Intelligent approaches to performance support

Philip Barker, Stephen Richards and Ashok Banerji

Interactive Systems Research Group, School of Computing and Mathematics, University of Teesside, UK

Abstract

Electronic performance support systems provide an important method of meeting on-demand educational and training requirements. They also provide efficient and effective ways of enabling the knowledge and expertise within an organization to be shared. This paper discusses the design of a distributed electronic performance support system and the ways in which 'intelligent agents' based on expert systems and neural networks can be used to locate and share distributed expertise. A case study illustrating our approach to the implementation and use of intelligent agents is presented.

Introduction

The performance of virtually all organizations depends on a wide range of internal and external factors that can interact with each other in many complex ways. One very important aspect of performance is competitivity - the ability to perform organizational tasks in ways that are more efficient and more effective than those of other organizations. Research has shown that the competitivity of any given organization in the manufacturing and services sector largely depends on four critical factors (Churchman and Cerhulst 1963; Beishon and Peters 1972): (1) a flexible management structure that is able to take advantage of the latest decision-making tools; (2) the availability of appropriate facilities to enable the organization to perceive and understand changes and trends in market forces; (3) an adaptable and responsive workforce; and (4) the ability to make information and knowledge available at critical points of need. Such aspects of organizational activity are, obviously, also broadly applicable to educational institutions, not least since each aspect involves people and human behaviour. In order to be successful and competitive, it is therefore imperative that an organization provides appropriate opportunities and facilities to enable its workforce to perform jobs to maximum effect in an ergonomically pleasing way. One approach to realizing this objective is through the use of various types of performance support system.

A performance support system is essentially a 'job aid' that enables its users to improve the efficiency and effectiveness with which they are able to complete particular tasks. A simple example of a performance support system is a pocket calculator. Word-processing systems and desktop-publishing packages are further examples: they significantly improve the efficiency with which their users can create documentation. Email and computer-supported co-operative work (CSCW) techniques are also important performance aids since they enable geographically distributed organizations to undertake rapid inter-site and interpersonal communication, to share intellectual and managerial resources, and to participate in collaborative problem-solving at a distance.

High-performance, general-purpose, multimedia, interactive, computer-based workstations now provide a powerful environment within which to implement a new type of sophisticated electronic performance support system (EPSS). Such systems can be designed to perform a number of generic job-aiding functions such as the provision of help and advice, the sharing of organizational expertise, and access to large volumes of information and documentation. Also, in situations where it is appropriate, an EPSS can provide on-demand instruction at a particular point of need. This latter function can be achieved by means of a just-in-time (JIT) paradigm which can be realized through interactive on-the-job training techniques. JIT training methods involve providing instruction to an individual as and when it is needed in order to perform a particular job function effectively. The concept and potential of EPSSs is well-documented in the literature (see, for example, Gery 1991; Barker and Banerji 1993; and Whitlock *et al* 1993).

One important aspect of EPSS design and implementation is the provision of organizational documentation in electronic form. This requirement is necessary for a number of practical reasons, but primarily because of the large volumes of information that are involved and the

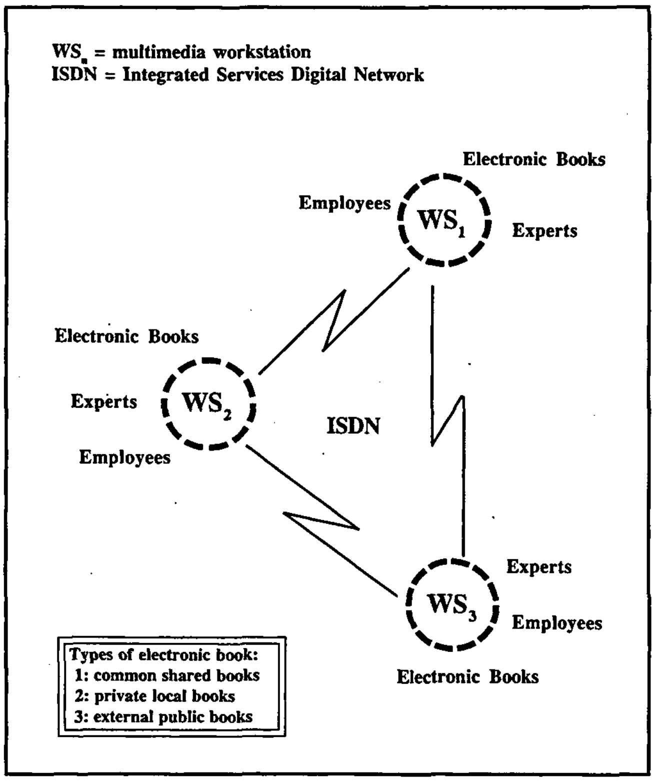

Figure 1: A distributed electronic performance support system

ease with which electronic documentation can be accessed, updated and shared. As part of our EPSS research we have therefore been exploring the problems involved in converting conventional paper-based documentation into electronic form.

