

Conference Proceedings

1991 IEEE

International Conference

on

Systems, Man, and Cybernetics

"Decision Aiding for Complex Systems"

October 13-16, 1991

Omni Charlottesville Hotel

and the

University of Virginia

Charlottesville, Virginia



IEEE

Volume 3

91CH3067 - 6

Using Neural Networks and Expert Systems to Model Users in an Object-Oriented Environment

C. Chiu, A. F. Norcio, and K. E. Petrucci
Department of Information Systems
University of Maryland
Baltimore, MD 21228-5398

Abstract—User modeling offers an effective technique for determining user requirements and for enhancing the relationship between a user's cognitive abilities and software sophistication. Specifically, this study discusses the usefulness of neural networks and expert systems for modeling users in an environment in which information about software is retrieved. This study also describes the framework of prototype implementation which is an object-oriented intelligent consulting system for selecting appropriate problem-solving software.

I. INTRODUCTION

Information retrieval is one of the most fundamental and frequently used activities in information systems. However, identifying appropriate information for the user is no longer the only requirement for effective and successful retrieval systems. As systems design focuses more intensely on the user, the system's capabilities for satisfying user needs is becoming the overriding criteria for determining system effectiveness.

Smith and Vokurka have underscored the importance of this criteria by noting that software that is not compatible with a user's mental characteristics probably can not satisfy the user's goals [24]. Vessey and Galletta also indicate that a user's problem-solving skills have a significant influence on the effectiveness of both the task solution and information acquisition [26].

Tyler and Treu mention several problems that frequently arise with traditional information retrieval systems such as:

- 1) retrieved information may be inappropriate to the user's needs;
- 2) the user may have insufficient knowledge to classify the problem domains;

- 3) the user may not adequately recognize information needs; and
- 4) a naive user who is not familiar with the keywords may find it difficult to express the problem requirements.

The purpose of an interface that is capable of modeling a user is to infer categorical information about that user. The purpose of classification in user modeling is not the classification itself but the use of that classification to predict future user behavior and effective interaction [1]. Information that is needed to classify the user should include cognitive style information about the user, information about the user's mental models of the application and the system, as well as the user's ultimate goals [20,25]. These concepts make up the user model. In this paper user models and their current problems are addressed.

II. USER MODELING

Information is perceived subjectively by individuals. That is, different people typically have different points of view given the same information. This is generally attributed to a cognitive style that is unique to an individual.

It has recently been recognized that in order to design effective systems, it is essential to accommodate these individual characteristics. These characteristics are used to construct a user model [22]. A user model, can be defined as system's knowledge of the user.

User modeling may be critical for satisfying a user's goal. For a system to identify software that is appropriate for an individual and the task, a user's cognitive characteristics such as education level and technical skill should be considered in the selection

strategy. This is best accomplished through user model.

Gutierrez reinforces this point by noting that techniques for selection requirement analysis must consider the role, purpose, and skill level of the user [14]. Brajnik *et al.* further underscore the importance of these considerations in retrieval systems. They point out that constructing a user model requires information about the user's professional background, position, and experience level [2]. Eberts and Eberts support this contention as well [7]. Nunamaker *et al.* add additional support for this position with respect to software selection systems. They specifically mention that such systems, which classify users into stereotypes, can help [21]:

- 1) identify the necessary level of assistance,
- 2) narrow the search solution, and
- 3) determine user requirements.

Clearly there is ample support and evidence that user models enhance the effectiveness of interfaces to advanced systems, especially systems that involve complex retrieval tasks. However, in spite of the intuitive appeal of user models and supporting experimental evidence for them, there still remains several fundamental and difficult issues that need to be addressed.

A. Problems Associated with User Models

The capability of predicting human behavior accurately is controversial. Allen questions how much human behavior can be predicted even with sufficient information [1]. Moreover, there are several major limitations associated with a user model that focus on the model's inherent characteristics. Specifically, a user model tends to be [2, 22]:

- 1) incomplete,
- 2) uncertain,
- 3) ambiguous,
- 4) unstructured,
- and 5) unstable.

