

An Agent Based Approach to Finding Expertise

Richard Crowder, Gareth Hughes, and Wendy Hall

IAM Group
Department of Electronics and Computer Science
University of Southampton
Southampton
{rmc, gvh, wh}@ecs.soton.ac.uk

Abstract. In many organisations people need to locate colleagues with knowledge and information to resolve a problem. Computer based systems that assist users with finding such expertise are increasingly important to organizations and scientific communities. In this paper we discuss the development of an agent based expertise finder (EF) suitable for use within an academic research environment. A key feature of this work is that the EF returns both recommended contacts and supporting documentation. The EF bases its results on information held within the organisation, for example publications, human resource records and not on CVs or user maintained records. The recommendations are presented to the user with due regard to the social context, and are supported by the documents used to make the recommendation. The technology used allows the development of distributed, interchangeable agents that use real time data to find expertise. It is our intention to use this approach within manufacturing and other knowledge intensive organisations.

1 Introduction

In the course of most activities, people face problems that they cannot solve alone. Their natural response is to study past experiences and re-use previously acquired knowledge, either from their own experiences or from resources within their organisation. Goa et al, [7] estimated that 90% of industrial design activity is based on variant design, while in a redesign activity 70% of the information is re-used from previous solutions [9]. For many problems, access to documentation through hypermedia or similar systems may give adequate solutions [6]. However to solve many problems people need to have specific expertise, that will allow the problem to be resolved. 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. In some cases expertise can be captured from a person and used to populate a database. This works very well when the problem is restricted to a very specific domain, for example robot maintenance, [2]. However for many problems the required expertise can only be accessed through a social network.

To solve a specific problem people want to quickly find other people with the required expertise. In many organisations, key personnel (managers, senior employees, information concierges [10]) will facilitate the contacts. Recommender systems are one approach to automate this process, by augmenting and assisting the natural expertise-locating behavior within an organisation. A recommender 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 expertise finder (EF), trust is important, 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, 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. This approach contrasts with a number of reported systems where web based information is used to provide the recommendation, [3, 4, 5]. Answer Garden 2 [1] 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 [10] who used software developed by employees to identify their expertise in various aspects of software development.

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 work reported in this paper presents details of an Expertise Finder system, which is summarised in Fig. 1. The problem that is being addressed is summarised in Fig. 1(a), how does a person located in Site A, locate the best expertise to solve a specific problem? The person's local network will only extend to within the site, and therefore expertise in other sites can not accessed. It should be remembered that sites can share common problems, but not necessarily be easily accessible to each other. For example within the academic community, a question on robotics could easily draw on expertise from either a Department of Mechanical Engineering or a Department of Cognitive Physiology. While the

sites may not form a cohesive social network, they do share common sets of resources, including e-mail, phone books, publication and report repositories, Fig. 1(b). In our approach to EF systems, these information repositories are used to identify the required expert, Fig 1(c).

3 How do we identify an expert?

In an academic organisation an expert will be the person who has the most publications, largest number of grants, and 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.

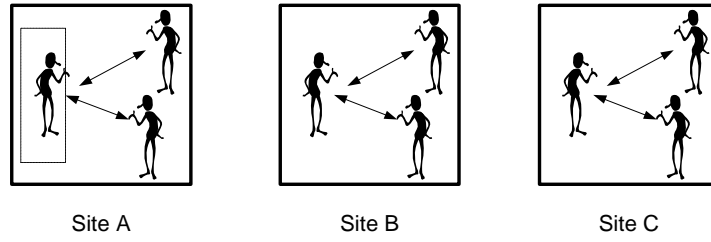
However when a person wishes to contact an *expert*, there are additional social factors that need to be taken into account. Without these factors, the single expert will be swamped with queries for everyone ranging from Undergraduates to Vice-Chancellors. The appropriate person depends on the query and the users 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 used for the recommendation available 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.

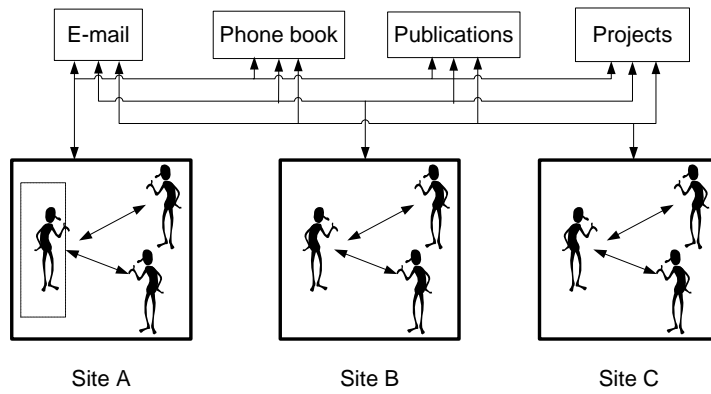
4 Implementation of the Expertise Finder

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. Therefore it is apparent that the high level steps that any system should take to identify an expert will be unique on each occasion.

