Capability Management

G. B. Wills, S. Kim, R. Ashri;

gbw@ecs.soton.ac.uk
Technical Report No: ECS-IAM-02-003, May 2002.
ISBN: 085432772X

Abstract

This report presents the design issues of an agent-based expert finder system. In particular, it focuses on collaborative activities where a multidiscipline team were drawn from the researchers and students at the Intelligence Agent Multimedia (IAM) group.

1. Introduction

This report presents the results of the work undertaken as part of the three-day IAM Fest 2001. The IAM fest is an annual event held within IAM group, within the department of Electronics & Computer Science at the University of Southampton. The IAM fest has two aims:

- To integrate new staff into the group.
- To investigate a multidisciplinary research topic.

The report focuses on the latter aspect, that is, an investigation into building a capability management system, using a team drawn from the many disciplines within the group. Capability management enables you to understand: who knows about a subject within an organisation. Capability management is a large research area, therefore to help focus the team it was decided that the team would concentrate on developing an expert finder system. An expert is a person who has different levels of expertise about different topics [McDonald et al, 1998].

1.1 Background

A dynamic Curriculum Vitae agent system has been developed by Weal et al [Weal 2001]. The drawback with this system is that all the communication is through a single agent called the dynamic CV agent. However, this system was used as an initial discussion point, in examining the way in which an expert finder could be developed. An expert finder in this context is a system that would suggest a list of people to be contacted who may be able to answer a specific query.

It was quickly realised that to try and enhance or adapt the dynamic CV system would be difficult within the allocated time of this project. It was therefore agreed that the group would examine the necessary features of a general agent based system, in order to achieve the same general aims. By re-examining the problem with a multidisciplinary team, a different solution might be achieved.

A capability management system must harness the functionality of a wide range of disparate systems that may reside in different operating environments and administration domains. This means that both the lower-level technical requirements for accessing each system as well as the higher-level interaction protocols with each system will differ. Furthermore, the relationships are not static but must evolve and adapt to meet changing access requirements and the addition or removal of resources. Finally, the information gathered typically undergoes a number of alternative analysis techniques in order to provide the best result to the end user. All these requirements lead to a system design that is complex and needs to takes into consideration interactions between a range of subsystems that change over time. As a result the software engineering challenge is significant and requires effective tools to be addressed.

Based on the above rationale we have decided to adopt an agent-oriented software engineering (AOSE) approach to the design of capability management systems. The main abstraction offered by this paradigm is that of an *agent* as an encapsulated computer system, situated in an environment and capable of autonomous problem solving action [Jennings, 2000]. Autonomy implies the ability to decide what to do next based solely on the inputs from the environments and the agent's current state but without human or other intervention. In addition to autonomy the other qualifying characteristics of agents, as discussed by Wooldridge and Jennings [Wooldridge & Jennings, 1995] are pro-activity (behaviour which is directed towards the achievement of specific goals), reactivity (the ability to respond to changes in the environment) and social ability (the interaction with other agents). Large, distributed systems are then decomposed into sets of interacting agents, leading to *multi-agent* systems where the organization and relationships between individual components of the system reflect more closely the true nature of the environment and are thus able to cope better in it. The communication between agents is achieved through the use of agent communication languages that lie at a typically high semantic level [Genesereth & Ketchpel, 1994].

The abstraction of agents and multi-agent systems is especially suited to the area of capability management systems. The various subsystems can now be represented by goal-directed agents that are able to deal with the specific access policies and interaction protocols of their operating environment. These systems communicate with other parts of the system using agent communication languages such as KQML [Finin et al, 1997] or FIPA [Burg et al, 2001], which more closely reflect the semantic level of knowledge management ontologies. An agent-based approach also allows for an overall more flexible system since, as resources and analysis techniques change, new agents can be introduced in the system or the old ones replaced without affecting the system as a whole. Of course, it could be claimed that similar results could be produced with alternative techniques such as object-oriented software engineering. The crucial difference, however, is the *mindset* that AOSE offers from the outset [Jennings, 2000]. AOSE should not be considered a replacement of object-oriented techniques but rather an extension that deals with large, complex systems.

