Artificial intelligence in office information systems

by PETER COOK and CLARENCE A. ELLIS
MCC
Austin, Texas
BIPIN C. DESAI
Concordia University
Montreal, Quebec, Canada
CLAUDE FRASSON
Université de Montréal
Montreal, Quebec, Canada
JOHN MYLOPOULOS
University of Toronto
Toronto, Ontario, Canada
NAJAH NAFFAH
Bull-Transac
Massy, France

INTRODUCTION

The recent interest in the application of artificial intelligence (AI) to office problems through the vehicle of automated office systems is due both to the “pull” of those interested in office problems and the “push” of the AI community. The pull from within the body of manufacturers and users of office systems comes from the lessons learned during the last decade while observing the rise of management information systems and their problems. The push from the AI community is due to a widely held view within that group that many of their systems have matured to an extent that these can earn, and are earning, their keep in the commercial sector.

Artificial intelligence, which has been confined largely to research and development laboratories, is finally moving to the office. One sees increasing evidence of AI technology that is being merged with existing products; this marriage of AI with the traditional product renders the union more powerful and easier to use. The goal of AI in the early days was to recreate the working of the human mind in a machine (and hence the oxymoronic term): this goal has evolved over the years into a more attainable one, namely, that of making computer systems easier to use by humans whatever their training and understanding.

The current efforts of the AI community, as related to applications in OIS, is to merge AI smoothly into existing software and systems, making them easier to use. Thus expert systems, the most prodigious product of the AI research, are mated with existing systems like the Automatic Teller Machine to make the latter more expert in allowing a withdrawal or an advance without the intervention of a human banker. Speech recognition and natural language interfaces allow additional ease in interacting with existing software systems.

USER INTERFACE

The evolution of user interfaces is based on several converging forces and components which have been the subject of recent research.

The first step in this evolution involved relational query languages with non-procedural approaches such as SQL or example-based systems such as QBE. The main problems with this type of language are that the users must be aware of the database structure such as the relations and their attributes. The users, particularly the untrained end-users, have difficulties in building the request using the database manipulation language. Furthermore, they generally have imprecise ideas about what they are searching for.

A natural language interface allows the user of a database to input a query in a natural language such as English or French rather than in the formal query language; the goal being to permit the user to express his/her information needs in his/her own language, and in conceptual terms particular to his/her understanding of the database application domain. The user is also freed from knowing about database management systems, data models, or database schemes. Allowing the user to access a database using natural language shifts onto the computer system the burden of mediating between the two views of data: the way in which the data is stored and the way in which an end-user thinks about it. A DBMS, particularly a...
relational one, accomplishes part of this task. What the user interface must do is reconcile the user's view with the DBMS's view.

In order to achieve this data independence, the interface must incorporate a considerable amount of knowledge. This includes knowledge about natural languages, the domain database application, and DBMS's and their query languages.

THE USER PROFILE

All the above approaches are means to facilitate the accessing of data, however, they do not identify the user. There is presently an increasing need for improving the user interface, but the effort should be directed toward considering the type of user for which the system is intended.

The present systems do not distinguish between users, who, in fact, are different according to their knowledge and experience in the use of the system, their interests in the database, the kind and the form of the reponse that they require from the database. These aspects should be considered in order to fit the users' needs. The introduction of artificial intelligence components able to recognize a wide variety of users, should greatly enhance the interaction in database systems. The benefits are characterized by an individualized dialogue, personally tailored help, simplification of the interactive process, and responses which are given according to the views and interests of the user.

To achieve these objectives, the user profile should constitute an important part of an intelligent interface. A user profile can be composed of various types of information such as: (1) information on customs, level of experience and qualifications of the user in his/her environment, (2) familiarity of the user with the system, (3) information objectives and interests of the user in the database, and (4) a recording of the transactions (history).

The important fact is that this knowledge must be dynamic. It is updated after each transaction in order to obtain a precise image of the user, who will then be identified automatically. With a classical system, the user is more or less able to query a database and extract pertinent data from the flow of the elements produced as output. An intelligent interface should be able to recognize the user and dynamically adapt the dialogue. The desired results should then be deduced by the interface.

Applications of the user profile are numerous. The most important concern information retrieval systems and intelligent tutoring systems. In this last domain, other aspects such as the psychological reactions and performance factor of the user must be included in the profile.

KNOWLEDGE REPRESENTATION

Artificial Intelligence can be of value in the development of the "office of the future" in at least two ways. First, by adding to the functionality of office information systems with natural language front-ends, problem solving and planning submodules, expert system front-ends that add some subjective "judgment" to the system's capabilities. Fikes, Barber, and Woo provide good examples of this type of application of AI. Second, and perhaps more important, AI can be used to facilitate the development of office information systems. One way to do this is to build "automatic programming" environments. Another way is to adopt knowledge representation ideas in the development of new classes of languages for requirements analysis and design. It is this last area of AI influence on Office Automation that will be the focus of the rest of the section.

