Windows platform users in order to enhance the learning environment with speech interaction. The schematic architecture of the George Mason University Intelligent Educator is presented in Figure 2.

In an offline process, a Knowledge Engineer works with the Protégé tools to build an ontology and subsequently update it. The ontology can then be accessed by the user through the system and its contents displayed to the user. The system is independent of the actual contents of the knowledge base and this fact guarantees the independence of the actual application domain.

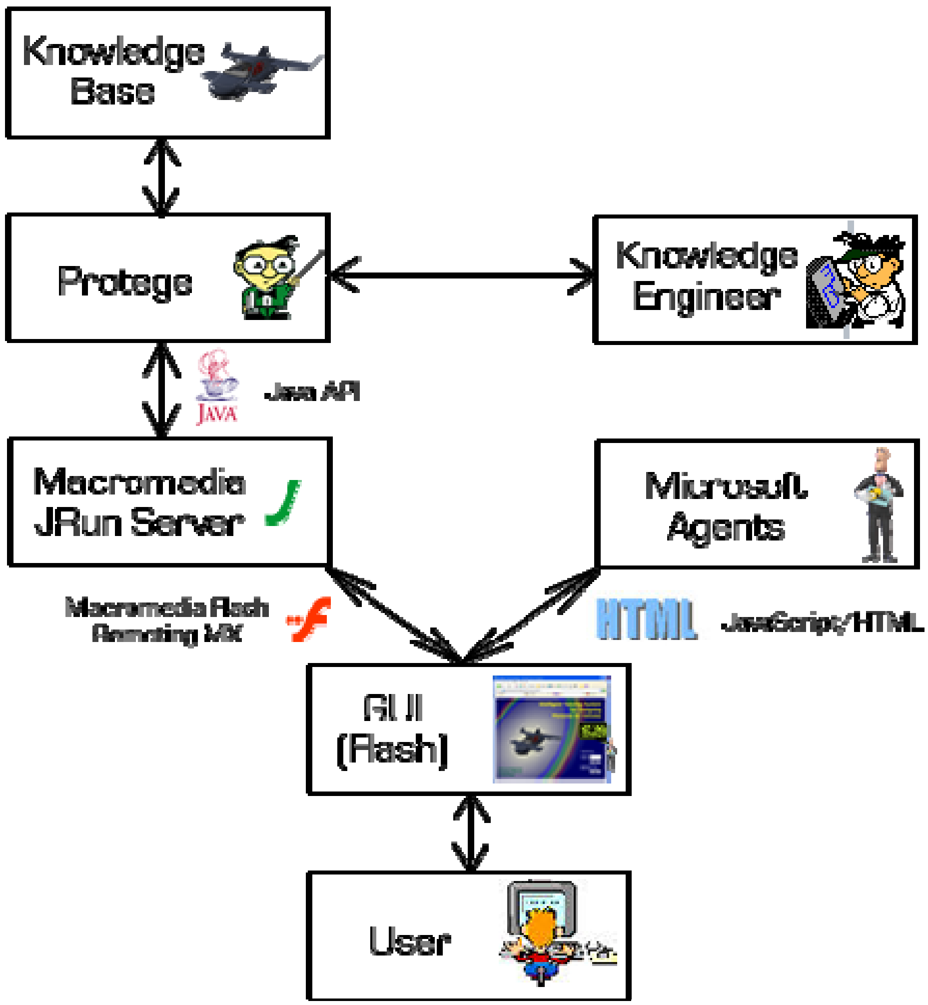

Figure 2. Architecture of the George Mason University Intelligent Educator

The developed system is modular, allowing for future replacement of various components as long as they comply with the current interfaces. Platform independence is provided at various levels:

- Back-end
    - ontology building: because the Protégé software is a Java application;
    - ontology access: because it is done through a Java interface;

- Middleware: because JRun is available for several platforms or can be replaced with a different compatible application server such as Macromedia ColdFusion, for example;

- Front-end: because it is Web-accessible and Flash Player plug-ins are available for a multitude of browsers and platforms.

Using the presented methodology and architecture, an actual tutoring system has been built as a "proof of concept" demonstration for the domain of Personal Air Vehicles (PAVs). Effort has been put into the study of this domain, resulting in a preliminary ontology that is continuously being refined.

The system was designed to support multiple users, as can be seen on the snapshot presented in Figure 3 of the welcome page of the tutoring system's GUI where you will also notice the Microsoft Agent technology.

Figure 3. Welcome Page on the Tutoring System

Two modes of operation have been envisioned for the system: guided (through tutoring scenarios) and autonomous (browsing). The current version only implements the browsing mode, but in the near future, both modes will be available and the user will be able to switch between them at any time.

In the browsing mode, the users can navigate on their own through the concepts present in the ontology, either by typing concept names or by clicking on the boxes showing concepts other than the current one (see Figure 4). Small snapshots of the hierarchical structure can be seen as each concept (central) is shown together with its super-concepts (above) and sub-concepts (below). Other related concepts are also shown to the right, together with a text description. The lower part of the screen displays images and movies.