The challenges of AOSE lie in the design of effective modelling mechanisms and methodologies for individual agents and the system as a whole, as well as the development tools for implementing the design. There has been a significant amount of work towards the design of effective modelling mechanisms [Luck, 1999] methodologies (Wooldridge et al 2001, de Loach 2001, Kinny et al 1996] and tools for agent development (e.g. Nweana et al 1999, Subranhmanian et al 2000, Bellifemmine et al 2001). In addition, modelling techniques such as "smart" [D'Inverno & Luck, 2001] can provide the theoretical underpinnings required for a better understanding of the approach.

2. Initial System

The interdisciplinary team first agreed on a very high level approach to be taken, and a block diagram of this initial system design is shown in Figure 1. The first area to be investigated

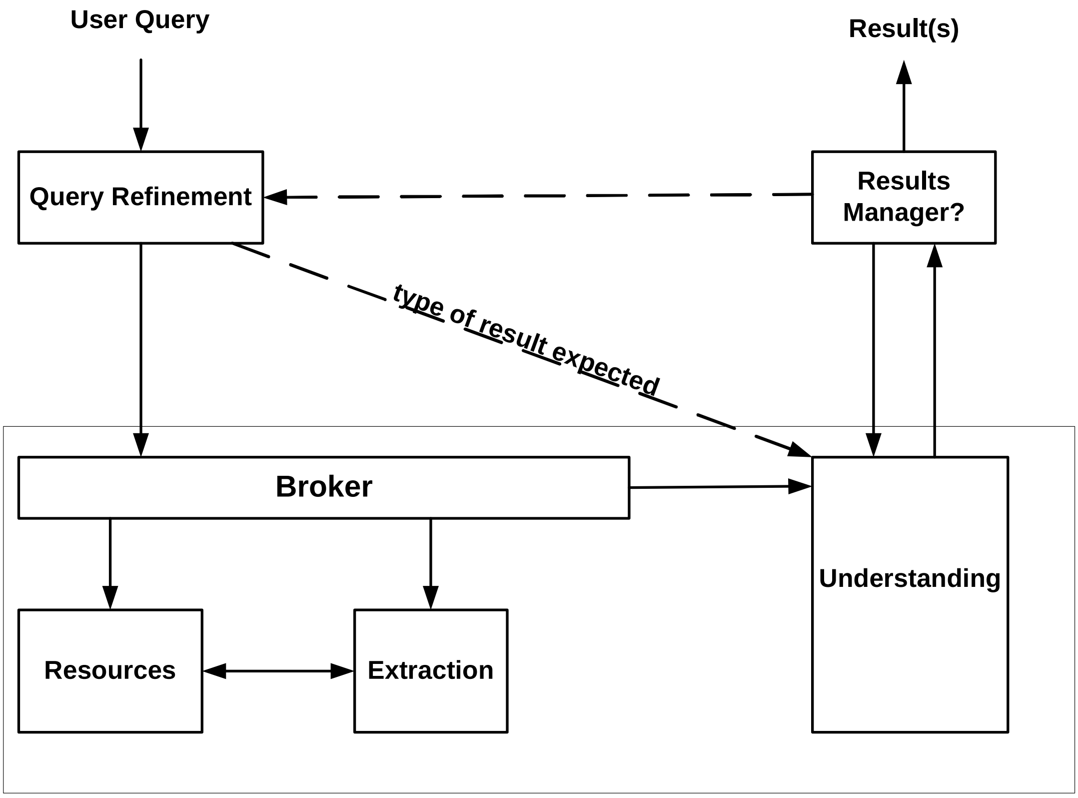

Figure 1 Block diagram of the initial system architecture.

was that of query refinement. It was the view of the team that it would be difficult to process any request from the user unless it could be refined into a format that an agent-based system could understand. Note that the query is not decomposed but refined, meaning that the given query is customised according to user information needs. At this stage, there is no limit on the types of query that could be asked by the user or the types of interface that could be used. Table 1 shows the types of questions that users would typically ask in the context of finding an expert. These questions are grouped under six headings of how, where, when, what, who, and why, each of which are associated with a set of example queries.

