KMi
KNOWLEDGE MEDIA INSTITUTE

Oracles, Bards, and Village Gossips, *or, Social Roles and Meta Knowledge Management*

Simon Masterton and Stuart Watt

KMI-TR-82

August 1999

The Open University

Oracles, Bards, and Village Gossips,
or, Social Roles and Meta Knowledge Management

Simon Masterton and **Stuart Watt**

Knowledge Media Institute

Open University, Walton Hall, Milton Keynes, UK. MK7 6AA

Email: S.J.Masterton@open.ac.uk, S.N.K.Watt@open.ac.uk

"Where is the wisdom we have lost in knowledge?
Where is the knowledge we have lost in information?"

T. S. Eliot, *Choruses from the Rock*

"And 'T. S. Eliot' is an anagram of 'toilets', so he should know"

Alan Plater, *Oliver's Travels*

Abstract

Knowledge management systems are used widely in many different organisations, yet there are few models and theories which can be used to help introduce and apply them successfully. In this paper, we analyse some of the more common problems for knowledge management systems. Using this background, we adapt models and theories from social and organisational psychology and computer supported collaborative work, and discuss a variety of different knowledge management systems in these contexts. We argue that knowledge management systems routinely adopt different social roles within an organisation, and that these social roles can have a major influence on a system's acceptability. With these principles in mind, we draw out some general practical lessons, and a 'character space' framework, which can help to inform the design of future knowledge management systems, so as to minimise the problems of acceptability within a given organisation.

1. Introduction

Like many things in life, knowledge management is important and useful, but hard to do well. For this reason, a large number of tools are often used to help, some of which have been specially developed for the purpose. More often than not, these tools are never really assessed to see how good they are for the purpose, but even when they are (e.g. Ackerman, 1994; Masterton, 1997, 1998), the lessons rarely make it into other systems.

In other words, the field of knowledge management has a knowledge management problem — an issue we'll come back to later in this paper. There is a lot expertise out there, spread out through people's experiences with many different systems. It is time to take stock, to look at some of the problems that happen in people's actual use of knowledge management systems, and to draw some lessons from these problems — lessons which can be used to inform the future design of this kind of system.

Knowledge management is an issue for organisations, so it inevitably involves many different people doing many different jobs. Some may be mostly providers of knowledge, for example, while others may be organisers of it, or users of it. Because there are people in different roles involved in the knowledge management processes, knowledge management is an example of "multiple role groupware" (Watt, 1993), and has a tendency to "role conflict" when the needs of the different participants are not commensurable. This often shows up as a disparity between who does the work and who gets the benefit. A common reason for the failure of multiple role groupware systems is when the system itself is biased in such a way that it reinforces the role conflict (Watt, 1993; Grudin, 1994) — and knowledge management systems are easy prey to the same problem.

To put it simply, intelligent agents are an elegant solution to this problem. Without going into them in too much detail, intelligent agents are computer programs that are constantly running around acting on the behalf of one or many people (Maes, 1994; Watt, 1997). The advantage of intelligent agent systems is that the normal *quid pro quo* rules of human social interaction need no longer apply. Agents can be constantly on the lookout for relevant information, or for links between different bits of information, and although they never do it as well as a person would, they don't expect to be paid for their trouble. They can make admirable assistants, in principle (Maes, 1994), but they can easily perform other, very different, but useful, roles in knowledge management, for example, as matchmakers (Foner & Crabtree, 1996), newspaper editors (Domingue & Scott, 1988), librarians (Watt, 1998), bards (Masterton, 1997, 1998), and village gossips (Krulwich & Burkey, 1996, 1997). In effect, they can take the roles where there isn't any benefit, leaving people in a position where they always get the best from the system.

This justifies agents as a technology for knowledge management, but in another sense, there is a kind of inevitability about it. Not only is the human-human relationship fundamentally social, so is the human-computer relationship (Nass *et al.*, 1994).

On the whole, then, there is a place for agent systems — although very understated, not personified, and rather role-centred agent systems — within knowledge management. But there are dangers. Overblown, very anthropomorphic agent systems are rarely successful outside the fields of entertainment and education. So there are risks. But careful design can, both in theory and in practice, offer agent systems that work effectively in knowledge management. Indeed, we'll come back to a few of these later, in section 4.

Independently of whether or not they take an agent-based approach, knowledge management systems are very diverse, playing many different roles within an organisation. It is hard to take lessons from a system in one role and apply those lessons in other roles, because these different roles shape very different patterns of interaction between the people and the systems involved. For this reason, knowledge management systems pose some rather special and rather hard problems for those who wish to assess the usability of these systems, and learn from it.

In this paper we're going to look at the usability of knowledge management systems a bit more systematically. We're going to look at some of the most common problems with knowledge management systems, and how some existing systems have tried to alleviate those problems. We'll show this in practice by discussing some example knowledge management systems as case studies. With this foundation, we'll draw out some of the lessons to be learned, and argue that an awareness of opportunities offered by new agent technology, and an awareness of the importance of usability, acceptability, and the social relationship between system and user, can significantly enhance the prospects for knowledge management systems. But first, let's look at the problem in a bit more detail.

2. What’s the problem with knowledge management systems?

Perhaps the most significant issue for knowledge management systems is the problem of how to get them accepted — and used — in practice. Very often there is a positive initial response to a new knowledge management system, many people will try it out a few times. Unfortunately, for one reason or another, more often than not the use of the system will begin to tail off pretty rapidly. The diagram in figure 1 below is pretty characteristic of the pattern of use of a new knowledge management tool. It is, perhaps, surprising how many special purpose knowledge management systems are regarded as successes, even when the number of people using the system has perhaps tailed off practically to zero. On the other hand, more general systems such as the World Wide Web, while used as tools for knowledge management, often fail to meet their initial promise. What is going on?

Insert figure 1 about here.

There are many different reasons for this tail-off in the pattern of use. Often, though, there is a common theme. For one or several reasons, people are put off using the system, or put off participating in the knowledge management processes afforded by the system, because they do not feel the system helps them in the right way. Some of these reasons are technical, the system may just be too hard to use, so that the benefits reaped are not worth the effort expended. On the other hand, many of the reasons why a person might not participate may be social, for example, people often feel embarrassed when communicating their opinions to others. It is because the problems for knowledge management systems are often not so much technical as social that it is especially important to learn lessons from previous, less successful, approaches to the area.

In general, the most common reasons for the non-acceptance of a knowledge management system break down into three categories.

- **Task related problems.** When the system makes tasks harder than they would be otherwise, perhaps adding to a person’s daily load, people will often stop using the system.
- **Culture related problems.** When the use of the system doesn’t fit with the culture of the organisation, or of the individuals using the system, this may put people off using the system.
- **Individual related problems.** When personal space is being infringed (or is perceived as being infringed) in one way or another, people will often stop using the system.