As an alternative publication medium we have been using CD-ROM to produce a series of electronic books for use with performance support systems (Barker 1991; Barker 1992). A range of different types of publication have been produced, containing textual information, multimedia information (text, pictures and sound) and hypermedia material. These electronic publications embed the information necessary to facilitate many different aspects of organizational activity, and a range of storage and retrieval techniques has been used in order to provide access to the information held on the CD-ROMs.

Our previous research into generic EPSS structures has been described in detail elsewhere (Barker 1991; Banerji 1993). This paper now describes the ways in which we have been using electronic books to provide intelligent approaches to performance support within a distributed EPSS facility. The basic type of system we have been considering is illustrated schematically in Figure 1.

The implementation depends on the creation of three basic types of electronic-book publication: common shared books, private local books, and external public books. Within our system, 'intelligent agents' are used to keep track of an organization's expertise within different problem-solving domains. These agents can be used to locate sources of relevant expertise, and to facilitate the transfer of this expertise (in electronic form) to the required point of need. This transfer can be accomplished using a dedicated network or, for example, the Integrated Services Digital Network (ISDN) when large amounts of data have to be transferred rapidly over public telephone lines (for a technical overview of ISDN, its limitations and practical potential within education and training, see Jacobs 1994).

Development tools for intelligent agents

A range of different artificial-intelligence techniques can be used to develop intelligent agents for use within distributed EPSSs. Two of the most important types of generic tool we have been using in our work are expert systems and neural networks.

An expert system is a software tool containing domain expertise which can be used to solve problems within a particular subject area. Most of the commonly available expert systems model domain knowledge in terms of collections of 'if-then-else' rules (Hart 1986). Essentially, these rules constitute a decision tree which, given the necessary input parameters, reduces a large problem-space to more manageable proportions. The rules then provide relevant sources of advice about the problem being solved. The major limitation of expert systems, however, is that they cannot learn: all the rules used have to be coded into an expert system by human experts before it can be used to solve problems.

A neural network system is a computing facility modelled on the basic way in which the human brain is thought to work (Beale and Jackson 1990). Such systems consist of a large collection of artificial neural elements interconnected in various ways in order to solve particular classes of problem. The interconnections within a given neural network are formed by a process of training. During its training phase, a neural network learns what connections are necessary in order to solve a given type of problem. Once the training phase is complete, a trained neural network can be put to work within the problem domain for which it has been created.

In many situations, tools can be constructed which depend on the useful features of both expert system and neural-network technologies. This is the approach we have adopted in order to create the intelligent agents used in the case study described below. In the example presented, the expert system is used to embed rule-based 'common-sense' knowledge, while a set of neural networks is used to encode 'deep' compiled knowledge. Each neural net represents a different conceptual view of a knowledge domain within a particular distributed EPSS facility.

Obviously, within the distributed EPSSs we are developing, the expert system and the neural-network components have to be integrated in a transparent and seamless manner. They also have to be combined with those other components of the EPSS that are needed for the particular application being considered (email, text retrieval, simulation, computer-based instruction, and so on). We will now describe in more detail the way in which these objectives are realized.

Case study - views of knowledge

The work we have been undertaking is aimed at finding ways of sharing expertise and knowledge within a distributed EPSS facility similar to the one illustrated in Figure 1. The particular aspect we have been exploring relates to the sharing of different views of a common knowledge domain within a distributed environment. The basic structure of the system we have been developing is illustrated schematically in Figure 2. In this particular work, the knowledge pool consists of large collections of text-based material that have been published in the form of electronic books on CD-ROM.