How	**When**
How do I submit a thesis? How do I go about submitting a technical report? How do I report a fault with my computer? How can I contact the VC?	When will Wendy next be in? When are the forced shutdowns? When is the next hypertext conference? When is the next seminar?
Where	**Who**
Where is Wendy’s office? Where is the next hypertext conference?	Who else is interested in (agents, hypermedia, etc…)? Who is responsible for cleaning the (microwave, kitchen…)?
What	**Why**
What does AKT mean? What was the name of the last head of group? What are the specialist fields of the next head of department? What is MMRG now called?	Why is the MMRG group now called IAM? Why did MMRG move from building 16? Why did the error XXX happen on my computer?

Table 1 Typical questions a user may ask when trying to find an expert.

The type of user interface could range from a simple command-line input to an agent-based anthropomorphic user interface [Power 2002].

The Query refinement agent (or sub-system) takes the query and then applies the usual processing to produce a query that could be processed by an agent system. The usual processes of query refinement are: stemming, removal of stop-lists, and the use of contexts by means of synonyms, thesaurus, or hyponyms. The result is a list of phrases and keywords and the query focus.

Once the query has been refined, the associated phrases and keyword-list are transferred to a broker that will query a number of documentary sources. At this stage, we have not specified any particular analysis or search mechanisms. Also at this stage we assume that there would be an appropriate agent that would know how to interrogate each of the different types of documents.

The next stage is to identify the documentary evidence and to identify which evidence would be required to support each type of questions. Table 1 summarises the type of documentary sources required to answer the different types of questions.

How	**When**	**Where**
Publications: Papers Minutes Reports Handbooks Project pages Home Pages FAQ	Diaries Publications: University Calendar Bulletin Minutes Personal Home pages Departmental/Group pages News groups E-mails WWW Pages Human Resources	Diaries Publications: University Calendar Bulletin Minutes Reports Personal Home pages Departmental/Group pages News groups Telephone Directories University WWW Pages Human Resources
Who	**What**	**Why**
Publications: Papers Minutes Reports Bulletin Handbooks Personal Home pages Departmental/Group pages Project pages News groups E-mails WWW Pages Human Resources FAQ	Diaries Publications: University Calendar Bulletin Minutes Reports Papers Handbooks Personal Home pages Departmental/Group pages File Store (servers) Project pages News groups E-mails Telephone Directories University WWW Pages Human Resources FAQ	Diaries Publications: Reports Papers Personal Home pages File Store (servers) Departmental/Group pages Project pages News groups E-mails University WWW Pages WWW resources FAQ

Table 2 The type of documentary sources required to answer the different types of questions.

Based on the type of query, the user expects a particular type of answer, and Table 3 shows an example mapping between the type of question asked and the form of the reply. This information is then passed to the Understanding agent (Figure 1), which receives the results from the broker in an unstructured format. The Understanding agent, based on the mapping, will then format the result accordingly. Hence, irrespective of the source media or formatting, the results agents will receive the information from a particular query in an appropriate format.

Query Type	Map	Result Required
How	Information	Document or Information
When	Time, Name, Information	Time
Where	Name, Information	Place
Who	Name, Appellation, Information	Person
What	Focus of Query, Complement (constraints)	Answer
Why	Focus of Query (motive) ,Complement	Answer and Document/Information

Table 3 How to map the type of questions, resources and results required.

The Understanding agent will pass the results onto the Results Manager (RM) Agent. The RM agent will then decide to:

1. Pass the results onto the user.
2. If there is a compound query, wait until all the answers are received.
3. Ask for a further query to be made.

3. Refinements to the Initial system.

The next interaction of the system design included the development of user profiles. The User Profiler agent would learn the preferences of the user and pass this information onto the Query Refinement agent and the results agents. In addition, the User Profiler agent would also continually monitor the user interaction with the system to keep the user's profile up to date.