To construct a requirements or design specification for an office information system demands the representation and integration of disparate knowledge into a coherent knowledge base. Thus, a requirements specification for an office information system needs to capture the knowledge that exists within the office, and which prescribes the patterns of behavior of the system to be built and its environment. For instance, to build a student information system one needs to describe students, their associated attributes—such as address, field of study, courses and supervisor—the activities they participate in (such as registering in courses) if certain rules are satisfied (such as pre-requisites for the courses have been completed successfully)—and the like. One needs also to describe the activities to be carried out by the intended system, the information it will handle, and how that information is obtained from the environment. In addition, for an office information system to be useful, there must exist a framework for the interpretation of its contents with respect to the intended application. Such interpretation is only possible if the system and/or its users "know" how accurate, complete, and precise the information handled by the system is. For the student registration system, for example, it must be known how often student records are updated in order to determine how the contents of the system "match" reality for a query such as "Who is taking course csc324?". If updating of the system is instantaneous, its contents fully reflect the current state of the environment. If, on the other hand, it is updated once a week, one can only answer the above question with something like "As of date X, the following students: ...".

It follows from these observations that knowledge bases built during the early phases of office information system construction will need to have a number of features. First, they must provide an account of the structure, static, and dynamic, of the environment within which the system must function. Second, they must provide an account of how the contents of the system to be built relate to the environment. Additionally, one can expect that such knowledge bases will, in general, be large, involving thousands of concepts, which must be described and organized in a way that renders the knowledge base comprehensible. Moreover, the knowledge bases will be dynamic in the sense that the rules and procedures that determine the behavior of the environment will change frequently.

What kind of a knowledge representation framework can accommodate these requirements? First, the framework must provide support for the representation of time to facilitate the modeling of the dynamics of the intended system and its environment. Second, the framework must draw a distinction between the contents of the information system and the state of the environment, so that one can make statements about the accuracy and completeness of the information handled by the intended office information system. Third, the framework
must offer structuring mechanisms since, as was said before, this is a knowledge engineering in-the-large activity. The requirements modeling language RML, and its successor CML exemplify the kinds of linguistic tools one can develop taking these considerations into account. Both languages treat a requirements specification as a history of the world being modeled. CML, however, goes further in treating a requirements specification as a history of our knowledge of the world. This feature makes CML powerful enough to talk about the completeness, accuracy, and precision of the information handled by the intended system.

Applying knowledge representation techniques to the development of design specification languages is slightly more problematic. One might argue (and many have!) that design specifications should be independent of the environment within which the system under development will eventually function, and should simply provide an account of the system's behavior. Consider, however, a system which maintains information about, say, students at a University. A formal specification of such a system which merely describes its behavior but doesn't try to provide an account of what the information handled by the system means in the first place (i.e., how it relates to reality), seems incomplete. It tells us how symbols will eventually be pushed around inside a machine. It doesn't give us any guidelines on how to interpret these symbols. This observation suggests that, at least for information systems, a design specification should come with an account of how mechanism behavior corresponds to the world being modeled. For design specifications of this sort we need linguistic tools that on the one hand, allow for the description of system components, their state, and I/O behavior; and on the other, come with a rich semantic theory that allows one to relate system states and functions to the world being modeled. The so-called semantic data models (attempt to) do just that. (See Brodie and Borgida.) An example of a semantic office model is described in Gibbs.

We have discussed how and why knowledge representation techniques can influence the features of requirements and design languages for office information systems. Implicit in our arguments is the assumption that building such systems requires several linguistic levels: some for requirements modeling, others for design, and still others for implementation. What implications does such an influence have for the environments offered for building such systems? To start with, each linguistic level used needs an environment. Those for the more procedural levels can offer typical facilities such as special purposes editors, interpreters, tracing and debugging packages, version control, and the like. The environments for the less procedural levels need reasoning facilities so that a user can probe a knowledge base to see if it is consistent with his/her expectations. A second type of facility needed for such an environment is intended to make it possible to generate lower level (and more procedural) specifications from higher level (and more declarative) ones. Depending on the nature of the two levels, it may be possible to have a compiler that handles this job. Alternatively, the environment may provide facilities for the interactive generation of the lower level specification from the higher level one. These facilities could include expert system features so that the environment plays the role of an active assistant rather than a passive bookkeeper.

in the generation of a specification. A third desirable facility involves the maintenance and management of multiple specifications for a particular software system corresponding to the different linguistic levels supported by the environment. Such a facility would allow a user to maintain a requirements specification, a design specification, and an implementation specification of his/her software, along with information on how parts of one specification relate to parts of others. With such a setup, it is possible to determine how changes of the specification at one level affect the specifications at other levels. The research project outlined in Jarke, Mylopoulos, Schmidt, and Vassiliou focuses on an environment intended to provide all three facilities mentioned above. It is fair to add that there are scores of research issues to be addressed in realizing an environment of the type advocated here and that a research program addressing such issues can only be described as long term.