One of the primary reasons for these characteristics of a user model is that information used to construct a user model is necessarily based upon the user's mental models. The user has a mental model of the task as well as a mental model of the system. The contents of a user model must take these mental models in consideration as well as the user's general level of knowledge.

Users are typically uncertain about their knowledge and this may result in incomplete, inaccurate, or imprecise mental models.

Further, as Wahlster and Kobsa note, a user model is frequently constructed by forming inferences that are based upon assumptions [18]. These assumptions are based upon inferences that are built through pre-defined rules, which are domain dependent. Frequently these pre-defined rules might be inappropriate for any specific user performing any specific task. In other words, the inherent domain dependencies of the rules can not describe or predict the great variety of human behavior.

With the problems and issues that surround conceptualizing user models, it seems appropriate to explore alternative and novel technologies for developing more accurate and complete user models.

B. Expert Systems and User Models

There are several facets in expert systems that may be inappropriate for modeling users. They include:

Problem Representation: systems work best when rule-based task analyses are available or easily derived. However, expert systems do not support tasks that are poorly understood or defined. Expert systems also tend to falter when conflicting issues are involved, or when the relevant expertise cannot be demonstrated or explained [4]. User modeling is a process of pattern identification and classification. The assumption of the existence of the heuristic (i.e. rule of thumb) that maps a user's characteristic to pre-established stereotypes seems to need more validation.

Maintenance: time-consuming and expensive tasks in developing an expert system [11]. Also, it is difficult to verify the effectiveness of a large knowledge base under continual maintenance. Especially, when coupling of problem and solution domains in a pure rule-based system, this makes the task of maintenance more difficult [10]

System Overhead: has an underlying (static or dynamic) user model that is either explicitly or implicitly stated [8]. A major concern with the traditional

expert systems for modeling a user dynamically is that they are computationally expensive [9]. In order to obtain a more accurate user model in a real-time mode, those approaches that can provide better system efficiency seems to be better recommended.

Fuzzy Reasoning: As Kobsa notes, "the inferences that a system equipped with a user modeling component can draw about the beliefs, goals, and plans of the user are associated with varying degrees of certainty" [17]. Management of uncertainty within incomplete and subjective information plays an important role in the design of an expert system. One of the most important criteria for evaluating an expert system is the correctness of its reasoning techniques [13].

Since the nature of the information used by an expert system is essentially incomplete, ambiguous, and uncertain, it is not clear that these traditional approaches provide sufficient richness to handle the degree of uncertainty or fuzziness that is usually present. The true-false structures of expert systems are unable to represent uncertainty about facts and rules [19].

In his classic paper, Zadeh points out the inherent fuzziness of information [27]. Under the problem situation of nonstatistical uncertainty, a fuzzy axiomatic structure for dealing with such problem usually increases both its mathematical tractability and physical realism [16].

C. Neural Networks and User Models

Recently, neural network is gaining its popularity in data-intensive pattern recognition application partly because of its inherent characteristics in solving incomplete and imprecise data.

Due to its parallel processing capabilities that provide a better system efficiency, neural network is especially useful in a real-time control environment.

Also, its strong learning capability that can generalize from specific examples to principles enables neural network to be as a means of efficient knowledge acquisition which is believed the bottle neck for building a traditional expert system [6].

Neural network is being considered appropriate in modeling user with the following aspects:

a) **Stereotype Identification:** As Kaiser notes that classifying command, plans, and goals according to the level of expertise is inappropriate and that global categorization of users as "novice," "intermediate," or "expert" is inadequate [15]. Traditional two-value logic (0 or 1) in categorizing users into one of the pre-defined independent stereotypes seems inappropriate. Instead, classification scheme that supports user's grade of membership from various sets of stereotypes would be more adequate for modeling user in terms of the nature of user characteristics and human reasoning process.

Neural network's capability dealing with situation that involves fuzzy information is thought to be more suitable than traditional expert system.

b) **Dynamic Modeling:** Kaiser recommends that expectations about what the user knows and should be told is based on the tasks that the user has completed in the past, rather than on broad ad hoc classification of commands [15].