The user profile passed from the User Profile to the Query Refinement agent is used to shape the type of query asked. Similarly, the user profile information passed from the User Profile to the Results Filter would influence the filters applied to the results before they are passed back to the user. The filters are used to rank the results, for instance if people are returned, it will suggest people that are in the same location or department at the top of the list. The term filter has many connotations and was changed to sieve in the final system as this was felt to reflect more accurately the process undertaken. The order that the results filters are applied depends on the user profiles and user information needs. Table 4 shows the type of filters to be applied in related to the documentary sources.

People	Technical info	Background	Current Practice
Location Site Status	Relevance Date of Origin	Relevance	Relevance Date of Origin Context (project)

Table 4 List of filters (sieves) applied to results.

There are additional heuristics when the results are related to people. That is, since people tend feel less threatened when asking questions of colleagues, the positions in organisational hierarchy can be exploited.

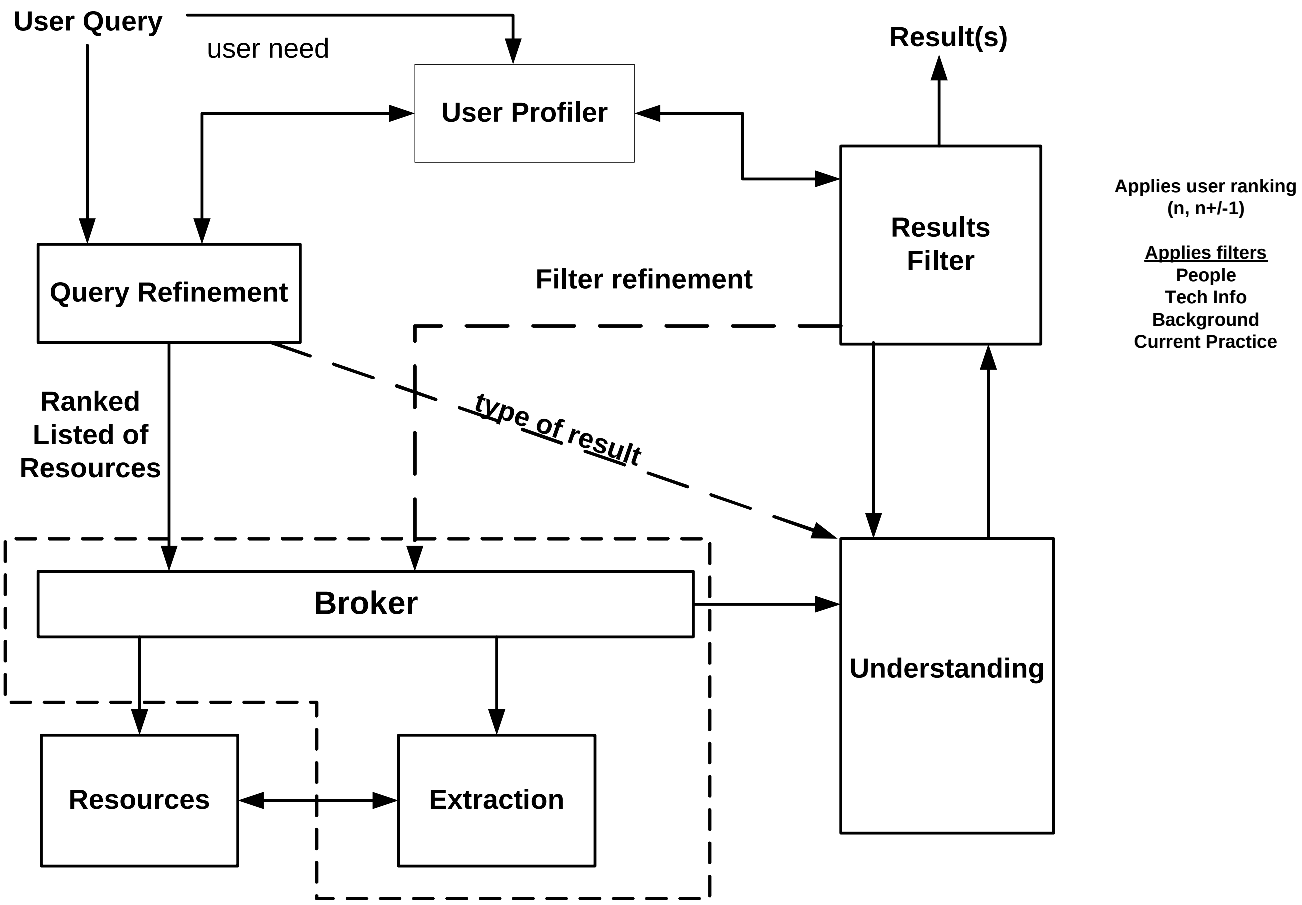

Figure 2 Block diagram of the refinements to the initial system.

The other minor changes were that the resources were seen as external to the system and could be in any location on the intranet or internet, so these are drawn separate from the broker and extraction agents.

4. Final System

The final system block diagram is shown in Figure 3, each block represents an agent or group of agents. The goals of some of the main agents are listed in Table 5.

Agents	Goals
Query Refinement Agent	Identify main components of question and compose an appropriate query.
Resources Identification	Identify possible sources of information Identify structure of answer
Results Manager	Broker Access to Resources
Personalization Agent	Construct user profile
User Interface Agent	Communicate question to query refinement agent Inform personalization agent who asked the question Present answers to the person who asked the question
Results Analyser	Sift through results to find appropriate answers to questions