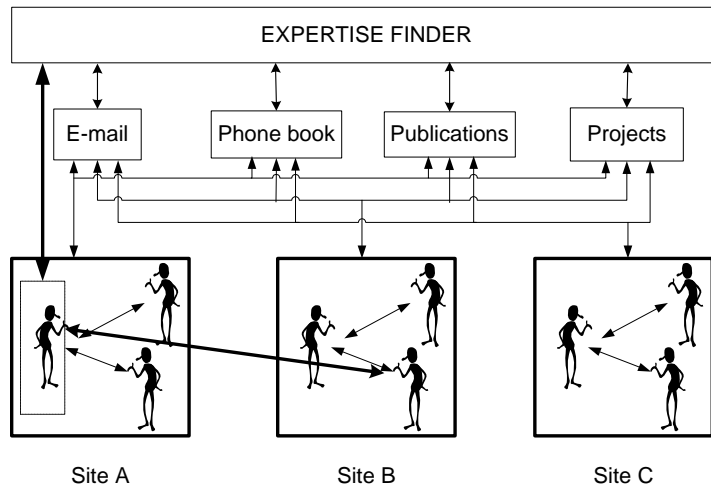
In order to communicate with each other agents use a shared understanding of a domain called an ontology. Ontologies are a conceptualisation of a domain into a form which can be understood both by humans and computers. A well known definition is *an ontology is an explicit specification of a conceptualisation*. [8]. Ontologies provide a mechanism to allow communication and interaction about a real world domain. They remove ambiguity from language through careful



(a) Local social network independent of each other



(b) Local groups are known to organisation wide electronic resources



(c) Expertise Finder finds people-to-people links regardless of location

Fig. 1. Overview of the Expertise Finder

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 within the IAM group but extended for this application. They represent the activities and people in our research group. A detailed explanation of their design and implementation can be found in [13].

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 2 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 outlined in solid.

Source Agents (the Academic Publications and Directory Services agents) are designed to represent sources of information and data within the organisation. These can range from the simple, an agent that understands the data stored in the internal phone book, to more complex knowledge such as an agent interface to a publications database. In the diagram we include examples of some of the ontological predicates the agents support. For instance the Directory Services agent can answer queries about the location of people or return all of the people with a certain phone number.

The EF application is based on a previous agent application, the Dynamic CV [13]. This application used the notion of query recipes to dynamically construct a Curriculum Vitae or other page about a person. For instance for 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 3 shows the result of a CV query.

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.

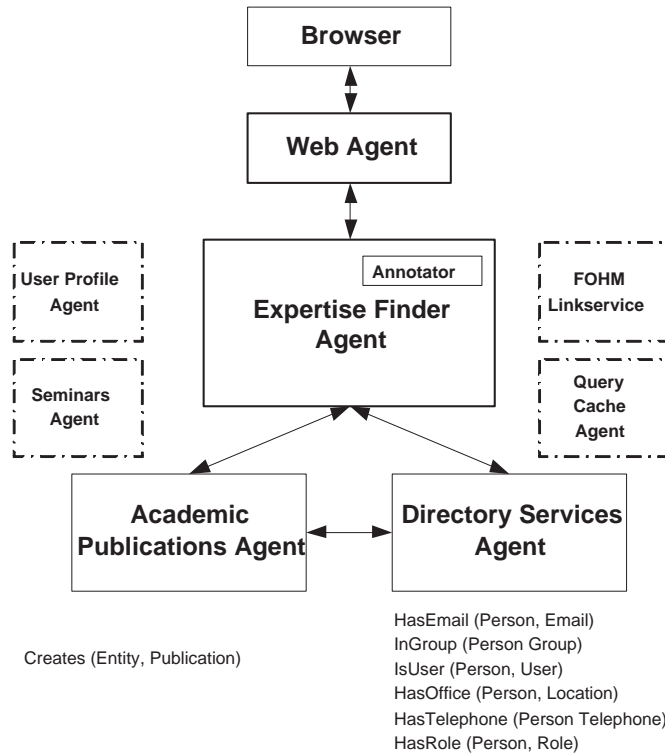


Fig. 2. Expertise Finder architecture. Typical predicates used are given below the respective agents

5 Running the system

The current application of the EF is used to find people using the scientific publication repository within the authors' Department. The goal being to aid people to find experts on a topic amongst the people in the Department. 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