Thus, to obtain the up-to-date user model, an pattern recognition reasoning techniques that can dynamically monitor the dialog behavior is required. Neural network can improve system efficiency than expert system does in this type of real-time adaptive control.

c) **Conflict Resolution:** One of the most important concerns in an user modeling system is its inconsistency problem. Several approaches such as truth maintenance system, nonmonotonic reasoning, and certainty factors have been raised to solve this problem. These approaches usually are embedded within the their global knowledge base. This coupling might not only increase system overhead, but also the maintenance difficulty. Caudil recommends the use of fuzzy mathematics to build a neural network which can handle the conflicts of rules and situation that involves the disagreement of inference rules from multiple experts [3]. A neural network can be trained to learn the fuzzy rules to reduce the inconsistency problem and to increase system efficiency.

III. RESEARCH FRAMEWORK

In this paper, we propose a hybrid architecture in which expert system is used for solving well-defined problem domains and interpret the inferred results;

neural network is applied to recognize the user's pattern of stereotypes dynamically and solve the reasoning conflicts.

To achieve a higher interaction efficiency and simplifying the maintenance of knowledge base with these two heterogeneous reasoning processes, object-oriented environment is proposed as an integration platform.

Figure 1 is the proposed conceptual framework which includes three components: rule-based subsystem, neural network subsystem, and the knowledge base for supporting the inference.

The advantages of this integration are aimed to support the following functionalities: a) Typifying users that involves fuzzy diagnosis, b) Adaptive modeling of the users, c) Reducing system overhead, and d) Suggesting a suitable list of software packages.

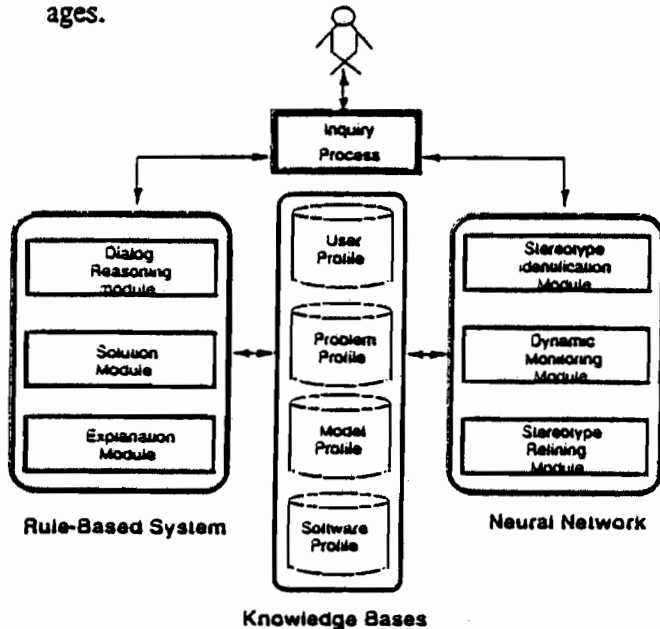


Figure 1

Knowledge Base: A variety of knowledge in our software retrieval environment can be categorized as follows: a) user profile - knowledge of user, b) problem profile - knowledge of problem, c) model profile - model information, and d) software profile - characteristics of software packages.

Rule-Based Subsystem: The major components of rule-based subsystem are: a) dialog reasoning module - controls dialog, b) solution module - infers solution, and c) explanation module - explains selections.

Neural Network Subsystem: This subsystem includes three modules: a) stereotype identification module - identifies stereotype b) dynamic monitoring module - maintains the accuracy of user model, and c) stereotype refinement module - resolve inconsistency and refines user model.

The function of the expert system technology is two-fold: a) initiate intelligent question- and-answer sessions with the user and b) explain the reasoning process for categorizing users and providing appropriate solutions or recommendations. The neural network technology classifies the user's characteristics based upon information obtained during the question-and-answer sessions. An object orientation provides an appropriate environment for integration.

