ELSEVIER

Interacting with Computers 14 (2002) 663–688

Interacting with Computers

www.elsevier.com/locate/intcom

# Living memory: agent-based information management for connected local communities

Kostas Stathis[a,*], Oscar de Bruijn[b], Silvio Macedo[b]

[a]*Department of Computing, School of Informatics, City University, Northampton Square, London EC1V 0HB, UK*
[b]*Department of Electrical and Electronic Engineering, Imperial College of Science, Technology and Medicine, London SW7 2BT, UK*

**Abstract**

We investigate the application of multi-agent systems to develop intelligent information interfaces for *connected communities*, a class of computer applications aimed at enhancing the way people interact and socialise in geographically co-located communities such as neighbourhoods. In this context, we study the problem of providing effective information management in support of social interaction when a diverse range of computing devices is employed. The novelty of our approach is based on combining innovative interactive devices with a framework based on agent roles in order to support the effective flow of community-related content for the people of a given locality. In particular, we have integrated existing techniques for information retrieval and filtering with measures of content popularity, to ensure that documents in the community system are optimally available. After reporting on the potential presence of the system in the community, we report on the development of a framework for multi-agent systems in which agents provide a number of services aimed at facilitating personalised and location-dependent information access to members of the community. We also present a summary of the results of an expert evaluation of the information flow resulting from the communication between agents, and a user-evaluation of the information dissemination facilities provided by the system. © 2002 Elsevier Science B.V. All rights reserved.

*Keywords:* Connected communities; Interaction design; Software agents; Information dissemination; User profiles; Acquaintance models

## 1. Introduction

*Connected communities* (Noack, 1994) describe a class of applications that aims to augment, and possibly improve, the way people interact and socialise in

[*] Corresponding author.

*E-mail addresses:* k.stathis@city.ac.uk (K. Stathis), o.debruijn@ic.ac.uk (O. de Bruijn), s.macedo@ic.ac.uk (S. Macedo).

PII: S0953-5438(02)00014-0

community environments such as neighbourhoods. In this paper we describe our work in Living memory (LiMe)—a long-term research and development project funded by the European Commission with the aim of creating a connected community. In LiMe, 'being connected' is interpreted as involving communication and sharing information, while 'being part of a community' assumes a shared collective memory. Unlike current virtual communities formed by people who are often separated by both distance and culture, LiMe aims to help people who share a locality (e.g. an urban neighbourhood) to interpret and preserve the richness and complexity of their *local* culture. Thus, the LiMe community is defined by a shared geographic location, rather than or in addition to the common practices and interests that often define virtual communities.

Communities exist by the grace of a critical mass of communal social content, as well as the appropriate network for its dissemination, which work together as catalysts for community interactions. Communal content can take many forms, varying from a postcard with a message to an exchange of emails. In the case of local communities, increasing the accessibility of local communal content can contribute to an enhancement of social cohesion that may have been fragmented by the recent pace of social and technological change. Local communal content may include items of information such as local news published in the local gazette, but also the flyers commonly posted on lampposts and boarding. The approach taken in LiMe, therefore, is to empower ordinary people by allowing them to interact with a range of different devices with easy-to-use human interfaces in the creation, assessment and annotation of local communal content, such as announcements of events, local history and personal anecdotes.

The LiMe project has explored a range of concepts and interactive devices (Norman, 1998), such as interactive bus stops, informative park benches and large-screen public displays, and their contexts of use (Matthews et al., 1999), for their role in supporting communication based on communal content. These devices can all be connected together to form a distributed network of devices able to communicate information from a user to another, when and where it is needed. Successful communication of communal content therefore depends, to a large extent, on the appropriate communication between the devices that form part of such a multi-device system. In addition, devices need to adapt to users' needs and the particular social settings in which they are used in order to optimise the benefit people gain from using these devices. Moreover, as some social settings would give rise to some quite complex interactions between devices, the interfaces provided must be able to hide this complexity so that ordinary people can use them.

In order to address the problems of making a device intelligent enough so that it can adapt to a user's needs, hide the complexity of the interactions from the device's interface, and manage the complexity of the information that flows between devices, we advocate the application of software agents and multi-agent systems technology (Mamdani et al., 1999). In this context, agents are software entities that have a certain degree of autonomy and intelligence and have the capacity to relieve the members of the connected community from having to explicitly make connections and associations with content, specific locations and fellow community members (Maes, 1994). Agents for connected community systems such as LiMe support interactions on two distinct levels:

- the *personal level*—to allow the community system to present itself differently to each user, and to relate users to both local information and places in the community;
- the *community level*—to support the creation of acquaintance groups, the pro-active emergence of voluntary roles, the flexible execution of accepted, non-intrusive procedures, and the integration of content created from multiple disparate sources.