Table 5 Goals of the main agents

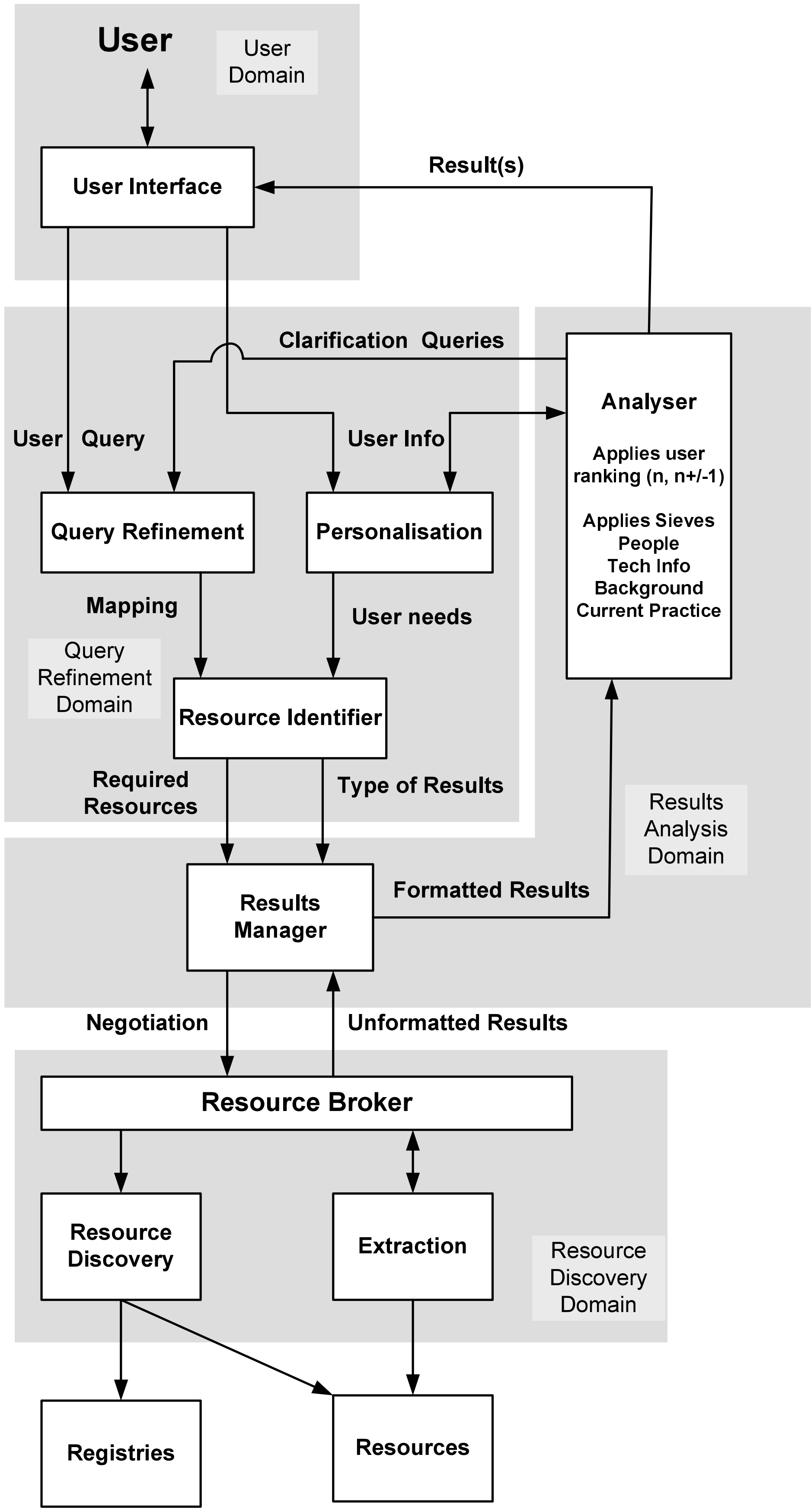

Figure 3 Block diagram of the proposed final system

There a number of minor changes from the previous system, these are:

- A User Interface agent has been introduced. As user may be at a terminal or in a mobile situation, the User Interface agent would be responsible for delivering the answer in an appropriate manner.
- The introduction of a Resource Discovery agent. In addition to the Extraction agent, it is necessary to have a separate agent that would locate new and identify changes in resources.
- The largest change is in the analysis block where the results sieve has been replaced with an analyser agent. This new agent will do more than just sieve the result as in the previous model; for instance, it may run a statistical analysis of the results to check the probability of reliable data and the level of trust associated with the information. In addition the Understanding agent has been replaced with a results manager, as this will manger the results and present them to the analyser in a consistent format.

5. Related work.

Cooper and Rüger point out that from the question type the format of an appropriate answer can be derived [Cooper and Rüger, 2000]. The different type of questions used by Cooper and Rüger: *where, when, who, where, whom, why, describe and define*, of which *who* and *whom* were the most difficult type of queries to answer and required additional heuristics. Another set of heuristic was used to weight and rank the answers by using the information retrieval algorithm.

