INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19-21, 2003

AN AGENT BASED APPROACH TO FINDING EXPERTISE IN THE ENGINEERING DESIGN ENVIRONMENT

Richard Crowder, Gareth Hughes and Wendy Hall

Abstract

During the engineering design process people need to locate colleagues with knowledge to resolve a problem. As identified by discussions with practicing designers the use of computer based systems that assist users with finding such expertise will become increasingly important. In this paper we discuss the development of an agent based Expertise Finder suitable for use within an engineering design environment. A key feature of our approach is that the Expertise Finder returns both recommended contacts and supporting documentation. The Expertise Finder bases its results on information held within the organisation, for example publications, human resource records, and not on personally compiled CVs or user maintained records. The recommendations are presented to the user with due regard to the social context. The technology used in the exemplar application allows the development of distributed, interchangeable agents that use real time data to find expertise.

***Keywords**: Knowledge Based Systems, Human Resources, Work Environment.*

1. Introduction

The engineering design environment is highly distributed in nature and is characterised by a large number of information sources, which together with the designers forms a complex sociotechnical system. The paper by Wallace et al [1], discussed the future vision of the engineering design environment, and concluded that a range of knowledge management tools would be required to support their vision. This paper discusses one such tool, an expertise finder as discussed in the accompanying paper, [2].

Knowledge is considered a major asset by any organisations, though this knowledge can be buried within the corporate memory, with much of the understanding and constraints surrounding the knowledge being held tacitly by people within the organisation, [3]. It is not uncommon in some multi-site organisations to repeat work or designs already undertaken elsewhere in the organisation, then try to discover if it has been carried out at a different location, [4]. In addition, people do not always stay in the same location; they move into different task locations, disciplines or other organisations.

As an aid to the designers we are developing an Expertise Finder (EF) that is capable of locating the best expertise within a very large organisation to solve a specific problem. The EF approach does not replace the social network but attempts to speed up the connection making process. In discussions with practising designers it was clear that to be a usable tool, the EF

had to satisfy a number of criteria, in particular the EF has to be accurate and reliable, have an ability to resolve uncertainty in the problem and the underlying data must be maintainable.

An EF system will locate an expert who has the special knowledge or skill that causes that person to be regarded as an authority on a specific topic. In this paper, the term expertise assumes the embodiment of knowledge and skills within individuals. This definition distinguishes expertise from an expert. An individual may have different levels of expertise about different topics. Expertise can be topical or procedural and is arranged and valued within the organisation. The quality of the answer will depend on the explicit knowledge being used for the search, in some EF the knowledge is not peer reviewed or moderated, but supplied by the experts themselves or as the result of a consultancy exercise. Expertise systems range from centrally held database of personnel skills [5], searching a limited range of personally selected documents [6] to systems that use real-time information held within the corporate system [7]. A recommender system is an extension of this approach, where the results will be modified by feedback provided by previous users, as to the quality and validity of the recommendations [8].

In our approach we use an organisation's common set of resources, including e-mails, phone books and peer-reviewed technical report repositories. These information repositories are used to identify the required expert to contact. As they are maintained as part of an organisation's normal management function they will be more accurate than locally maintained or individually provided information resources.

In contrast to automated systems, many organisations use key personnel (managers, senior employees, information concierges, [9]) to facilitate the contacts, however in a large multinational company, it is impossible for a single person to know all the contacts. EF systems are an approach to automating this process, by augmenting and assisting the natural expertise-locating behaviour within an organisation. Figure 1 shows the relationship of an EF system to the knowledge cycle, where it forms part of the information retrieval activities.

A system that suggests people who have some expertise with a problem holds the promise to provide, in a small way, a service similar to these key personnel. Expertise recommender systems can also reduce the load on people in these roles and provide alternative recommendations when these people are unavailable.

In the recommendations provided by the EF, trust is important, we believe that this can be achieved by showing why people were not recommended or why a document was not considered so important. A document might seem relevant based on a full text search but is actually twenty years old, which be an important factor in some situations, but not in others. The provision of evidence for its decisions in the form of a list of documents and other data is considered a key EF output. Answer Garden 2 [10] has an explicit expertise-location engine and provided computer-mediated communications mechanisms to find others with a range of expertise, though the mechanisms were not very elaborate. A different approach was taken by McDonnald [9] who with a software company monitored the software developed by employees to identify their expertise in various aspects of the development process.