Within these broad classifications, we can show a few clearer examples of these different kinds of problems with knowledge management systems. But before doing this, it is worth pointing out that very few of these problems are specific to knowledge management systems. Task problems are common to practically all computer tools of one kind or another. Problems with cultural fit and individual feelings are also very common within any groupware or computer-supported collaboration system. Knowledge management just happens to usually involve many people collaborating, and collaborating on one of the most sensitive elements in our lives, our knowledge. It would, therefore, be rather surprising if these knowledge management systems did not tend to suffer from the same shortcomings as other collaborative systems.

Grudin (1994) sets out these as the eight main problems for the developers of collaborative ‘groupware’ systems:

1. **Disparity in work and benefit.** People may have to do extra work for little or no benefit; when this happens, the system will often fail. This is the problem of “role conflict” (Watt, 1993).

2. **Critical mass problems.** Some group applications really only work when a ‘critical mass’ of people use them.

3. **Disruption of social processes.** Group applications may break existing social rules and roles within an organisation or institution.

4. **Exception handling.** Group interaction is very complicated, and a lot of repair and improvisation may happen; applications need to be able to accommodate this.

5. **Unobtrusive accessibility.** In group interaction, some infrequently used features are still very important (e.g. privacy settings) and must always remain accessible.

6. **Difficulty of evaluation.** Because of the number of people involved, and the cultural embedding of the interaction, it is very hard to evaluate collaborative systems properly.

7. **Failure of intuitions.** Intuition is not a reliable guide when conceiving or designing any system, although it works rather more often for single-user systems.

8. **The adoption process.** Group systems need to be introduced into workplaces much more carefully than single-user systems.

So knowledge management systems are inevitably groupware systems; not only do they often connect people to people directly, even when they don't, they connect people to other people indirectly, through sharing knowledge (O'Leary, 1998). All the problems of groupware, then, pile up on knowledge management systems, and a successful knowledge management system must succeed as a groupware system as well.

What makes the problem of knowledge management *really* awkward, though, is that the stuff which is being shared is often felt to be both valuable and quite personal. Quite simply, people tend to get very upset if knowledge management is handled badly — far more upset than if, for example, a shared authoring system had simply failed to work properly. Carelessly treating knowledge in people's heads as just another asset feels like — and is — a kind of violation of people's personal space.

Bearing this in mind, we can look at some of the most common problems of typical knowledge management systems in a little more detail. This list is not exhaustive, but was derived by a reasonably systematic analysis of the way people talked about a significant number of different knowledge management systems.

Task related problems

1. **"I need special software."** Many knowledge management systems depend on special software. Any special software has some learning curve; it may take some time to learn to use the software at all, and possibly considerably longer to grasp the cultural practices needed to collaborate with others using this software. Only if the benefits clearly outweigh the effort needed is this kind of system likely to be successful (Watt, 1993; Grudin, 1994).

2. **"I have enough to do each day as it is."** People are usually pretty busy without having to use yet another tool, unless it really helps them and saves them time. This links to Grudin's (1994) point about the disparity in some systems between those who do the work and those who reap the benefit.

3. **"I can't find anything useful anyway."** Some knowledge management systems, particularly the ones with big archives, can be rather variable in quality. There may be diamonds in there, but

they can be very deeply hidden and hard to find. This puts a large burden on anybody who wants to find the knowledge.

4. **"I don't know how to do the task very well."** Even when the system doesn't have an apparent learning curve (for example, when no new software is needed) it may take some time to learn to use effectively. In the meantime, if they don't help people from the first, people can be put off before they begin to reap the benefits of the new system.

Culture related problems

5. **"I need to change the way I work."** Sometimes, knowledge management systems expect people to change the way that they work. A good example is when using the knowledge management system requires people to use software that doesn't fit with their organisation's culture.

6. **"I don't think this system is right for this organisation."** Some systems, particularly those which have built in assumptions about the structure, work practices, or culture of the organisation they are intended for, suffer from particular problems.

Individual related problems

7. **"I feel my privacy is being invaded."** Sometimes, knowledge management systems make people feel that 'their' knowledge is being appropriated and used by others, without due acknowledgement or benefit in kind.

8. **"I'm shy."** This is very common! Many people are quiet even in face-to-face meetings, and new media don't help. Trying to give everyone a chance to participate comfortably in the knowledge management processes may be very hard. KMi Planet (Domingue & Scott, 1998) shows the significance of shyness particularly clearly; few people were happy to write stories about themselves or their own work.

9. **"I feel the system is not representing my views."** In some knowledge management systems, people put up arguments or documents for public display (for example, the Knowledge Sharing Environment, Davies *et al.*, 1998). In these systems, there is a potential entanglement between the person and the views of the document or argument they are putting forward. This is a special

case of a deep problem: human social interaction is very rich and complex, and many subtleties can be lost when these interactions are *re*-presented in a knowledge management system.

10. **"I can't control the system."** Some knowledge management systems record 'profile' information about each individual user, for that user's benefit. However, unless it is very easy for people to change this information, if it is incorrect the system can quickly become a burden.[1]

It is very important to note that none of these points is specific to computer supported knowledge management. All of the same issues apply when human knowledge management systems are put into place. Intellectual asset audits carried out by people still cost time, invade privacy, and may even run against the culture of the organisation. They are vulnerable to exactly the same kinds of problem.

Looking back at these problems, there is a clear pattern; while systems may fail for technical reasons, even a system that is technically perfect may fail when it is introduced in the wrong way, or in the wrong context.

Knowledge management is itself a rather diverse problem and the tools that have been used to support its processes do many different things. Some, for example, help people get in contact with others with similar interests (Foner & Crabtree, 1996). Others help to keep people informed about new information as it becomes available. In other words, knowledge management tools can perform many different roles. Here are some of the more common roles for knowledge management tools and systems:

- Knowledge management systems as assistants (e.g. Abecker *et al.*, 1998)
- Knowledge management systems as matchmakers (e.g. Foner & Crabtree, 1996)
- Knowledge management systems as librarians (e.g. Watt, 1998)
- Knowledge management systems as reporters (e.g. Domingue & Scott, 1998)
- Knowledge management systems as editors (e.g. Domingue & Scott, 1998)
- Knowledge management systems as oracles (e.g. Ackerman, 1994)

[1] For example, one of us (SM) constantly received information about Olympic skiing events from one information finder system, because this topic seemed to have somehow crept into the profile. Unfortunately, there was no way to remove it, so the system became just another source of junk mail.