Kanfer et al have used an agent based system to recommend people from within the user's own social network (Kanfer et al, 1997). The paper emphasis the social nature of communication, and that people prefer to contact people they know or are acquainted with, when asking for help. This work is supported by the study undertaken by McDonald et al in which the social, cognitive and information aspects of the system play a key role [McDonald et al, 1998]. McDonald et al observed that in the social context of any expertise finder systems, the user has two problems to solve, expert identification and expertise selection.

Maybury et al have developed a system that exploits the intellectual products produced within an organisation to support automatic expertise identification (Maybury et al, 2000). The system considers a user as an expert if he/she is linked to a wide range of documents and/or a large number of documents about that topic. It combines multiple evidence demonstrating associations with the user in determining the level of expertise of the user.

6. Conclusions and Future Work

Many of the approaches to capability management systems have focused on the various aspects of the system, for instance, the social aspects, or methods used to infer expertise, etc.

The system designed during the IAM fest, described above, has taken a step back from any particular implementation, and intended to produce a generalised agent based approached to finding an expert in an organisation or on the world-wide-web.

Many of the agents shown in the block diagrams (Figure 1, Figure 2 & Figure 3) will require further detailed design. At which stage the advantages and disadvantages for a particular statistical technique or the exact heuristic to be used can be decided.