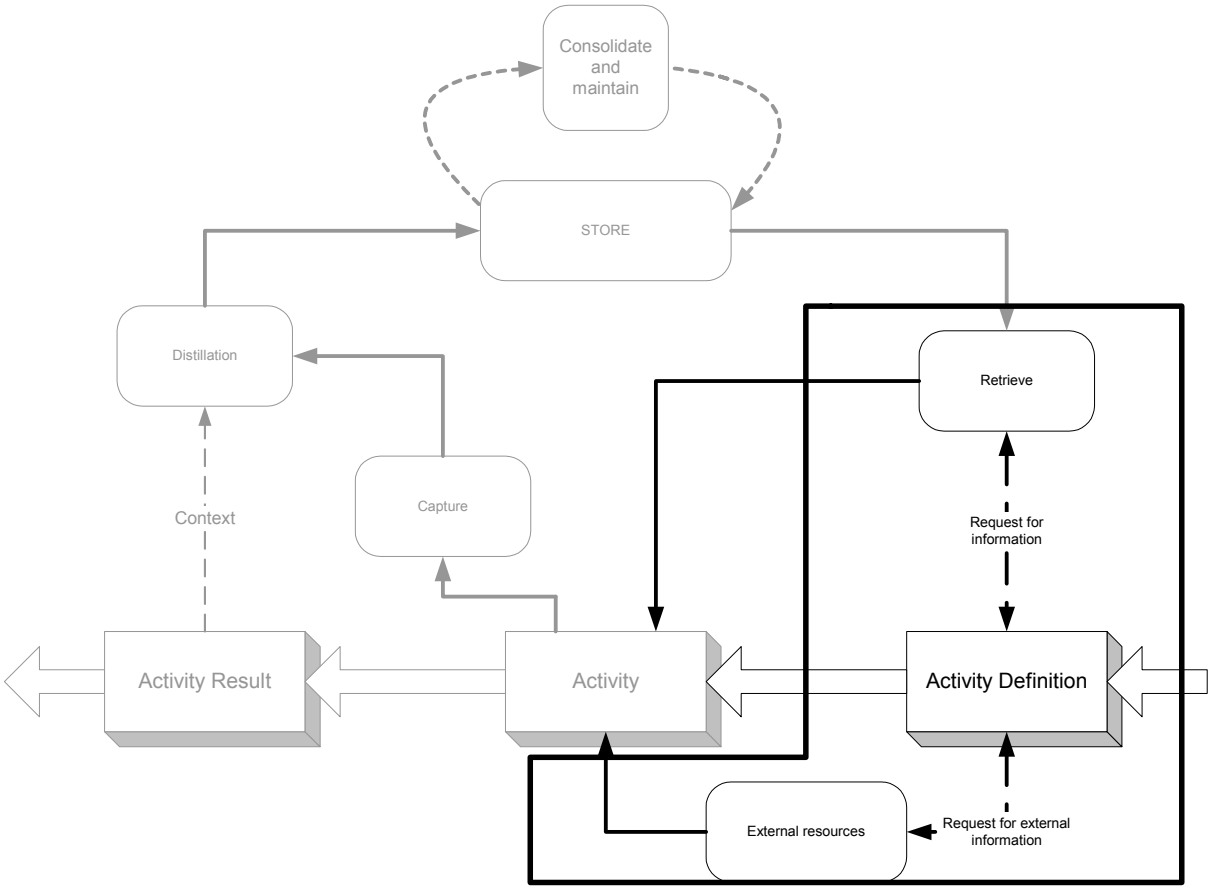

Figure 1. Location of the Expertise Finder within the conventional knowledge cycle

2. Problem definition and context

When attempting to find an answer to a problem people will tend to use the social network around them. It is natural to first ask people nearby if they know the answer or if they can recommend someone else who may know the answer. Thus a chain of connections are made utilising the experienced members of an organisation. As people are now being moved around organisations at a faster rate and organisations are becoming increasingly distributed this model starts to fail. There may be no social connection between specially separated groups even though they work on similar problems. Our system attempts to alleviate this by using the company's own resources to recommend people to contact. It does not replace the social network but attempts to speed up the connection making process.