- Knowledge management systems as critics (e.g. Fischer *et al.*, 1990)
- Knowledge management systems as bards (e.g. Masterton, 1997, 1998)
- Knowledge management systems as village gossips (e.g. Krulwich & Burkey, 1996, 1997)

Some of these roles deserve rather fuller explanation. We have described Answer Garden (Ackerman, 1994) and similar systems as 'oracles', because, like the Delphic Oracle, people can go to them to get relatively definitive answers to questions they might have. We have described the Virtual Participant (Masterton, 1997, 1998, see section 4) as a 'bard', because, to put it simply, it re-tells stories from the past. We have described InfoFinder and ContactFinder (Krulwich & Burkey, 1996, 1997) as like 'village gossips' because they pass on short snippets of information that they believe you might be interested in, without this information being solicited. These three particular systems all do knowledge management, yet they adopt very different social roles, and, therefore, cannot really be compared. A good knowledge management solution could easily combine all three. Even so, we regard it as a fundamental point that knowledge management systems, to a significant extent, can adopt a variety of social roles within an organisation in much the same way that people can and do (Goffman, 1956).

All these roles are very different, and it would be surprising if the social rules which apply to one role would apply to all the others. For example, assistants and village gossips treat privacy very differently. And you have to do more work to learn from an oracle than from a bard. Yet in some way all these different roles create a culture of knowledge of one kind or another, allowing some people's knowledge to be transmitted to others. These different roles bring out rather different social rules and assumptions in the people who are to use these systems, and, therefore, create very different genres of communication (Yates & Orlikowski, 1992). The communicative genre adopted in knowledge management is shaped, therefore, by the social roles adopted by any human or computer systems involved.

All these differing roles, then, are appropriate for certain kinds of knowledge management, yet none is a complete answer to the problem. Each role offers different kinds of solutions to some of the problems.

3. How do you solve the problems of knowledge management systems?

Most of the problems of knowledge management systems can be solved, or at least mitigated, by taking some special care in the early stages of the design process. Perhaps most important, though, because only some of the problems of knowledge management systems are task related, these other, non task related problems, often can not be solved by a task centred design process (e.g. Lewis & Rieman, 1994). This should not be read as arguing that task centred design is not appropriate when designing knowledge management systems, just that it needs to be complemented with an awareness of the special, non task related problems of this kind of system.

Let's go through the more common problems again, this time looking at some possible solutions to each of the common problems that we've raised.

Alleviating task related problems

1. **"I need special software."** The obvious way to reduce this problem is to build on existing software, for example, electronic mail or web browsers, and indeed, many knowledge management systems work through these media. On the other hand, when special software is needed, it helps if this software is not particularly difficult to learn or to use, and if the software can be used by an individual to help them in their day to day work anyway. Once the software is adopted into people's daily work, though, the software issues of the knowledge management system fade away (Ackerman, 1994).

2. **"I have enough to do each day as it is."** Again, it helps if people don't have to do new tasks, unless they are going to gain from them. There are two solutions to this: either integrate the task with an existing one (such as reading one's email, e.g., Masterton, 1997) or make sure that there is a clear perceived time-saving or effort-saving benefit, compared to the effort involved in participating in the knowledge management processes.

3. **"I can't find anything useful anyway."** In large information systems, things can be difficult to find. Currently, there are two common strategies for dealing with this problem. One is visualisation and the other is the use of agents. The visualisation strategy aims to give people a gestalt of the overall content, allowing them to find what they need quickly and directly. The agent strategy gives people a personal assistant, which knows enough about them to be able to

‘cut to the chase’. There is another important difference between the visualisation strategy and the agent strategy: because agents are both continuous and autonomous, people can be notified when the system itself finds something relevant, without the user having to be involved at all. This can reduce the amount of work involved in using the system, but there is a price to be paid, in that agents are generally not as discriminating as people, and the quality of material found is often lower.

4. **“I don’t know how to do the task very well.”** Sharing knowledge is not easy, as any teacher will tell you. And understanding knowledge shared by other people can be just as hard. There are two approaches to this problem: making the task easier, and giving people lots of help and support. With many knowledge management systems (e.g. Answer Garden, Ackerman, 1994) the user’s who stick with the system learn how to make best use of it, and the tasks become easier with time and practice.

Alleviating culture related problems

5. **“I need to change the way I work.”** There are many solutions to this problem. First of all, if there’s no good reason for people to change the way they work, then the knowledge management system should probably be changed to let them carry on as before. Forced changes that aren’t perceived as beneficial commonly backfire. On the other hand, if there is a good reason to change, there are strategies which can be used (Grudin, 1994). These strategies include, for example, giving people a clear rationale for using the system, handholding, training, and follow-through support. As Grudin points out, these are common strategies for large organisational information systems, and this kind of support is necessary for the acceptance of groupware systems as well. The same goes for knowledge management systems, with a special empathy for those undergoing the change.

6. **“I don’t think this system is right for this organisation.”** Every organisation has a distinct, and usually rather heterogeneous, culture. Yet many knowledge management tools send cultural messages to their users. There are many solutions to this problem. One is simply to change the culture (Grudin, 1994). Alternatively, the tools needed to use the knowledge management system (for example, the browser, or email system) can be selected to fit into the existing organisation, rather than going against its grain (Masterton, 1997).

Alleviating individual related problems

7. **"I feel my privacy is being invaded."** Many knowledge management systems afford an element of privacy, in one form or another. To put it simply, knowledge management systems need to provide what Goffman (1963) calls "interaction shields". Systems which require explicit action for knowledge sharing (such as the Knowledge Sharing Environment, Davies *et al.*, 1998) have less of a problem here than systems which constantly watch a user's actions (e.g. GroupLens, Konstan *et al.*, 1997). In GroupLens, privacy is maintained by offering anonymity.

8. **"I'm shy."** This is not so much a problem to be overcome as to be recognised and respected. However, keeping a feeling of a close and familiar community can help, making it clear precisely *who* knowledge is being shared with, and perhaps even allowing people to remain anonymous through virtual 'ghost writers' (e.g. KMi Planet, Domingue & Scott, 1998).

9. **"I feel the system is not representing my views."** One of the more complex problems of knowledge management is the relationship between knowledge and people. Whenever a person's name is associated with some piece of knowledge, the knowledge reflects on the person and *vice versa*. One way of alleviating this problem is to decouple the name from the knowledge, for example by letting people remain anonymous, or by offering people a language that allows them to remain neutral about the knowledge they are sharing.

10. **"I can't control the system."** Because knowledge management systems tend to be permanently running, they pose special problems for a user who wants to control them. This is not a special problem for knowledge management systems; operating systems suffer very similar problems. Giving good dynamic feedback in control panels and configuration options is a good strategy for alleviating this problem.

It would be folly to suggest that there is a single solution to these problems. But they can, for the most part, be alleviated. But first, let's look at some of these issues in live systems.

4. Studies and Examples

In this section we take the opportunity to look at a few knowledge management systems and how they have approached the problems associated with acceptance. Firstly we will look at Ackerman's

(1994) classic system Answer Garden and the pointers it gives in designing knowledge management systems. We will then look at a system developed by one of the authors (SM) known as The Virtual Participant and the studies associated with that. Finally we will look at the Knowledge Sharing Environment developed by British Telecom, to which we have had some input into the design process.