OTHER OIS AIDS

Meetings constitute an important part of the communication and coordination work of organizations. They are used to disseminate information, explore ideas, resolve disagreements, and enhance teamwork to achieve organizational goals. The large amount of manager time consumed by meetings has been documented and reported in the literature.

The task of organizing and executing an effective meeting can, however, be both time consuming and difficult. After this time consuming preparation, there are no guarantees that the meeting will proceed smoothly, and evaluation of the "goodness" of a meeting is quite elusive. Unsuccessful or wasteful meetings are experienced frequently, but the exact causes of these failures are largely undocumented and not well understood. One of the reasons for this lack of understanding is that the results and effects of a meeting can be quite diverse, ranging from meeting minutes and action items to feelings of wasted time and latent disdain for certain people and tasks: these can be very difficult to capture and quantify. Another reason is related to the sometimes surprising and unpredictable nature of participant behavior. A further reason is the lack of formal models of meetings, and lack of a "theory of meetings" within which researchers can work and relevant results can be interpreted. The thrust of Project Nick at MCC is to perform interdisciplinary research into the analysis of content, structure, and protocols of meetings. Our group has been performing research in the areas of meeting analysis and meeting augmentation. In this ongoing effort, we are theorizing on a large set of ideas; assembling and implementing a subset of these ideas; and performing experiments to validate these ideas which appear most promising and applicable. We believe that new technology may present the opportunity for new and better meeting styles and organizations.

The meetings research is based on a two pronged approach with both aspects proceeding in parallel and complementing each other. The two prongs are building theories and building systems.

Building Theories—This aspect of our work began with a careful definition of our basic notions (meeting, exploration, design) and proceeds by creating intuitions, wild ideas, asser-
tions, and hypotheses. The hypotheses, in turn, suggest certain systems which we can build to confirm these hypotheses. This work has tried to pinpoint aspects of meetings which frequently go wrong, and to develop measures of goodness of meetings; all of this interdisciplinary work is clearly dependent upon factors such as the type and intent of the meeting.

Building Systems—This includes the construction, test, and usage of electronic meeting aids. Our meeting aid equipment includes an electronic blackboard and interconnected personal computers for all meeting participants.

In the future, we envision analyses of information derived during the meeting, and used to enhance the meeting in real time.

HARDWARE FOR OIS USING AI

Current work in office information systems has demonstrated the applicability of various techniques of artificial intelligence to the office environment. The main hardware components of an office information system must be developed specifically to support artificial intelligence programming in order to achieve maximum utilization. This is particularly true for a workstation, the site of interaction with a user which demands the highest degree of intelligence. Three characteristics of the workstation can be identified; emphasis on AI techniques, application to the office environment, and inclusion of multimedia and natural language processing (NLP) capabilities. Workstations for future office information systems must combine these features if they are to achieve any degree of success.

In addition to the fairly standard hardware, the workstation requires a number of less common components that are required to fulfill the software objectives of the projects. These components include:

1. A telephone interface.
2. Audio hardware. Specialized devices are to be used for the recognition and generation of human speech and possibly for sampling and compressing audio signals. This hardware is necessary for the multimedia user interface.
3. A data filtering device. To perform pattern-oriented retrieval from the disk at high speed, a special processor known as the Schuss filter will be integrated with the workstation hardware. The use of a hardware filter on the intelligent workstation is one of the most novel aspects of the system.

The main requirements for the operating system used by the intelligent workstation are support for: (1) distributed processing, (2) Real-time processing, and (3) Knowledge-based applications.

Since office applications are inherently distributed, the operating system must provide facilities for networking and interprocess communication. The data handled in the office include audio and image data in addition to the traditional numeric and character data types. To perform such functions as acquisition and storage of audio data, the operating system must have real-time processing capabilities. Finally, knowledge-based applications can benefit from operating system support. For example, many intelligent applications for the office will make use of large, shared knowledge bases in which case adding and retrieving knowledge becomes crucial. It is expected that the Schuss filter can be used to improve the efficiency of knowledge retrieval.

CONCLUSION

We have examined the various areas of application of AI technology in office information systems and the office of the future. The use of more convenient user interfaces in the form of natural languages and the convenience of individualized interface require, the maintenance of a wide variety of knowledge by the interface system. In addition, the system must be able to interpret the contents of the information system. Such interpretation requires that the system know the dynamic environment within which the system operates, as well as the accuracy of the information contained in the system.

Support for other office functions like meetings, which constitute an important part of the communication and coordination work of an organization, is also essential. The use of modern technology, including AI, to improve meetings is thus vital.

The need of an intelligent workstation which supports AI programming is obvious. Such a workstation must have the capability to support AI programming, multimedia and natural language processing.

REFERENCES