The concept of an individual agent leads progressively to the concept of multi-agent systems (Jennings et al., 1998). A multi-agent system is a complex software system, which is built from many agents interacting with other agents as well as with humans. Each agent in a multi-agent system has incomplete information due to a limited viewpoint on its operational environment. There is, therefore, no global system control. Information is often decentralised, and computation is asynchronous. In other words, agents work in virtual islands of information, able to communicate only with other agents and humans that are within their information horizon. It is impossible, therefore, for individual agents to predict the consequences of their actions on the behaviour of the overall multi-agent system. Hence each agent must be given a set of mechanisms that define how it should behave in relation to the desired behaviour of the system. Thus, it is possible to design a multi-agent system for a particular purpose, despite a lack of centralised control and knowledge.

As a convenient basis for creating the mechanisms that agents use to decide how to behave appropriately in particular circumstances, each agent is designed to play a particular *role* within the multi-agent system (Skarmeas, 1995). Suitable specification of roles for agents running on electronic devices could make such devices aware of other devices, agents and people within an electronically connected community, and facilitate the creation of interfaces that are persistent, active and adaptable to the highly dynamic requirements of interactivity, particularly in social settings. In this context, we identify what kind of agent roles are required in order to support community interactions at both the individual and community levels.

Using roles to build agents facilitates the development of a multi-agent system as a community of agents whose behaviour reflects the contexts in which community members interact and communicate. Often communication between members of a local community occurs asynchronously and between people who are in different locations within the community. However, one of the characteristics of local communities is that community activity also takes place synchronously, often by chance, in diverse locations. People quite simply meet up in the pub and bump into one another while shopping at the local mall, or they leave messages, such as ‘for sale’ notices, in particular locations directed to anybody interested who might be passing by. In LiMe we are dealing with the unique situation, therefore, in which the physical locality occupied by the community can be used in the enhancement of social interaction.

The multi-agent system presented here is an extension of our original work with agents for connected communities (Mamdani et al., 1999), seeking to investigate the kind of roles the agents play for providing information dissemination in local communities. Other authors have advocated agent technology with the aim of enhancing social interaction between members of geographically co-located communities. For example, Sumi and Mase (2002a,b) have used agents for the purpose of user profiling and information retrieval as part of a system aimed at knowledge sharing between visitors of an academic conference. Agents have also been employed to provide collaborative filtering of community-related information (Koch, 1998) for the purpose of increasing community awareness among the people inhabiting the Venetian suburb of Campiello (Agostini et al., 2002).

An important characteristic of the LiMe devices and agents is that their design is based on a multi-disciplinary effort with contributions from ethnographers, interaction designers and computer scientists. The ethnographers conducted a study of the prevalent interaction habits of the members of the LiMe target community, a leafy suburb of Edinburgh in Scotland (Davenport et al., 1998). The particular aim of that study was to identify the characteristics of places, such as the local café, the pub, library and medical centre, in terms of the nature of community interactions taking place there. Then, the interaction designers and computer scientists used the results of that study as the basis for the design of the LiMe multi-agent system.

In summary, we are concerned with the use of intelligent information interfaces to enhance social interactions between members of a local community. For this purpose, we aim to exploit the co-located nature of local communities by embedding interactive devices within locations that have a history of supporting community interactions. Allowing these devices to work in combination with mobile devices carried by community members empowers ordinary people to access and contribute to communal content. We further invest these devices with software agents designed to provide services aimed at enhancing the accessibility and visibility of context-sensitive communal content.

We structure the presentation of the paper as follows. In Section 2, we briefly describe some of the device concepts for agent-based information provision that have been investigated and prototyped as part of the LiMe project. Each of these devices is host to one or more agents whose roles reflect the contexts in which these devices are used. In Section 3, we describe these LiMe agents and their roles in the multi-agent system. This is followed by a description in Section 4 of how communal content is managed through the communication between these agents. In Section 5 we describe how the system was validated in order to assess the impact of the LiMe multi-agent system on the quality of information provision to members of the project's target community. We conclude this paper in Section 6 with a discussion of the contribution of our work to the design of connected communities and multi-agent systems, and provide directions for future work.