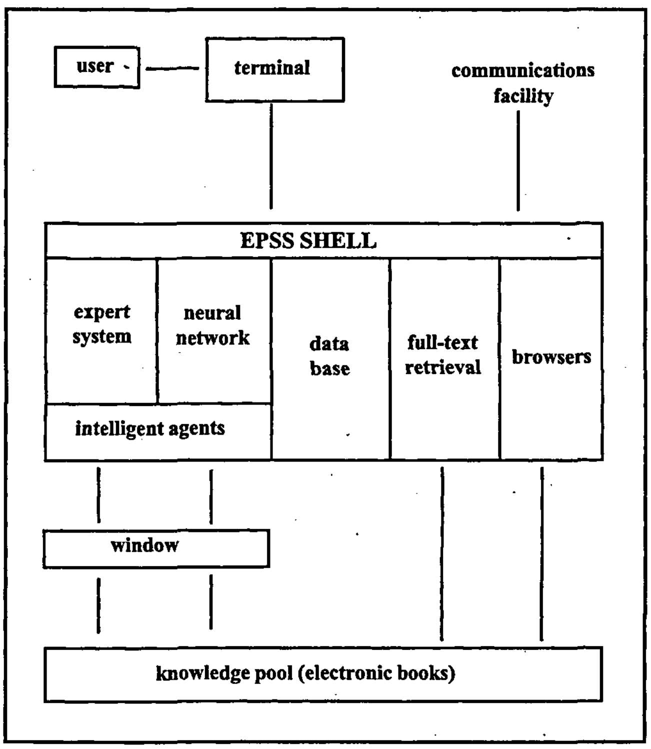

Figure 2: Use of intelligent agents in an EPSS facility

Fundamental to the basic EPSS shell is its system database, used to organize and control access to all the underlying information. A user has full access to all the available material through the mechanisms associated with conventional computer-based full-text retrieval (Heaps 1978), book indexes and book contents lists. Of course, if necessary a user could also simply browse through the material in the knowledge pool using a range of different browsing techniques, but because of the significant volumes of information involved this approach would be time-consuming and inefficient.

The full-text retrieval method is difficult to direct with regard to context. This can be both an advantage and a disadvantage. It is an advantage in the sense that all information associated with a given set of keywords is retrieved. On the other hand, it is a disadvantage in that high degrees of relevance are often difficult to achieve; that is, the problem with full-text retrieval is that it can retrieve a substantial volume of information which is irrelevant, even when Boolean search techniques (AND, OR etc.) are used.

Within the system we have been developing, users are able to initiate both 'direct' or 'assisted' information access. Direct access involves invoking either the full-text search engine, the book browser, or a direct query to a specified neural net (that is targeted towards a particular part of the knowledge pool). Such direct access depends on a user having a clear idea of both problem formulation and knowledge-pool organization. While this may be feasible with small amounts of knowledge, it can become a problem when access to information from a large corpus is required. An alternative strategy involves a user seeking guidance from an expert system for direction as to the most suitable access method based on his/her current orientation to a particular problem. In this situation, the expert system uses basic rules about the knowledge pool and the user's problem definition in order to identify the strategy most likely to yield the desired information.

If a user chooses an approach based on a neural network, the expert system will access the system database in order to obtain the necessary information (keywords and their associated weightings) to supply to the neural-network engine. Because people differ in their views of knowledge, different views of the knowledge pool can be constructed by changing the weightings assigned to the keywords. This information is used to create a new window (or 'view') onto the knowledge pool through the production of a new net (see Figure 2). The resulting net, with particular values of keywords and strengths, constitutes the compiled version of various experts' knowledge. These nets provide multiple views of the information - for instance, from the perspective of a particular type of expertise, context, interest or problem area.

We are currently using two strategies to create the information needed to train our neural networks. One of these is automated, while the other (manual method) depends on the encoding of knowledge by an expert. The automated approach uses software tools, which we have developed, in order to identify potential keywords and calculate their frequency within the various sections of a document. These frequencies are then used to calculate weightings and produce a training set for the neural-network engine. A typical training set could be as follows (where kw = keyword and w = weighting):

Input: kw1,w3; kw2,w2; kw4,w5; ..... Output: a

Input: kw1,w5; kw3,w4; kw5,w1; ..... Output: b

The expert encoding technique relies on an expert to input relevant keywords and/or phrases, along with respective weightings, into our training-set generation tool. The latter technique requires greater production time and effort but provides increased accuracy. When using the delivery system, a net appropriate to the current task is selected - either manually by the user or automatically by the expert system. Information about the current problem, consisting of keywords and relative weightings from the user, is then encoded into a query. This query is passed, along with the appropriate neural-net information from the database, to the neural network engine. Information is returned in the form of a list of locations in order of relevance to the current problem. Direct access is then available to the locations suggested. Only on the occasion when such a strategy fails will the user need laboriously to access every instance of a particular set of keywords using the full-text retrieval engine.