Answer Garden

Perhaps the classic Organisational Memory system is Ackerman's Answer Garden (Ackerman, 1994). Designed as a way of aiding the search for 'Information in an Organisational Context'. Answer Garden, along with its sister applications LiveDoc and Assist, were used as ways of managing and sharing information within groups of people. Answer Garden was tested in the domain of X Windows programming, with the simplifying assumption of distinct groups of experts and information seekers. This in depth field study used a number of research methods to gain a detailed insight into the users of the system. From these studies Ackerman was able to categorise his users into three basic groups, "tire-kickers", "intermittent users", and "heavy users". The reasons behind these user behaviours and details from other qualitative data can be related back to our reasons for giving up knowledge management systems given in section 2 and how problems can be solved in section 3.

Answer Garden required the users to make two key changes, one task related (additional software) and the other culture related (change in working practices). The software itself was implemented as a separate application run remotely and required users to change their information seeking approach to incorporate it. An important point to gain from this study was that those who had become heavy users had incorporated it into their normal working practices. This is exemplified in a quote from one participant in the study "... I've always got it running. It's just one of my normal suite [of tools]". However for a number of potential users during the study they were unable to take part either because they did not have the technical ability to run the application or could not even connect to the server. Others were put off just at the thought of having to "master another tool".

We make a point that there is a content problem with knowledge management systems reflecting on either, or both, the quality and the access. In Answer Garden the quality was guaranteed by the group of experts, however access still proved a problem for some of the participants. Some found it hard to locate the correct answers, other could not decide if Answer Garden might contain the answer in the first place. One student commented that "There was a learning curve issue in finding out where the good information was" and another said "I was often unclear as to what were legitimate Answer Garden questions."

Ackerman states, correctly, that these are all problems experienced in introducing most software systems. However we feel that they are of particular importance when it comes to any groupware system, which knowledge management systems often are. For these kinds of systems mass acceptance by their intended audience is much more important and directly related to the success of such systems. In Answer Garden this is at least taken into account by the attempt to create "the video-game idea of the self-evident application, where one has only 30 seconds to convince and train the user to operate the program". However we would argue from the standpoint that if such systems are integrated into existing work processes and packages then this task is made all the much easier.

A subtler problem faced by many systems is whether they actually do fit into the social context they are trying to match, which is often distinctly related to how public/private such systems are. The original suppositions embedded in Answer Garden stem from research into how engineers go about the information seeking process. The key point from this was that engineers would attempt to explore all the possibilities before bothering experts so as not to look stupid, and certainly so as to appear knowledgeable about the problem. In a number of studies of Answer Garden like approaches some users would conduct only superficial searches, or even just identify a possible area, and then ask the expert responsible for this area for some help. This then defeats the idea of Answer Garden in its attempt to stop this from happening. Ackerman addressed this problem in Answer Garden 2. We think that part of the cause of this problem is due to the anonymity available to those users who ask questions — this would not be available to them in the equivalent social context. It is also seen by many to be quicker and easier to send an email than to first spend time searching for a solution.

It is not entirely clear from the study why the users behaved how they did, or even whether they would classify themselves as having given up using the system. However from the usage patterns it would not be too far fetched to infer that a number of users had given the system a chance, but had stopped using it. Our observations and deductions on this matter come from the data and feedback presented in Ackerman's (1994) thesis.

The Virtual Participant

The ‘Virtual Participant’ was designed to address a number of observed problems in a computer supported collaborative learning (CSCL) environment, although the experiences with using it are generally applicable to other organisational memories and computer supported collaborative work systems. Initially the project looked at students using FirstClass, an asynchronous, text-based, electronic conferencing system. FirstClass is used to enable groups of students and tutors to keep in touch with each other. Commonly each course using the system will have a number of conferences dedicated to different aspects of the materials. Each year discussions occur on similar topics to those seen previously. However each year the discussions, even when archived and made available as is, are practically lost to the next. The basic solution to this problem is the generation of FAQ’s (lists of ‘frequently asked questions’). However these take effort to generate and maintain, as well as proving generally ineffective.

The aim of the Virtual Participant was to develop a way of storing the core points of these previous discussions and then making them available in the current discussions when relevant to the current context. The approach taken was to develop an initial prototype (Masterton, 1997) to test out a number of these ideas. The results of this initial study was then fed back into the design of a second prototype (Masterton, 1998). The second prototype addressed a number of the problems experienced by its predecessor, and was consequently much more successful.

The main cause for the lack of use of the first prototype was because the user was forced to interact with the system in public by sending a message to a public discussion conference. The intention behind this was to allow other students to learn vicariously from observing these interactions. Instead this approach put students off from using the system. This is not to say that all our experiences were negative. The system was fully integrated into the conferencing environment that the students normally used and appeared as just another user. This avoided the need for additional software on the half the user and reduced the learning necessary to interact with the system. There were additional problems with the system that affected different types of user in different ways. For the tutors there was a strong feeling that they had insufficient control over the system. They did not know the content of the systems knowledge base and had no control or warning about when it would post a message — although the conferences to which it would post were constrained. For the students particularly the system did not conform to the social aspects of the conferencing system —

there were complaints about the messages being ‘too long’ and ‘looking ugly’ (an unfortunate technical constraint limiting it to 72 characters per line, 10 point Times Roman text).

Improving signal to noise and reducing the quantity of material are two key points to the success of these systems. The Virtual Participant database is culled from previous discussions, enhancing the information content level — similar to Answer Garden’s expert generated answers. To then reduce the quantity of material and maximise gain the messages generated by the system are targeted to the context of the discussion thread to which they are posted. The feedback and comments received about the first prototype permitted us to redesign the system to counter the problems experienced.

Prototype number two opened up the ability to completely control the system by the tutors. However when empowered in this way they then chose to forgo this control as it was deemed too much additional effort on their part. To encourage interaction requests from the students to the system were conducted via private messages. Vicarious learning by others was encouraged by the posting of questions, asked by 5 or more students, to the conference.

The Knowledge Sharing Environment

Recently published (Davies *et al.*, 1998) the Knowledge Sharing Environment (KSE) is the latest in a long line of work from British Telecom in this area. The intention of KSE is to provide “closed user groups or communities” with an environment which enables them to more easily share both explicit and tacit knowledge. The KSE approach gives each user their own personal agent. Each agent manages a user profile intended to “model that user’s information needs and interests”. The KSE agent acts as an information management system with the users choosing to add documents they find of interest. At the point of addition the user can annotate the document and select which interest groups the information should be stored with. By making the users responsible for pre-filtering the information added to the database this guarantees high relevance of the information.

KSE supports a number of approaches to enhance the sharing of the knowledge stored within the system. These fall into four clear groups:

1. **Email notification.** When new information is added to the system through your personal KSE agent, this agent then checks with other agents in its ‘local community’. If another user is

identified as being interested in this information, using a simple thresholding function, then an email message is automatically generated to inform them of this.