## 2. The presence of LiMe in the community

In order to encourage community interactions, a number of devices were designed to give LiMe a ubiquitous presence within the environment of the community. These devices are the vehicles through which the LiMe agents operate on communal content in order to manage the flow of information within the community. The devices have been designed along two orthogonal dimensions, distinguishing between static and mobile devices on the one hand and between personal and shared devices on the other. Thus, some devices are static and personal, such as PCs located in the homes of many community members, whereas others are static and shared, such as interactive tables and large displays in public

Table 1
Examples of the four different types of devices resulting from the orthogonal combinations of the static/mobile and personal/shared dimensions

| | Personal | Shared |
|---|---|---|
| Static | Home PC | Table |
| Mobile | Mobile Phone | Token |

spaces. Similarly, some devices are mobile and personal, such as mobile phones and personal digital assistants (pocket PCs), while others are mobile and shared, such as the tokens created to allow information to be easily exchanged when members of the community meet (Table 1). Some of these devices have been built and were briefly installed within the LiMe target community for evaluation. The nature of these devices is described next.

### 2.1. Static devices

The appearance and characteristics of static devices are determined by their location in the community. In private locations, such as peoples' homes, we find personal static devices such as PCs or Internet enabled television sets, with personal accounts for each person having access to the device. These devices are used to access all kinds of content, some of which may be of a private nature. This information includes the owner's personal history—a record of the owner's interactions with people and devices—and personal profile—a record of the owner's interests, needs and acquaintances.

Shared static devices, on the other hand, are typically found in places that form natural crossing and meeting points in the community, such as local cafés and pubs, bus stops, train stations and shopping malls. These devices may be embedded into the physical fabric of these locations such that communal content can be created and accessed naturally. User interfaces to these embedded devices are typically informal and easily accessible, blending with the immediate physical local environment. Content available on these devices is strictly public in nature. A useful example of an information device, albeit not an interactive one, is the large plasma display at Victoria train station in London displaying news headlines for people waiting on the concourse (ITN, 2002). Other public embedded devices concepts, such as interactive park benches (Gaver and Martin, 2000) have been explored. Some of the LiMe concepts for shared static devices are discussed briefly below.

#### 2.1.1. The coffee corner

The coffee corner is a metaphor for a public place, where people meet informally and spend some time in social interaction. For these places we have developed an interactive table that gives people the opportunity to interact with communal content in a semi-private setting (Fig. 1). A coffee corner is made up of a collection of these tables, each displaying communal content, such as announcements of lost cats on a screen that forms part of the surface of the table. The LiMe table is an example of the kind of static device that facilitates information dissemination in a way that is unobtrusive, but also very accessible. Sometimes, useful information can be discovered serendipitously while the people sitting

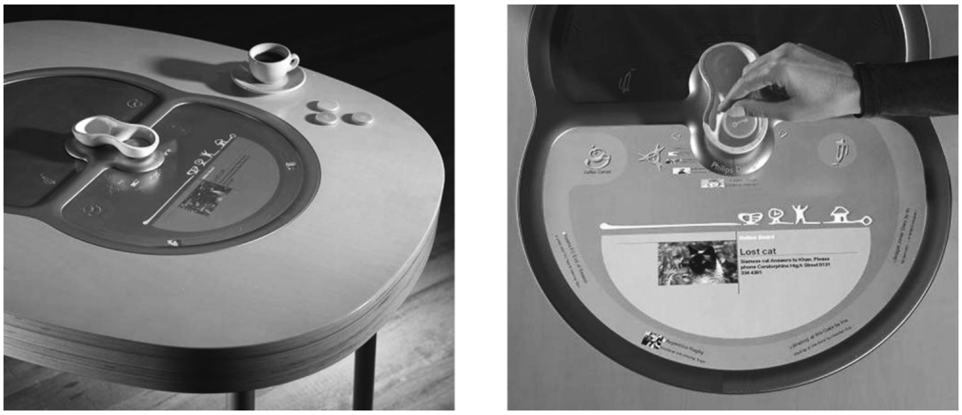

Fig. 1. The LiMe table and three of the LiMe tokens (left), and an example of the use of a token (Section 2.2.1) to capture content displayed on the LiMe table (right; images provided courtesy of Philips Design Eindhoven, the Netherlands).

around the table are engaged in social discourse (de Bruijn and Spence, 2001). At other times, the LiMe table may become the focus of social interaction between the people sitting around it.