The problem that is being addressed is summarised in Figure 2, *how does a person located in Site A, locate the best expertise to solve a specific problem?* The person's local social network will only extend to within the Site A, and therefore expertise within other sites can not be readily and quickly accessed. Across a multinational company it is a recognised principle that sites have common problems, but the required expertise is not necessarily easily accessible to each other. While the individual sites may not form a cohesive social network, they do share common sets of resources, including e-mail, phone books, publication and report repositories, in our approach to EF systems, these information repositories are used to identify the required expert. In the figure the user in Site A, uses the EF to locate an expert is Site C, as the documents used to make to recommendation are also made available, the user can be fully briefed on this person's work before an approach is made.

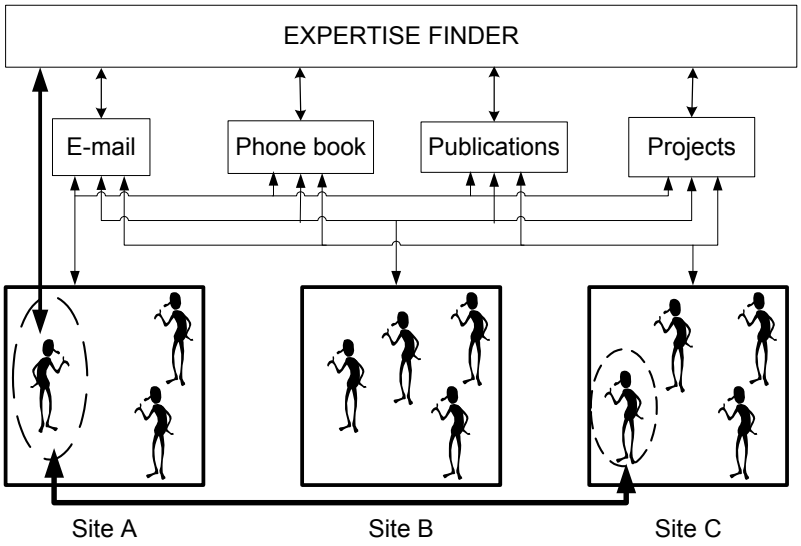

Figure 2. Expertise Finder concept

3. Expert Identification

In an engineering organisation an expert will typically be the person who has authored the largest number of reports of significance, worked on the largest number of contacts, and has extensive experience either with the current or similar organisation. In addition they will tend to hold senior posts. These people are subject experts not skills experts.

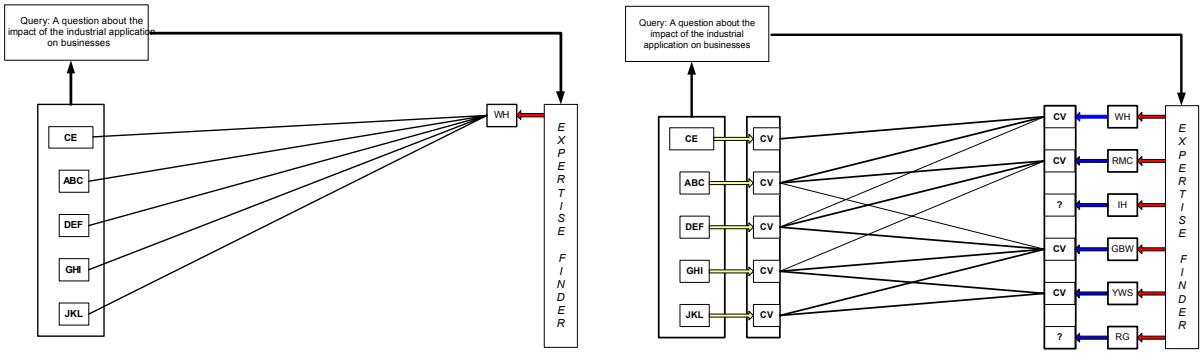

Figure 3. Information overload will occur if the social context is not taken into account (left), however if the system has knowledge of the people through their profile a better recommendation is provided (right).