2. **Keyword retrieval — accessing information and people.** KSE allows access, via a personalised web page, to a search engine. In addition to searching the Web and the Intranet the user has access to two other facilities. The first of these is the ability to search all the pages stored by KSE, the second is a search system to find other users. The aim of the second facility is to aid in the sharing of tacit knowledge by enabling users to identify others with interests in particular areas. Users may also request lists of other users with 'similar' interests to themselves, as well as identifying list of users who may be interested in a particular document. Although KSE does not claim to capture tacit knowledge this approach is intended actively to facilitate the sharing of such knowledge.

3. **What's new?** This facility is intended to help users discover what information has recently been stored. This allows the user to see both what has been added recently and how well it matches their profile. In addition to this the user may add their own annotations to the information, give feedback on a piece of information to modify their profile, examine summaries of the information before deciding to access it in more detail, and as in 2 above they can identify other users who are interested in a certain piece of information.

4. **Interest groups.** By the creation of 'interest groups' and setting the addition of information to a certain group KSE generates information pages for each group. This allows users to keep track of their own specific interest groups, and aids knowledge seeking as a user becomes interested in a new subject area.

The Knowledge Sharing Environment has addressed a number of problems that we have identified with the use of such systems. By limiting the groups to which users belong they have eliminated the risk of embarrassing yourself in front of strangers, but perhaps introduced the greater risk of embarrassing yourself in front of your peers and your boss. However there is no additional software required (the system is interfaced through a web browser), content has been decided upon by the users, and the users have complete control over the system. The only concern with KSE is shared by the implementers of the GroupLens (Konstan *et al.*, 1997), which is how to encourage all users to contribute to the system. From studies of recommender systems (Resnick & Varian, 1997) it is often the case that a small number of users contribute the majority of the information. To give an idea of the trend and example might be 10% contribute 50% of the information, and the next 10% contribute up to 25% of the information. This trend is similar to the usage patterns we have observed with the Virtual Participant. Recommender systems often fail because the top contributors become disenchanted with the system, often through feeling that they are not gaining anything from the system, or that their contribution is not acknowledged. It is often thought that this problem is solved by an appropriate critical mass of users. In the case of KSE the system has been designed for users who already work together independent of the system, reducing the numbers necessary for success.

We have looked at three different systems none of which we would claim to be perfect. However there are clear lessons to be learned from each one which provide guidelines for the design of such systems. These systems exemplify a number of the problems stated in section 2, and solutions stated in section 3. Pooling these experiences and those of others we can define the existing knowledge about managing knowledge — the next section talks about this meta-knowledge management.

5. Lessons Learned: Knowledge Management in Practice

This section looks at the meta-level knowledge necessary to design knowledge management systems. Meta-knowledge about design and implementation problems has been gained by a number of people from their experience in building knowledge management systems. We will first look at some of the guidelines that have already been produced. Then we go on to look at our own lessons learned in building and evaluating systems.

Practical requirements for Organisational Memories

Simultaneously, but separately, to our selves Abecker *et al.* (1998) have developed a number of practical requirements for Organisational Memories. They also champion the need for ‘computer-assisted knowledge capitalisation’ using assistant systems, similar in some ways to the Virtual Participant. They argue, with the use of Fredrick Brooks co-operation formula: IA > AI, that the combination of an assistant system with the user (intelligence-amplification) is much more powerful than any separate AI system. An argument we are in full agreement with.

Abecker *et al.* use their experience to specify a set of 5 requirements for the success of Organisational Memories:

1. Collection and systematic organisation of information from various sources
2. Minimisation of up-front knowledge engineering
3. Exploiting user feedback for maintenance and evolution
4. Integration into existing work environment
5. Active presentation of relevant information

These requirements have grown out of their experience in successful industrial practice. These guidelines, before we were even aware of them, have implicitly been followed in the development of the Virtual Participant. To take each point in turn we can see how the Virtual Participant has addressed these requirements, we can then go on to look at some of the specific aspects of the Virtual Participant's design and what we have learned from them, and how other systems have handled similar problems.

1. The course that the Virtual Participant is used on is typical of many distance education courses and involved a variety of materials. The students are provided with course texts, books, videos, audio cassettes, electronic conferencing, and tutorials. This makes up a wide variety of multi-media and multi-modal resources for the students, and for the Virtual Participant. Although the initial prototypes have only made use of the text based resources, which are already wide-ranging and varied, an advanced system would wish to make use other resources.

2. In the development of the virtual participant there was minimal, practically no, access to an expert. This forces upon us the need to minimise up-front knowledge engineering. In parallel to this is the need for the system to be rapidly applied to other courses which necessitates minimising expensive knowledge engineering effort.

3. The first prototype did not make any allowances for user feedback, however this was rectified in the second version. This feedback enables the system not only to identify whether it has matched information to the correct discussion, but also if that information is actually of any help. In addition to this, the monitoring of discussions enables the system to at least identify those discussions to which it has information and so can enhance that information, and identify

discussions about subjects which it has no knowledge, and hence store that knowledge for future re-use.

4. The Virtual Participant appears like any other user of the conferencing system. This removes the need for the users to run any specialist software, and minimises the amount of learning required to use the system. With the second prototype we were also able to use the same methods of formatting as normal users.

5. Relevant information needs to be presented at a time and in a context where it will be useful. This is the shortcomings of FAQ's where the user has to realise that the FAQ might contain the answer they were looking for, then they have to find the FAQ, and search that for the answer. The Virtual Participant contextualises its information to the current discussion, helping the students gain information that perhaps they did not even know they needed.

These five points cover key aspects which have helped the success of the Virtual Participant. The next section highlights four key dimensions in its design, which we have been forced to consider during our research. We have chosen to look directly at the Virtual Participant, because it exemplifies these design problems. We will then use these dimensions to refer back to our other example studies from earlier and related systems.

Lessons Learned in the Design of the Virtual Participant

This section presents an overview of four key design issues or 'lessons' which emerged as our experiment and evaluation progressed. These lessons provide four key social dimensions which have proved important in the acceptance and usability of the Virtual Participant. The four threads of development we have identified are (a) anthropomorphism versus mechanomorphism, (b) private versus public, (c) closed versus open, and (d) fixed versus extensible. Each of these sections presents the features we have changed and the situations which drove those changes and parallels in other systems.

Anthropomorphism versus mechanomorphism

A common problem with AI programs that interact with humans is that they must present themselves in a way that reflects their ability. Where there is a conflict between the ability of the system and the users perception of that ability a breakdown occurs and users may either fail to exploit its full potential or become frustrated with its shortcomings. We anthropomorphised our system with the hope of presenting it as something friendly and useful, but not fully human and with human abilities. Two other anthropomorphised systems can be seen in NewT and Maxims (Maes, 1994) which present visual representations of the agents' internal states. There is no clear evaluation of how helpful these representations were to the users. Koda (1996) found that there were large differences in users opinions on 'agents with faces'. In our own research we have found it necessary to alter the presentation of systems to improve their acceptance (Watt, 1997).