A major advantage of a neural-net approach is the potential for the various networks to continue learning about the knowledge pool from interaction with expert users. Keywords can be added or removed, weightings can be adjusted, new knowledge can be added to the pool, and new training sets can be produced which allow the first-time hit rate of the networks to be improved. In this way, an information environment is created which can learn and adapt to constantly changing environments. Obviously, such a strategy is also very useful when organizational expertise is distributed. For example, in on-the-job situations, information requirements are often immediate. If a particular individual does not have the necessary expertise, then very often he/she will consult the organization's expert in that particular field. In a situation where the expertise is at a remote site, or simply unavailable, efficiency is obviously diminished. Even in cases where the remote expert is available, it can take some time for that expert to locate the particular information required. The distributed system we are developing goes some way to reducing such problems. When a query is made to an expert, the expert has access to a number of neural nets providing windows on the knowledge pool which are tuned to particular perspectives or sets of problems. This allows the expert quickly to obtain the required information and respond promptly to the remote request. In addition, the expert's neural nets can be sent to the remote sites whenever any new learning has taken place. This means that in the absence of an available expert, any individual can formulate a query based on a combination of the expert system and the relevant expert's neural nets, providing the potential for non-experts to solve problems independently outside their knowledge domain. Such immediate access to essential information can greatly improve both individual and organizational efficiency.

The prototype system which has been built to implement the model described above and depicted schematically in Figure 2 runs under Windows 3.1 on a 486 PC. The EPSS shell is written in KnowledgePro for Windows (version 2.21). KnowledgePro also provides the environment for the creation of the expert-system component of our system (Barker 1989). The neural networks used within our prototype have been developed using a low-cost system called NeuroShell (Ward Systems 1990). Full-text retrieval facilities are available through the STATUS information retrieval package (Harwell Computer Systems 1991). Depending on the particular tasks that have to be carried out, the EPSS shell can invoke the full-text retrieval engine or the neural network facility - or use its own inferencing mechanisms in order to take appropriate action.

Conclusion

Working together, expert systems and neural networks can offer a powerful set of tools for the implementation of intelligent agents for handling information retrieval problems within an EPSS. This paper has described some of the ways in which we have been using these techniques for sharing expertise in a distributed EPSS facility. The methods we have described have been tested using relatively small textual-knowledge corpora, and they seem to work successfully. The next phase of our work will extend these methods to large collections of multimedia knowledge.

Note

An earlier version of this paper, here significantly updated, was presented at the International Computing Congress held in Hyderabad, India, in 1993 (Barker and Banerji 1993).

References

Banerji, A.K., *Designing Electronic Performance Support Systems*, draft Ph.D. thesis, School of Computing and Mathematics, University of Teesside (in progress, 1994).

Barker, P.G. (1989), 'KnowledgePro: a review and assessment', *Engineering Applications of Artificial Intelligence*, **2**, 4, 325-38.

Barker, P.G. (1991), *Electronic Books*, special edition of *Educational and Training Technology International*, **28**, 4, 269-368.

Barker, P.G. (1992), 'Electronic books and libraries of the future', *The Electronic Library*, **10**, 3, 139-149.

Barker. P.G. and Banerji, A.K. (1993), 'Multimedia environments for EPSS development', in Balagurusamy, E. and Sushila B. (eds), *Proceedings of the Indian Computing Congress (ICC '92) - - Innovative Applications in Computing*, New Delhi, Tata McGraw-Hill Publishing Company, pp. 423-30.

Beale, R. and Jackson, T. (1990), *Neural Computing - an Introduction*, Bristol, Adam Hilger.

Beishon, J. and Peters, G. (1972), *Systems Behaviour*, London, Open University Press.

Churchman, C.W. and Cerhulst, M. (1963), *Management Sciences: Models and Techniques*, New York, Pergamon.

Gery, G.J. (1991), *Electronic Performance Support Systems - How and Why to Remake the Workplace through the Strategic Application of Technology*, Boston MA, Weingarten.

Hart, A. (1986), *Knowledge Acquisition for Expert Systems*, London, Kogan Page.

Harwell Computer Systems (1991), *STATUS User's Manual*, Oxford.

Heaps, H.S. (1978), *Information Retrieval - Computational and Theoretical Aspects*, New York, Academic Press.

Jacobs, G. (1994), 'ISDN: theoretical specification, practical potential' in Mason R. and Bacsich, P. (eds), *ISDN Applications in Education and Training*, London, Institution of Electrical Engineers.

Ward Systems (1990), *NeuroShell User's Manual*, 3rd edition, Maryland VA.

Whitlock, Q. *et al* (1993), 'Hypermedia and performance support', to appear in *Proceedings of ED-MEDIA '93*, Orlando FL.