However when a person wishes to contact an *expert*, there are additional social factors that need to be taken into account. Without these factors, the single expert could be swamped with queries for everyone ranging from trainees to Chief Engineer, Figure 3. The appropriate person depends on the query and the user's requirements, typically the peer-to-peer approach is considered best in the first instance; however the person requiring the expertise needs to be free to make a valued judgement as whom to approach. It is for this reason we make available all the sources for review.

As discussed by McDonald [11] the details matter in successful expertise location. The heuristics used to select the expert are bound to the organisational environment. Systems that augment expertise locating must be capable of handling large number of details that depend on the specific context and problem. This does imply that there is a considerable amount of

customisation of EF systems between organisations, making the agent approach highly suitable to the development of these systems.

4. Implementation of the Expertise Finder

The Expertise Finder system is designed to mimic the reality of an organisation in terms of its social structures and information infrastructure. The implementation of the Expertise Finder consists of a number of DIM (Distributed Information Management) Agents operating within SoFAR (Southampton Framework for Agent Research) [12]. SoFAR was developed at the University of Southampton as an agent framework designed to address the problems of distributed information management. On each occasion that the EF system is deployed the sources of data available to be used and their structures will be different. There will be commonalities due to the use of standards such as being able to access a database using standard query language or the use of protocols such as LDAP. There will still be subtle differences that require the customisation of the system. From this it is obvious that the high level steps that any system should take to identify an expert will be unique on each occasion.

In order to communicate with each other agents use a shared understanding of a domain called an ontology. Ontologies are a conceptualisation of a domain into a form which can be understood both by humans and computers. One definition of an ontology *is an explicit specification of a conceptualisation*. [13] Ontologies provide a mechanism to allow communication and interaction about a real world domain. They remove ambiguity from language through careful design. Pragmatically it allows us to concentrate on high level concepts rather than spend time on the implementation details such as communications and data representation. It therefore follows that the design of the ontology is crucial to the project and careful work is required to correctly understand and map the real world situation into the ontological vocabulary. Further technical details of how ontologies are implemented and used in the SoFAR framework can be found in [12]. The ontologies used in this work were designed previously but extended here. They represent the activities and people in our research group. A detailed explanation of their design and implementation can be found in [7].

The EF system consists of a main agent, the EF Agent, which uses a set of simpler *Source Agents* in some algorithm to determine a list of people and documents to recommend to a user. The EF agent builds an answer as XML before transforming that to HTML for delivery to the user via the Web server agent. The use of XML allows the EF Agent to be reused in other systems and its results transformed as required. Figure 4 shows the overall architecture of the system. In it we show all of the agents we have at our disposal but here we concentrate on the core interactions between those in solid lines.

The EF application is based on a previous agent application, the Dynamic CV [7]. The application used the notion of query recipes to dynamically construct a Curriculum Vitae page about a person. For instance, in the CV query, a general information page about a person, it would find and use agents to obtain telephone number, office location, and email address. The answers were combined into a Web page in which links to new queries were automatically added and thus a user could navigate around the information space. Figure 5 shows the result of a CV query.