On the course we work with there is much use of metaphor and as the Virtual Participant needed a user name. At first we opted for the user name 'Uncle Bulgaria', a cartoon character from popular British children's television, *The Wombles*, a senior and knowledgeable figure amongst a group who recycle things. Although the majority of students were neutral to the name some really loved it, but others had strong negative feelings. One of the returned questionnaire comments puts it tactfully "the name had an adverse affect on what the objectives of UB were". Given the feedback we received and the specific dislike expressed we believe this is due, not the contents of the messages, but to the 'avuncular' presentation of the name.

In conjunction with the section on 'closed versus open' we 'mechanomorphised' the system with the name 'The Active Archive'. This has provoked no reactions and received favourable feedback, from which we infer that we have been successful. No attempt has been made to make it appear as anything more than an automated search engine. There is anecdotal evidence to suggest that younger students and children are more comfortable with anthropomorphic representations.

Private versus public

In the first prototype, and in contrast to Contact Finder (Krulwich & Burkey, 1996), all interactions with the students were public. We felt that by keeping these interactions public we would encourage group discussion and not be seen to be unfairly benefiting a sub-set of students.

In our original concept the Virtual Participant would function like any other user, with certain caveats: it only had access to certain conferences and it could only hold public conversations. The interaction model we used went like this: (a) the Virtual Participant tracks student discussions attempting to match them to its database, (b) when a match is found the root message is posted, (c) students can find out more information by asking pre-defined questions, (d) the system then posts the follow-up information to the conference.

Although in our first study over half the students said that they were comfortable with asking questions publicly, hardly any questions were asked. Additionally, despite specific instructions, a number of students posted the questions by private email. This was altered with the second prototype and although the Virtual Participant still accepts public questions the main method is by private email. Instead we have added the idea of ‘mass interest’, where any question asked more than five times is posted to the conference. The second study has seen many more questions asked, both in number and in range.

In comparison to this KSE registers submitted information to the users. In this way it is possible to track who has added what to the database. Although this aids in finding out who added a piece of information to the database, it perhaps discourages the addition of information a user might not wish to be credited with. Answer Garden makes all question-asking of experts anonymous, to deal with possible status problems for novices bothering experts. Yenta, a matchmaking system, is specifically designed to enable users to find people with similar interests while protecting their privacy (Foner, 1996). Finally (Resnick & Varian, 1997) provide a taxonomy of ‘recommender’ systems which highlights the public, anonymous, and pseudonymous nature of these systems.

Closed versus open

In the first study the students had no direct access to the Virtual Participant’s knowledge base. The feedback we received from the students prompted us to give this year access to everything in the knowledge base. This was presented to them as an archive conference containing a number of threads on different topics, the subject reflecting the contents.

This year we have asked students to give feedback on both the relevance of the message to the current discussion and the usefulness of the message to them. Work conducted in (Resnick & Varian, 1997) shows that there is a range of user types including some who are happy to give feedback with little or no prompting. Unfortunately, in the Virtual Participant, because there was no clear benefit to giving feedback, it was only given rarely in practice. Also, because we did not clearly acknowledge receipt of the feedback messages, some users felt that the messages were being discarded.

The core benefit of the Virtual Participant comes from contextualising the contents of the archive to the current discussion. Making the entire database available has been driven by user demand to have access to everything. In other systems there is a clear trade off between presenting identified information and allowing the users to browse everything. In Answer Garden the user answers questions to navigate to the appropriate part of the database where they are presented with an ‘answer’. If this answer does not help them, then the next stage is to ask an expert. However it may be the case that the user did not ask the right questions but the answer is in the neighbourhood. In this instance the user needs an option to, for example, ‘search for more documents like this one’. In both the Virtual Participant and Answer Garden the databases have initially been generated by experts with an application in mind. This has made them open in their own way, but the addition of information then becomes a problem — we will look this at in the next section. In the case of KSE everything is open and all information is available to all users, at the expense of the structure of Answer Garden and the Virtual Participant.

Fixed versus extensible

One problem is that the Virtual Participant’s knowledge base was, with some automated assistance, hand crafted. To be extensible to other domains this system must be simplified and made as automatic as possible. To be extensible within a domain the database must be kept up to date. With Answer Garden the same problem exists with the initial database having been hand crafted by experts. To the users of these systems the databases appear fixed, although the experts/tutors can extend them.

We believe that these are not two separate problems but rather two aspects of the same one. As before the problem can be reduced to one of benefit to those involved in the production and maintenance of the knowledge base. Those who benefit directly are the tutors/experts by reduced demands on their time. However they may not be motivated enough. In the case of the Virtual Participant it must become adaptive to the changing needs of the course, and identify what it does not know.

The approach of the Vicarious Learner (McKendree *et al.*, 1997) has been to make part of the student’s mark depend on their contributions to the group discussion. We could use this approach to get students to contribute to the case base. However we believe that this is a poor teaching approach and believe that if this becomes necessary then the evaluation model for the course should be changed.

Like Answer Garden, the way forward for the Virtual Participant is to find ways of motivating some section of user population to the maintenance of the knowledge base, encouraging a sharing culture. In the Virtual Participant we can do this by getting the students to reflect on what they have learned — adding these points to the database. With KSE, as with recommender systems, there is likely to be different user groups of contributors to the database risking a drop in usage from high contributors if they do not feel they are benefiting. However by encouraging the system to be used within groups of users who already collaborate anyway the risk of this should be reduced.

6. Pragmatic Issues: The Usability and Acceptance of Knowledge Management Systems

Up to this point we have looked at problems, solutions, studies, requirements, and lessons. It is now time to focus on something more practical, some guidelines for the successful introduction of computer-supported knowledge management systems. There are two key aspects to consider at this point. First is usability, *can* people use the system? Second is acceptability, *do* people use the system?

As far as usability is concerned, any knowledge management system needs to be successful at a straightforward usability level, and there is plenty of research in the field of human-computer interaction that can help achieve this (e.g. Norman, 1988). Acceptability is more frequently an issue for groupware systems, and Grudin’s (1994) paper contains a number of useful strategies and guidelines which may be of help here.

Agent-based tools introduce other issues. Norman (1997) and Erickson (1997) looks at the social issues raised by agent systems a bit more fully — almost all of these issues apply directly to any knowledge management system, even if it wasn’t conceived or intended as an agent system. This is because like agent systems, knowledge management systems tread on potentially sensitive human social relationships. Being aware of this affinity between agent systems and knowledge management systems can help to make the sensitivity of these social relationships more explicit. And designing agents to fit into existing social roles (like the oracle, the bard, and the village gossip), rather than trying to force the creation of new social roles, can be a powerful technique in the development of knowledge management systems.