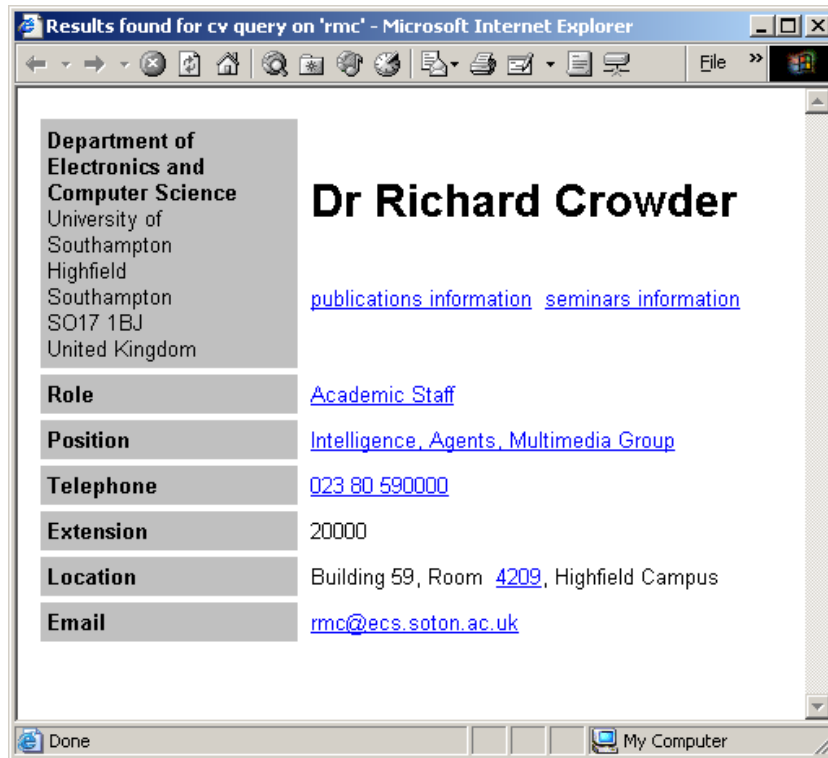


Fig. 3. The Dynamic CV agent system found agents to fill in query templates. There was no attempt to understand or use the information that is returned.

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 less useful to the user.

6 Discussion

The results page, Fig 4, 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.



Fig. 4. The results of the prototype EF, the first ten publications used to rank the experts are given unranked at the top of the page.

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 resultant list of experts relates well to our knowledge of the group and individuals' profiles. The system invariably finds more senior members of staff and less of the PhD students and researchers because they tend to leave after a shorter period of time and have fewer publications. 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 profile 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. The use of peer reviewed document and not an individual's own documentation gives our approach to EF results in a high degree of trust to the recommendations. We are currently considering the use of this system within industry, using technical reports and similar controlled documentation. By the use of corporate data resource we will maximise trust, and minimise possible confidentiality issues. The latter issue will be address by using only information supplied to the organisation as part of a persons employment. 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 rational, and we will also undertake further evaluation using exemplar information resources.

References

1. M. Ackerman and D. McDonald. Answer garden 2: Merging organisational memory with collaborative help. In *Proceedings of ACM 1998 Conference on Computer Supported Cooperative Work, CSCW 96*, pages 97–105. ACM, New York, 1996.
2. E Auriol, R M Crowder, R J McKendrick, R Rowe, and T Knudsen. Integrating case-based reasoning and hypermedia documentation: An application for the diagnosis of a welding robot at Odense steel shipyard. In *Proceeding of the International Conference on Case-Based Reasoning (ICCB'99)*. 1999.
3. I Becerra-Fernandez. The role of artificial intelligence technologies in the implementation of people-finder knowledge management systems. In *Proceedings of the 2000 American Association for Artificial Intelligence Spring Workshop*. AAAI, March 2000.

4. K. Bollacker, S. Lawrence-S, and C. Giles. Citeseer: an autonomous web agent for automatic retrieval and identification of interesting publications. In K. Sycara and M. Wooldridge, editors, *Proceedings of the Second International Conference on Autonomous Agents*, pages 116–23. ACM, 1998.
5. P Chandrasekaran and A Joshi. An expertise recommender using web mining. In *Proceedings of AAAI 14th Annual International Florida Artificial Intelligence Research Symposium*. 2001.
6. R. Crowder, G. Wills, I. Heath, and W. Hall. Hypermedia information management: A new paradigm. In *3rd International Conference on Managing Innovation in Manufacture*, pages 329–34, University of Nottingham, 1998.
7. Y. Goa, I. Zeid, and T. Bardez. Characteristics of an effective design plan to support re-use in case-based mechanical design. *Knowledge based systems*, 10:337–50, 1998.
8. Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSL-93-04, Knowledge Systems Laboratory, Stanford University, August 1993.
9. D Khadilkar and L Stauffer. An experimental evaluation of design information reuse during conceptual design. *Journal of Engineering Design*, 7(4):331–9, 1996.
10. D. McDonald and M. Ackerman. Just talk to me: a field study of expertise location. In *Proceedings of ACM 1998 Conference on Computer Supported Cooperative Work, CSCW 98*, pages 315–24. ACM, New York, 1998.
11. D. McDonald and M. Ackerman. Expertise recommender: a flexible recommendation system and architecture. In *ACM 2000 Conference on Computer Supported Cooperative Work*, pages 231–40. ACM, New York, December 2000.
12. L Moreau, N Gibbins, D DeRoure, S El-Beltagy, W Hall, G Hughes, D Joyce, S Kim, D Michaelides, D Millard, S Reich, R Tansley, and M Weal. SoFAR with DIM agents: An agent framework for distributed information management. In *Proceedings of PAAM00*. Manchester UK, 2000.
13. M. Weal, G. Hughes, D. Millard, and L. Moreau. Open hypermedia as a navigational interface to ontological information. In H. Davis, Y Douglas, and D Durand, editors, *Proceedings of Hypertext'01*, pages 227–36. ACM, 2001.