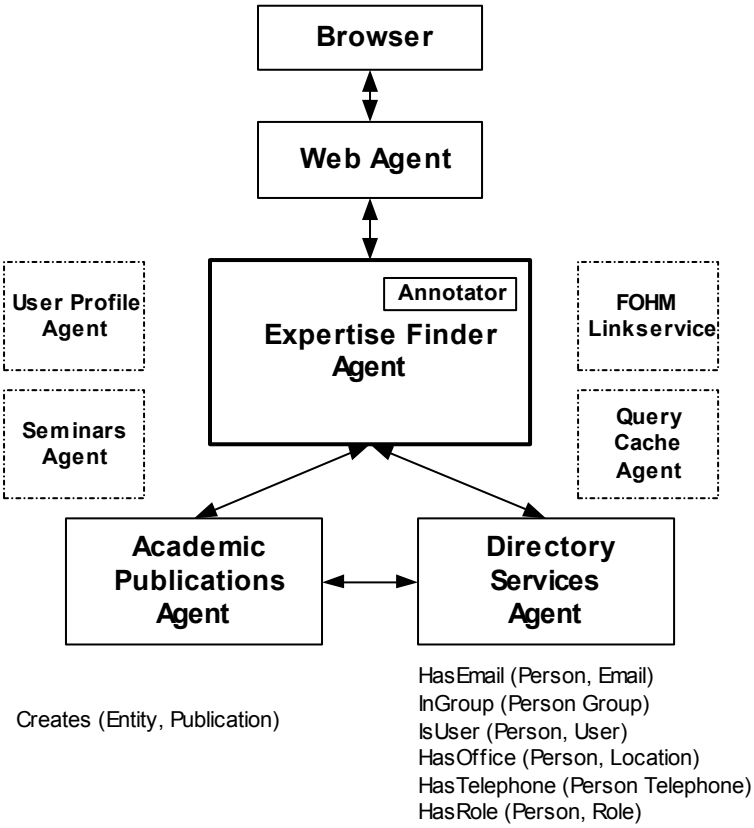

Figure 4. Architecture of the Expertise Finder. Typical predicates used are given below the respective agents.

The key weakness of the Dynamic CV application was that the main agent would gather information from Source Agents following the instructions of a query template. It would extract the data and place it onto the Web page with no understanding of the results. The EF Agent is a total redesign of this system with the express intention of not only supporting the types of query performed by Dynamic CV but also to perform complex interactions with Source Agents in order to build towards a final answer. In order to do this the Source Agents have been radically improved and the services they provide have been expanded considerably.

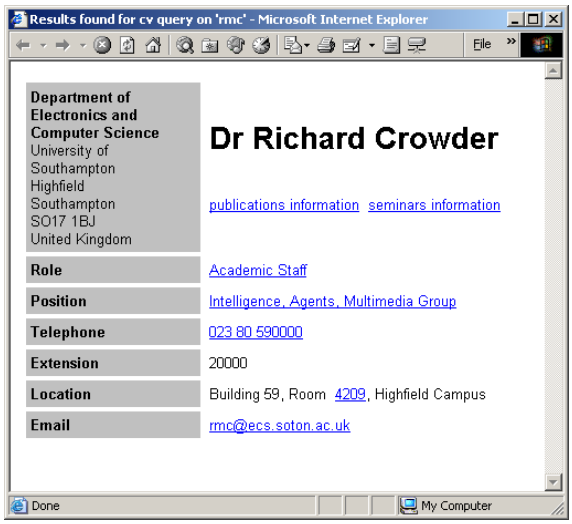

Figure 5 The Dynamic CV agent system found agents to fill in query templates

5. Prototype EF application

The current prototype application of the agent based EF is current being used to find people using the scientific publication repository within the authors' Department. As in the industrial application, the objective being to aid people to find experts on a topic amongst a group of people working for the same organisation. A user enters a query on a research subject into a Web search page. This query is given to the EF Agent by the Web Agent. The EF Agent first asks the Publications Agent to find publications using the search terms. The Publications Agent takes the query terms from the Predicate and uses them to form an SQL query. The query is run on the department publications database. The publication database lists authors by a list of full names and a corresponding parallel list of full email addresses. Hence some understanding of this and some data translation must be performed. The Publications Agent uses the Directory Services to help identify authors. It then uses the results of the query to build new Creates Predicates and return them to the EF Agent. The EF Agent will maintain a record of their details, saving duplication of queries, and begin to count the number of times the person appears in the returned publications. The EF Agent will also maintain a list of people not identified.

The final results page is made up of the returned publications, the list of found authors with a count of their occurrences and their status within the department. The list of unknown authors is also returned to allow users to decide for themselves the usefulness of such information. In the context of this application this list consists of people who have left the department or external collaborators, and are therefore less useful to the user.