The common problems of knowledge management systems, raised in sections 2 and 3, also bring out some issues relating to acceptability. Here, it is especially important to be aware of the individual and cultural issues. So, for example, computer-supported knowledge management systems need to allow people their privacy, to give people interaction shields, to be tolerant of the inevitable errors without causing social embarrassment, and to give people the feeling of personal space, freedom, and control.

The other side of agent systems — the often rather anthropomorphic character of the "agent metaphor" (Erickson, 1997) — is probably best avoided; as we've discussed earlier in this section, this change was probably one of the contributory factors to the success of the Virtual Participant (Masterton, 1998).

The level of anthropomorphism, then, is one dimension which can influence the acceptability of a knowledge management system, and with the other dimensions of privacy, visibility, and flexibility (discussed in section 5) forms part of a 'character space' for this kind of system. Careful placement of a system within this character space can, as the Virtual Participant showed, make the difference between a system's success and failure. There are other dimensions, too, which may also have an effect. Green and Petre's (1996) "Cognitive Dimensions Analysis" can be adapted to give us a model for this character space.

A Cognitive Dimensions Model for a Knowledge Management System's Character Space

Cognitive dimensions analysis (e.g. Green & Petre, 1996) works by placing systems on a number of (in principle) orthogonal dimensions. Although traditionally used for more 'cognitive' systems as argue that it has some useful properties for looking at usability and acceptance. Cognitive dimensions analysis is really a design tool; it helps a designer think about some of the factors and trade-offs which influence a system's usability and acceptability.

The following dimensions are probably the most helpful for assessing the character of knowledge management systems.

- **Closeness of mapping**; how close is the relationship between the system's interface and the system's domain?
- **Consistency**; when some of the interface has been learned, to what extent can the rest be inferred?
- **Diffuseness**; how much text, and how many symbols or icons are needed to express a meaning?
- **Error-proneness**; does the design of the system induce 'careless mistakes'?
- **Hard mental operations**; are there places where the system places a heavy cognitive burden on the user?
- **Hidden dependencies**; to what extent do things change behind the scenes, without the user being aware of the cause of those changes?
- **Premature commitment**; do users have to make decisions before they have the information they need?
- **Viscosity**; how much effort is needed to make a change to the system?
- **Visibility**; to what extent does the system show all the relevant information at any one time?

(adapted from Green & Petre, 1996). Of these, 'viscosity' more or less corresponds to the dimension of flexibility in the analysis of the Virtual Participant, and 'visibility' to the openness. To these, we can add the other dimensions of anthropomorphism and privacy, and it's also worth including the cultural compatibility, discussed in sections 2 and 3.

- **Anthropomorphism**; how anthropomorphic is the presentation of the system?
- **Privacy**; to what extent does the system respect the user's privacy?
- **Cultural fit**; does the system fit into its cultural context?

This gives us a more complete character space, which seems to address most of the important dimensions influencing the usability and the acceptability of knowledge management systems. One of Green and Petre's most important points is that these dimensions affect each other, and that trade-offs between them are both common and to be expected (between for example, visibility and privacy). In planning a system it is necessary to balance these dimensions for the role it is to take, for example, village gossips would be placed very differently to assistants on the dimension of privacy. The real power with this method is when you want to compare a family of possible systems — or when you want to look at the implications of design changes. Overall, it seems to offer a pretty good “broad brush” (Green & Petre, 1996) way of looking at both the usability and the acceptability of knowledge management systems, and a tool which the designer can use to map out the different roles that a system can adopt.

In practice there is more to getting a system accepted than planning it properly, and placing it appropriately in the character space. Even the way it is described in flyers, manuals, and other texts has a substantial effect on how users perceive the system. Perhaps you might wish to shy away from describing the system, at least to its users, as a knowledge management system. Avoiding using the term ‘agent’ to describe it may also be a good move. There are no clear-cut solutions to the problem of gaining acceptance within a real organisation (although Grudin, 1994, describes some of the common strategies used for organisational information systems, which have similar problems). But the rule is pretty simple, really, be sympathetic, and consider what the system’s intended users will think about its aims and intentions to help shape the story you tell.

7. A few conclusions

Successful use of knowledge management systems seems require a quite exceptional understanding of the special usability issues they raise. As computer-based systems, there are all the traditional cognitive, task-based, usability problems that are familiar to human-factors researchers. As groupware systems, knowledge management systems also face a whole raft of new and special problems (Grudin, 1994), because they also have to help people interact and collaborate with one another. And to cap it all, because knowledge management systems deal with valuable stuff, people’s feelings can run rather deeper in ways that often exacerbate these other problems.

Interface agents also introduce a host of usability issues that we haven’t yet really begun to address. The interesting point, though, is that by and large these are the same problems as knowledge management systems anyway. Agents raise the significance of social issues, such as privacy, flexibility, and an awareness of the social and cultural context; and by making these issues explicit, well-designed agent systems can alleviate these problems (e.g. Masterton, 1997, 1998). Evaluation

— and learning from evaluation — becomes especially important, however, given the unreliability of people's intuitions when designing and developing any collaborative system (Grudin, 1994).

Perhaps the most important issue raised by agents is that of the different social roles that knowledge management systems can adopt. By looking more closely at some of the existing systems, the parallel between human cultural knowledge transfer practices and those being introduced by knowledge management systems becomes much clearer. Intelligent agents can help because they make it possible to build systems which adopt these roles without putting the additional work burden onto people who are already busy enough (Watt, 1993; Grudin, 1994). Knowledge management systems can, and genuinely do, play the roles of oracles, bards, and village gossips within today's modern organisations. And why shouldn't they?

Acknowledgements

We wish to thank Simon Buckingham Shum, Trevor Collins, Chris McKillop, Paul Mulholland, and Tamara Sumner of KMi, The Open University, and Nader Arzami, John Davies, Roger Martin, Divine Ndumu, and Charanjit Sidhu of British Telecom Laboratories at Martlesham, UK, for helping us to form and discuss the ideas in this paper. SM is funded by EPSRC studentship No. 95306409 and a CASE Award from British Telecom Laboratories. SW's work has been greatly helped by a Short-term Research Fellowship in the Intelligent Systems Research Group at British Telecom Laboratories. We are very grateful to BT for their support of this research.

References

ABECKER, A., BERNARDI, A., HINKELMANN, K., KÜHN, O., & SINTEK, M. (1998). Toward a Technology for Organizational Memories. *IEEE Intelligent Systems and their Applications*, **13**(3), 40-48.

ACKERMAN, M. S. (1994). Answer Garden: A Tool for Growing Organizational Memory. Unpublished PhD thesis, Sloan School of Management, Massachusetts Institute of Technology. February, 1994.

BUCKINGHAM SHUM, S. (1997). Negotiating the Construction and Reconstruction of Organisational Memories. *Journal of Universal Computer Science* (Special Issue on IT for Knowledge Management), **3**(8), 899-928.