7. Acknowledgements

The authors acknowledge the contribution made by the IAM group personnel who formed the Capability management group in the IAM fest, these are: Harith Alani; Richard M Crowder; Cora B Excelente-Toledo; Gareth Hughes; Martina T Johnson; Fabiola Lopez Y Lopez; Mike M Luck; Yannis Kalfoglou, Stuart E Middleton; Muan H Ng; Georgia Roidouli; Yee W Sim; Kate P Dickens; Panchit Longpradit; Hugh Davis.

8. References

Bellifemine F., Poggi A., Rimassa G (2001) JADE - A FIPA2000 Compliant Agent Development Environment. *Proceedings of the Fifth International Conference on Autonomous Agents, pp 216--217*

Burg B, Dale J, Willmott S. (2001) Open Standards and Open Source for Agent-Based Systems. *Agentlink News, No. 6, pp2-5*

Cooper RJ, Rüger SM. (2000) A Simple Question Answering System. *Proceedings of TREC-9, NIST (Gaithersburg, MD, 13 - 16 Nov 2000), NIST Special Publication.*

Finin T, Labrou Y, Mayfield J. (1997), KQML as an agent communication language. *Software Agents, editor Bradshaw, J. MIT Press.*

Genesereth MR, Ketchpel SP. (1994) Software Agents *Communications of the ACM, Vol. 37, No. 7, pp48-53*

D'Inverno M and Luck M, (2001) Understanding Agent Systems, *Springer-Verlag, Springer Series in on Agent Technology*

Jennings NR. (2000) On agent-based software engineering. *Artificial Intelligence, Vol. 117, pp277-296.*

Kanfer, A., Sweet, J. and Schlosser, A. (1997) Humanizing the Net: Social Navigation with a "Know-who" E-mail Agent, *In Proceedings of the Third Conference on Human Factors and the Web, Denver*.

Kinny, D. and Georgeff, M. and Rao, A. (1996) A Methodology and Modelling Technique for Systems of BDI Agents. *Lecture Notes in Artificial Intelligence Vol. 1038 pp56-71*

de Loach SA. (2001) Analysis and Design using MaSE and agentTool *Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Concference (MAICS 2001), Miami University*

Luck, M., (1999) From definition to development: What next for agent-based systems *Knowledge Engineering Review Vol 14, No. 2, pp119-124*

Maybury, M., DAmore, R. and House, D (2000). Awareness of Organizational Expertise, *MITRE Technical Papers October 2000 On-line at http://www.mitre.org/support/papers/tech_papers99_00/maybury_awareness/index.shtml*

McDonald, D. W. and Ackerman, M. S. (1998) Just Talk to Me: A Field Study of Expertise Location, *In Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work (CSCW), place, pp.14-18.*

Nwana H, Ndumu D, Lee L, Collis J. (1999) ZEUS: A Tool-Kit for Building }istributed Multi-Agent Systems *Applied Artificial Intelligence Journal, Vol .13, No. 1, pp129-186.*

Power G, Wills GB, Hall W (2002) User Perception of Anthropomorphic Characters with Varying Levels of Interaction *HCI2002* - http://cise.sbu.ac.uk/hci2002/

Subrahmanian V.S, Bonatti P Dix J, Eiter T, Kraus S Ozcan F, Ross R. (2000) Heterogeneous Agent Systems. *The MIT Press*

Weal MJ, Hughes GV, Millard D, Moreau L. (2001) Open Hypermedia as a Navigational Interface to Ontological Information Spaces. Proceedings of the Twelfth ACM Conference on Hypertext and Hypermedia HT'01, Aarhus, Denmark August 2001, pp 227-236. On-line at *http://www.bib.ecs.soton.ac.uk/data/5161/pdf/ht01.pdf*

Wooldridge M, Jennings NR. (1995), Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review, Vol. 10, No. 2, pp115-152.*

Wooldridge M, Jennings NR, Kinny D. (2000) The Gaia Methodology for Agent-Oriented Analysis and Design. *Journal of Agents and Multi-Agent Systems, Vol.3, No. 3, pp 285-312*