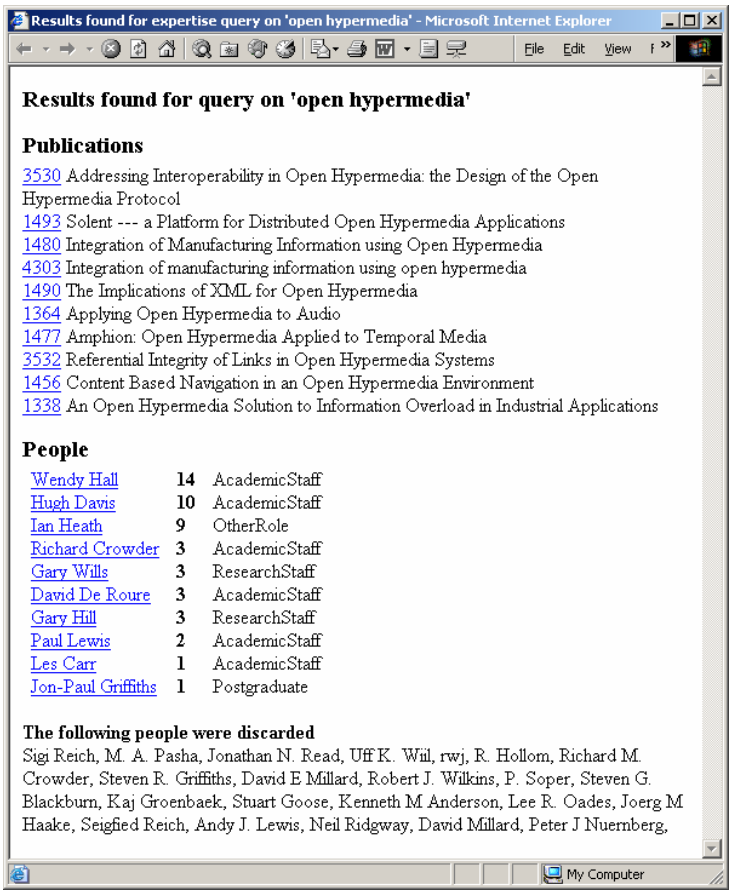

Figure 6. The results of the prototype EF, the first ten publications used to rank the expert are shown at the top.

6. Discussion

The results page, Figure 6 gives a list of papers followed by a list of people together with the number of times that person's name appears in the publications list, together with their status within the department. For brevity the results have been condensed in the illustration. The system also lists names that could not be matched by the Directory Services agent so the user understands more about the reasoning for the final answer.

Our prototype procedure to find an expert is simple but effective and the results show that it gives a fair indication of who would be a good person to contact. The system invariably finds more senior members of staff and less of the junior staff because they tend to have fewer publications or have worked on fewer contracts. It is quite likely that the contacted person would be able to quickly point the enquirer to a specialist or less senior member of staff to help with the answer. However an important social connection has been made regardless of the location of the member of staff. If the EF system provides just that one connection saving valuable time then it has been useful. The documents returned in this example are not presented with any great intelligence or ranking and the search is crude but effective. The agent approach means that a new query method or publications database can be added without disturbing the rest of the system. Currently the system does not take account of the user making the query. However we have the majority of the components required to build a user profile agent in order to add this factor into the equation.

The procedure involved in this system seems to be simple at first glance but the complexities of implementation have taught us many lessons. The agents are not as independent of each other as was planned. The complex interactions between the source agents and the large amounts of error checking they need to do are a classic example of writing systems to deal with real life data. Also if one of the underlying data sources changed significantly then it may not be able to support some part of the ontology and hence the EF Agent will need to be modified. This demonstrates that where complex queries and interactions between component systems are required there will be more brittleness in the design. This applies equally to an agent based system as it does to one based on some other black box design such as RMI or Web Services.

7. Conclusion

Significant progress has been made in the development of our Expert Finder and its associated agents. It is clear that the results from the EF are trustworthy as they are based on peer reviewed document and not an individual's own documentation. We are currently considering the use of this system within industry, using technical reports and similar controlled documentation. We are proposing to undertake the development of additional agents to refine the system including integrating other information sources for example, skills databases and captured design rationale, and we will also undertake further evaluation using exemplar information resources.