CONKLIN, J. (1996). Designing Organizational Memory: Preserving Intellectual Assets in a Knowledge Economy. Group Decision Support Systems Inc., available at http://www.gdss.com/DOM.htm.

DAVIES, N. J., STEWART, R. S, & WEEKS, R. (1998). Knowledge Sharing Agents Over the World Wide Web. *British Telecom Technology Journal*, **16**(3), 104-109.

DOMINGUE, J., & SCOTT, P. (1998). KMi Planet: A Web Based News Server. In the Proceedings of the Asia Pacific Computer Human Interaction Conference (APCHI'98), Shonan Village Center, Hayama-machi, Kanagawa, Japan, 15-17 July, 1998.

ERICKSON, T. (1997). Designing Agents as if People Mattered (pp. 79-96). In J. Bradshaw, ed., *Software Agents*. AAAI Press / MIT Press.

FISCHER, G., LEMKE, A. C., MASTAGLIO, T. (1990). Using Critics to Empower Users. In the Proceedings of CHI'90, Seattle, Washington, pp. 337-347.

FONER, L. (1996) A Security Architecture for Multi-Agent Matchmaking, presented at The Second International Conference on Multi-Agent Systems, Keihanna Plaza, Kansai Science City, Japan, December 1996.

FONER, L., & CRABTREE, I. B. (1996). Multi-agent Matchmaking. *British Telecom Technology Journal*, **14**(4), 115-123.

GOFFMAN, E. (1956). *The Presentation of Self in Everyday Life.* New York: Doubleday.

GOFFMAN, E. (1963). *Behavior in Public Places.* New York: Free Press.

GREEN, T. R. G., & PETRE, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages and Computing*, **7**, 131-174.

GRUDIN, J. (1994). Groupware and Social Dynamics: Eight Challenges for Developers. *Communications of the ACM*, **37**(1), 92-105.

KODA, T. (1996). Agents with Faces: A Study on the Effects of Personification of Software Agents. Masters Thesis, MIT.

KONSTAN, J. A., MILLER, B. N., MALTZ, D., HERLOCKER, J. L., GORDON, L. R., & RIEDL, J. (1997). GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, **40**(3), March 1997, 77-87.

KRULWICH, B., & BURKEY, C. (1996). The ContactFinder Agent: Answering Bulletin Board Questions with Referrals. In the Proceedings of AAAI'96. AAAI Press.

KRULWICH, B., & BURKEY, C. (1997). The InfoFinder Agent: Learning User Interests through Heuristic Phrase Extraction. *IEEE Intelligent Systems and their Applications*, **12**(5), September 1997, 22-27.

LEWIS, C., & RIEMAN, J. (1994). *Task-Centered User Interface Design: A Practical Introduction*. Shareware book, available at ftp://ftp.cs.colorado.edu/pub/distribs/clewis/HCI-Design-Book/.

MAES, P. (1994). Agents that Reduce Work and Information Overload. *Communications of the ACM*, **37**(7), July 1994.

MASTERTON, S.J. (1997). The Virtual Participant: Lessons to be Learned from a Case-Based Tutor's Assistant. In Proceedings of the Conference on Computer Supported Collaborative Learning, Toronto, Canada, 10-14th December 1997.

MASTERTON, S. J. (1998, to appear). Computer Support for Learners Using Intelligent Educational Agents: The Way Forward. To appear in the Proceedings of the Sixth International Conference on Computers in Education, ICCE'98. Beijing, China, October 1998.

MCKENDREE, J., STENNING, K., MAYES, T., LEE, J., & COX, R. (1997). Why observing a dialogue may benefit learning: The Vicarious Learner. In the Proceedings of PEG'97, May 1997, Sozopol, Bulgaria.

NASS, C., STEUER, J., & TAUBER, E. R. (1994). Computers are Social Actors. In the Proceedings of CHI'94.

NORMAN, D. A. (1988). *The Psychology of Everyday Things*. New York: Basic Books.

NORMAN, D. A. (1997). How People Might Interact with Agents. In J. Bradshaw, ed. *Software Agents*. pp. 49-55. AAAI Press / MIT Press.

O'LEARY, D. E. (1998). Knowledge-Management Systems: Converting and Connecting. *IEEE Intelligent Systems and their Applications*, **13**(3), 30-33.

RESNICK, P., & VARIAN, H. R. (1997). Recommender Systems. *Communications of the ACM*, **40**(3), March 1997, 56-58.

RHODES, B. J., & STARNER, T. (1996). *Remembrance Agent: A Continuously Running Automated Information Retrieval System*. Paper presented at the First International Conference on The Practical Application Of Intelligent Agents and Multi Agent Technology (PAAM'96).

STEFIK, M. (1986). The Next Knowledge Medium. *AI Magazine*. **7**(1), 34-46.

WATT, S. N. K. (1993). Role Conflict in Groupware. In the Proceedings of the First International Conference on Intelligent and Cooperative Information Systems (ICICIS'93).

WATT, S. N. K. (1997). Artificial Societies and Psychological Agents. *British Telecom Technology Journal*, **14**(4), 89-97.

WATT, S. N. K. (1998). Psychological Agents and the New Web Media. In M. Eisenstadt & A. T. Vincent, eds., *The Knowledge Web*. Kogan Page.

YATES, J. & ORLIKOWSKI, W. J. (1992). "Genres of Organizational Communication: A Structurational Approach to Studying Communication and Media." *Academy of Management Review*, **17**(2), 299-326.

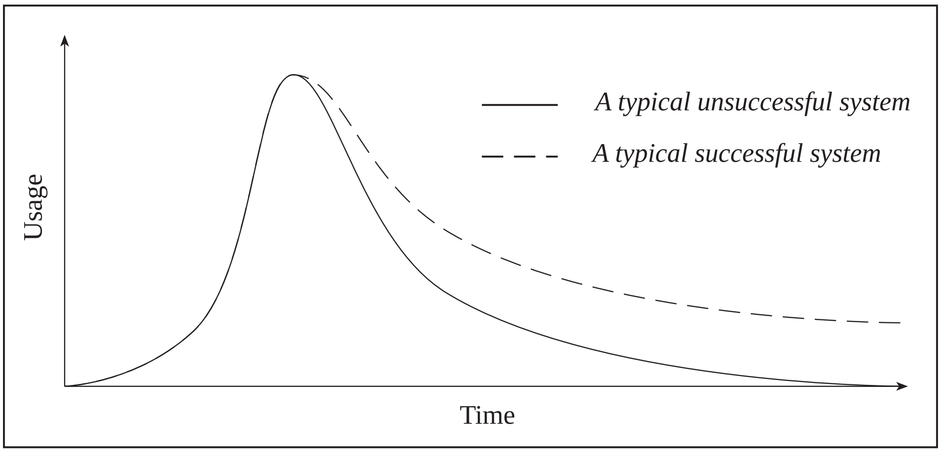

Figure 1. Typical usage curves for new unsuccessful and successful knowledge management systems.