Figure 4. Multimedia Learning Environment

CONCLUSIONS AND FURTHER WORK

Our initial research has clearly demonstrated that building intelligent tutoring systems that utilize the concept of intelligent agents is feasible. However, the development process is still much more complicated, difficult, and time consuming than initially anticipated. At present, there is not a single development tool that could be used for building an entire intelligent tutoring system. Currently, therefore, an intelligent tutoring system can be developed only by a team comprised of engineers (domain experts), knowledge engineers, and computer scientists. Unfortunately, the optimal use of a combination of various tools and the integration of their products require deep understanding of computer science and extensive programming. This leads to a long, costly and complicated development process in which careful coordination of the efforts of the individual team members is required. However, all these reported problems could be overcome if a significant effort is undertaken in building a tool for the development of intelligent tutoring systems. The development of such a tool could significantly reduce the costs of building intelligent tutoring systems and could lead to the expansion of their use.

The research team intends to continue the project. During the second year, the developed intelligent tutoring system will be thoroughly tested in cooperation with the NASA experts. Also, the content of the system will be expanded and modified, if necessary. More importantly, the system will be integrated with Inventor 2003/G, an experimental design tool, developed in the Information Technology and Engineering School at George Mason University. Inventor 2003 is a multi-population evolutionary design system developed in Java and intended for use over the Internet. It is a generic system that can be integrated with various simulation packages to be used for the evaluation of the generated designs. The integration of Inventor 2003/G with the George Mason University Intelligent Educator will allow the user not only to learn the conceptual knowledge about PAVs, but also to develop some understanding of the design concepts and various notions related to the domain being studied.

The reported research has clearly revealed many developmental challenges related to intelligent tutoring systems. On the other hand, it has demonstrated the feasibility of building such systems and has shown the potential educational benefits of their use in an engineering environment. Most likely we will witness the emergence of various intelligent tutoring systems in the near future and that would mark a paradigm change in engineering education.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support for their research from the NASA Langley Research Center under the Grant 01-1231.

REFERENCES

Coffey, J. W., Hoffman, R., Cañas, A. J., and Ford, K. M. (2002) “A Concept Map-Based Knowledge Modeling Approach to Expert Knowledge Sharing” in Proceedings of the IASTED International Conference on Information and Knowledge Sharing, November 2002, Virgin Islands.

Gruber, T.R. (1993) “A translation approach to portable ontology specification”, Knowledge Acquisition, Vol. 5, pp. 199–220.

Maher, M.L. and Gu, N. (2002) “Design agents in virtual worlds” in Agents in Design 2002, Gero J.S. and Brazier F.M.T (eds.), Key Centre of Design Computing and Cognition, University of Sydney, pp. 23 - 38.

Musen, M.A. (1992) “Dimensions of knowledge sharing and reuse”, Computers and Biomedical Research, Vol. 25, pp. 435 - 467.

McGuinness, D.L. and Wright, J. (1998) “Conceptual modeling for configuration: a description logic-based approach”, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Special Issue on Configuration.

NASA Small Aircraft Transportation System (SATS) homepage, http://sats.larc.nasa.gov/main.html.

Noy, N. F., Fergerson, R. W. and Musen, M. A. (2000) “The knowledge model of Protege-2000: Combining interoperability and flexibility” in Proceedings of the 2nd International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France.

Noy, N. F. and McGuinness, D.L. (2001) “Ontology development 101: a guide to creating your first ontology”, Knowledge Systems Laboratory/Stanford Medical Informatics Report No. 2001-0880, Stanford University.

Rothenfluh, T.R., Gennari, J.H., Eriksson, H., Puerta, A.R., Tu, S.W. and Musen, M.A. (1996) “Reusable ontologies, knowledge-acquisition tools, and performance systems: PROTÉGÉ-II solutions to Sisyphus-2”, International Journal of Human-Computer Studies, Vol. 44, pp. 303 – 332.

Skolicki, Z. and Arciszewski T. (2003) “Intelligent Agents In Design”, ASME Conference On Design, (Submitted).

Skolicki, Z. and Kicinger, R. (2002) “Intelligent agents for designing steel skeleton structures of tall buildings” in Computing in Civil Engineering, Songer A.D. and Miles J.C. (eds.), Proceedings of the International Workshop on Information Technology in Civil Engineering, Washington, D.C. Nov., pp. 25 – 35.

Ugwu, O.O, Arciszewski, T. and Anumba, C.J. (2002) “Teaching agents how to solve design problems: a mixed initiative learning strategy” in Computing in Civil Engineering, Songer A.D. and Miles J.C. (Eds.), Proceedings of the International Workshop on Information Technology in Civil Engineering, Washington, D.C. Nov., pp. 11 – 24.