IV. IMPLEMENTATION OF THE SYSTEM

There no OR/MS package will be perfectly suited to all user's needs. For any given needs, there could be a number of software packages that can perform adequately. As user's needs are organized into abstract form which could be identified by the system, an intelligent choice of OR/MS software package can be made. A careful assessment of their individual suitability for a particular user needs (user requirement and personal characteristics) is important for making a optimal selection.

In order to transform the user needs into the system recognized representation, an intelligent retrieval system is needed to facilitate the requirement elicitation process.

The detailed processes are described as follows:

1. Identification of user's level of problem solving skill: User's relevant information about experience, familiarity, and preference with the problem solving procedures are collected through question-answer mode.
2. Identification of problem domain: Some typical practical examples are provided in windows by which user can browse the contents to locate the closest example related to his problem in scrolling bar menu. If the user cannot find out the representative example, then keyword search is provided to further explore the search. User model(level of

expertise and personal information) will be needed for refining his needs.

3. Identification of task domains: Information about the size and complexity of task is needed for matching the software package(s). User model will be referenced when providing keywords search facility if needed.

V. DISCUSSION

The proposed design integrates expert systems technology with neural networks in an object-oriented approach that can enhance the proposed user model as follows.

Providing a framework better suited to represent fundamental components of user models: One of the key elements of object-orientation is abstraction where abstraction is "a simplified description, or specification, of a system that emphasizes some of the system's details or properties while suppressing others. A good abstraction is one that emphasizes details that are significant to the reader/user and suppresses details that are, at least for the moment, immaterial or diversionary" [23].

Abstraction will represent the essential characteristics of users that are needed to accurately design user models as they become more apparent.

Increasing the power of expert system to provide accurate solutions: Encapsulation is another key element of object orientation. Encapsulation (or information hiding) allows program changes to be reliably made with limited effort [12]. In practice, this provides an important solution to one of the major problems associated with expert systems technology: maintainability [5]. Object-orientation improves maintainability by providing explicit barriers between objects such that modifications to each may be performed independently.

Providing a unified system integration paradigm: A third key element of object orientation is polymorphism. Polymorphism is a concept in type theory in which a name (such as the characteristic "programmer" may denote objects of many different classes that are related by some common superclass. Thus, any object denoted by this name is able to respond to some common set of operations in different ways. Without polymorphism, the developer

ends up writing code consisting of large case or switch statements. The combination of polymorphism and encapsulation offers the advantage of modularity, which eases the task of the system developer who aims to develop a seamless integration between heterogeneous environments.

Assumption/Limitations of the proposed model

To implement the proposed model, the following assumptions have been made: a) cognitive psychologist will continue to work with computer scientists to develop an accurate measurement of user capabilities; Currently, there is no consensus on what characteristics measure a user's capabilities. Both object and subjective measures are needed as part of an indexing systems; b) object oriented software will continue to mature as the field matures and gains acceptance. Although object orientation is not a new concept, it has taken several years to gain acceptance in a mostly structured oriented world. Only recently have high powered object oriented development environments been available; c) the power to provide useful user modeling systems will continue to be dependent on the power and speed of computers. Neural networks require speed and power. Parallel processing is not yet commonplace.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

This paper has examined the pros and cons of both expert system and neural network for modeling user in an object-oriented environment. The framework of synergy of expert system and neural network is introduced. Detailed functionalities of components and their associations are discussed. Based on the proposed research framework, we have been developed the prototype which is nearing completion.