In developing an EF system one of the major challenges is the sheer scale and complexity of the organisation we have worked with, particularly with reference to the age and quality of the data available. It can be envisaged that each company site will uses and produces different types of information so no single solution will be appropriate organisation-wide, a major factor in the evolution of our design. This could result in the ontologies and agents being designed on a site by site basis.

In conclusion our work has shown that it is possible to develop EF that search on information currently available across organisations. With our approach companies do not need to undertake a knowledge audit across the organisation to populate the system. In practice the viability of our approach is dependent on the ability to access all the information available in electronic format. Even with these caveats we believe that the introduction of this approach is feasible, though a considerable number of challenges remain, in our discussions these are more social than technical.

Acknowledgements

This work was funded in part by the University Technology Partnership for Design which is a collaboration between Rolls-Royce, BAE SYSTEMS and Universities of Cambridge, Sheffield and Southampton. The work reported in this paper was funded UK Engineering and Physical Sciences Research Council under grant number GR/M83582. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the EPSRC or any other member of the UTP.

References

[1] Wallace K., Clegg C. and Keane A., "Visions for Engineering Design: A multidisciplinary perspective", Proceedings of ICED 01, Glasgow, 2001.

[2] Crowder R., Bracewell R., Hughes G., Kerr M., Knott D., Moss M., Clegg C., Hall W., Wallace K. and Waterson P. "A Future Vision for the Engineering Design Environment: A Future Sociotechnical Scenario", to be presented at ICED 03 Stockholm, 2003.

[3] O'Leary D., "How knowledge reuses informs effective system design and implementation", IEEE Intelligent Systems, Vol. 16(1), 2001, pp44-49.

[4] DeRoure D, Hall W, Reich S, Hill G, Pikrakis A, and Stairmand M. "Memoir – an open distributed framework for enhanced navigation of distributed information", Information Processing and MangementManagement, 2001, 37, pp53-74.

[5] Davenport T. and Prusak L. "Working Knowledge: How Organizations Manage What They Know." Harvard Business School Press, 1998.

[6] Maybury M, Damore R, and House D. "Awareness of organizational expertise". Technical report, MITRE, October 2000.

[7] Weal M, Hughes G, Millard D, and Moreau L. "Open hypermedia as a navigational interface to ontological information", Proceedings of Hypertext'01, 2001, pp 227-36

[8] Resnick P. and Varian H. "Recommender systems". Communications of the ACM, March 1997, Vol. 40(3), pp56-58.

[9] McDonald D and Ackerman M "Just talk to me: a field study of expertise location", Proceedings of ACM 1998 Conference on Computer Supported Cooperative Work, CSCW 98, 1998, pp 315-24.

[10] Ackerman M and McDonald D. "Answer garden 2: Merging organisational memory with collaborative help" Proceedings of ACM 1998 Conference on Computer Supported Cooperative Work, CSCW 96, 1996, pp 97-105.

[11] McDonald D and Ackerman M. "Expertise recommender: a flexible recommendation system and architecture" ACM 2000 Conference on Computer Supported Cooperative Work, 2000, pp 231-40.

[12] Moreau L, Gibbins N, DeRoure D, El-Beltagy, S, Hall W, Hughes G, Joyce D, Kim S, Michaelides, D, Millard D, Reich S, Tansley R, and Weal M. "SoFAR with DIM Agents: An Agent Framework for Distributed Information Management", Proceedings of PAAM00, 2000.

[13] Gruber T., "Toward Principles for the Design of Ontologies Used for Knowledge Sharing" Knowledge Systems Laboratory, Stanford University, 1993. Technical Report KSL-93-04.

Corresponding author:
Dr R M Crowder,
Intelligent Agent, Multimedia Group,
Department of Electronics and Computer Science,
University of Southampton,
Southampton, SO51 5SD.
Tel +44 23 8059 3441
Fax +44 23 8059 2865
E-mail: rmc@ecs.soton.ac.uk
URL: http://www.iam.ecs.soton.ac.uk