REFERENCES

1. Allen, R. B., "User models: Theory, method, and practice," *International Journal of Man-Machine Studies* 32, pp. 511-543 (1990).
2. Brajnik, G., Guida, G., and Tasso, C., "User modeling in expert man-machine interfaces: A case study in intelligent information retrieval," *IEEE Transactions on Systems, Man, and*

- Cybernetics* 20(1), pp. 166-185 (1990).
3. Caudil, M., "Using neural nets: Fuzzy cognitive maps," *AI Expert*, pp. 49-53 (1990).
 4. Chignell, M. H. and Peterson, J. G., "Strategic issues in knowledge engineering," *Human Factors* 30(4), pp. 381-394 (1988).
 5. Chorafas, D., *Knowledge Engineering*, Van Nostrand Reinhold, New York (1990).
 6. DARPA., *Neural Network Study*, AFCEA International Press, Fairfax, VA (1988).
 7. Eberts, R. E. and Eberts, C. G., "Four approaches to human-computer interaction," in *Intelligent Interfaces: Theory, Research and Design*, ed. P. A. Hancock and M. H. Chignell, Elsevier Science Publishers, New York (1989).
 8. Finin, T. W., "GUMS - A general user modeling shell," in *User Models in Dialogue Systems*, ed. A. Kobsa and W. Wahlster, Springer-Verlag, Berlin (1989).
 9. Fox, K. L., Henning, R. R., Reed, J. H., and Simonian, R. P., "A neural network approach towards intrusion detection," *Proceedings of the 13th National Computer Security Conference*, pp. 125-134 (1990).
 10. Frankel, R., "A hybrid knowledge-based approach to structured selection," *Journal of Systems Software* 13, pp. 13-23 (1990).
 11. Gallant, S. I., "Connectionist expert systems," *Communications of the ACM* 31(2), pp. 152-169 (1988).
 12. Gannon, J., Hamlet, R., and Mills, H., "Theory of mudlers," *IEEE Transactions on Software Engineering* SE-13(7) (1987).
 13. Gaschnig, J., Klar, P., Pople, E., Shortliffe, E., and Terry, A., "Evaluation of expert systems: Issues and case studies," in *Building Expert Systems*, ed. F. Hayes-Roth, D. A. Waterman, and D. B. Lenat, Addison-Wesley Publishing Co., New York (1983).
 14. Gutierrez, O., "Experimental techniques for information requirement analysis," *Information and Management* 31, pp. 31-43 (1989).
 15. Kaiser, G. E., "AI Techniques in software engineering," in *Knowledge Engineering: Applications*, ed. A. Hojjat, McGraw-Hill Publishing Co., Inc., New York (1990).
 16. Kandel, A., *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley Publishing Co., Reading, MA (1986).
 17. Kobsa, A., "User modeling in dialog systems: Potentials and hazards," *AI and Society* 4, pp. 214-240 (1990).
 18. Kobsa, A. and Wahlster, W., "User models in dialog systems," in *User Models in Dialogue Systems*, ed. A. Kobsa and W. Wahlster, Springer-Verlag, Berlin (1989).
 19. Minasi, M., "Exhibiting uncertainty," *AI Expert*, pp. 13-15 (1990).
 20. Norcio, A. F., "The software engineering of adaptive human-computer interfaces," *Proceedings of the 1989 IEEE International Conference on Systems, Man, and Cybernetics* (1989).
 21. Nunamaker, J. F., Konsynski, B. R., Chen, M., Vinze, A. S., Chen, Y. I., and Heltne, M. M., "Knowledge-based systems support for information centers," *Proceedings of the Twenty-First Annual Hawaii International Conference on Systems Sciences*, pp. 96-105 (1988).
 22. Rich, E., "Users are individuals: Individualizing user models," *International Journal of Man-Machine Studies* 18(3), pp. 199-214 (1983).
 23. Shaw, M., "Abstraction techniques in modern programming languages," *IEEE Software* 1(4), pp. 10-26 (1984).
 24. Smith, R. and Vokurka, John F., "A software selection model for the special student," *The Computing Teacher*, pp. 36-38 (1991).
 25. Staggers, N. and Norcio, A. F., *Mental models: Concepts for human-computer interface research*, submitted for publication.
 26. Vessey, I. and Galletta, D., "Cognitive fit: An empirical study of information acquisition," *Information System Research* 2(1), pp. 63-84 (1991).
 27. Zadeh, L. A., "Fuzzy sets," *Information and Control* 8, pp. 338-353 (1965).