### *2.1.2. The bus stop*

A bus stop is a metaphor for a public place where people spend a variable (but often brief) period of time in which they are often not engaged in any particular activity. A bus stop is an obvious place to display information, as witnessed by the advertising already present in these locations (Fig. 2). During the time spent at these locations people may want to interact in some way with an embedded information display, even though this interaction may necessarily be limited to simple operations such as the capture of content on an information token or mobile phone as described in Section 2.2.

## *2.2. Mobile devices*

Mobile devices are typically carried around by the members of the community, who use these devices to interact with their environment and to capture communal content. Interacting with the environment means that mobile devices interact both with other mobile devices and with the embedded static devices (cf. Sumi and Mase, 2002b). Personal mobile devices are owned and used privately by a single person to interact with information, which is often of a private nature and not unlike the information stored on personal static devices. Personal mobile devices are already in common use and include mobile phones, personal digital assistants (also widely known as PDAs), and pocket PCs. Shared mobile devices such as the tokens described later, on the other hand, allow access to information that is not private and which people may therefore want to share with others by handing these devices over. In general, mobile devices can be used to capture information that is encountered on static devices in order to make it permanently accessible, even when one is no longer in the location at which the information was

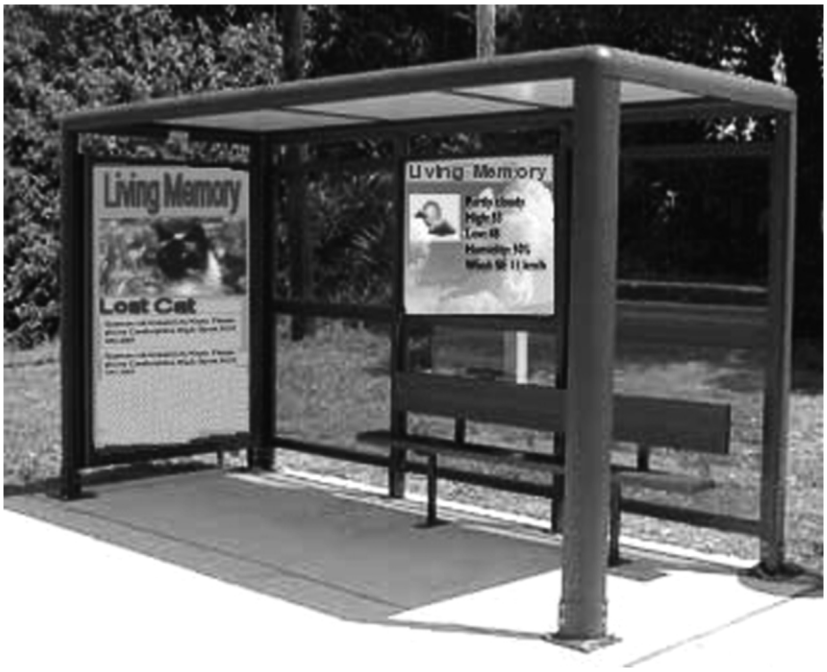

Fig. 2. Information displays embedded in a bus stop. The smaller display may present information about bus times and routes that people may browse interactively. The larger display may present information about the community relevant to both the location of the bus stop (for people coming off the bus) and the locations along the bus routes (for people leaving on the bus). As people browse information about bus routes on the smaller display, information about locations along these routes may appear on the larger display.

encountered. In particular, shared mobile devices provide a way of capturing and physically sharing communal content by giving these devices to other people.

### *2.2.1. Tokens*

A token is a physical electronic device that serves as a container for information. It is a small, coin-sized artefact that enables specific pieces of content to be 'bookmarked', after which they can be accessed directly and efficiently in any location where there is a static LiMe device available (Fig. 1).

## 3. LiMe agents

In Section 2 we described how people interact with communal content using static and mobile devices. Now we discuss how to adapt these devices for different circumstances through the deployment of software agents. We organise agent functionality through the notion of the roles (Heugeneden, 1993; Wooldridge et al., 2000) that agents play on behalf of people (Pitt et al., 2001) in order to manage dissemination of communal content.

Our model, shown in Fig. 3, identifies three different types of agent, which inhabit

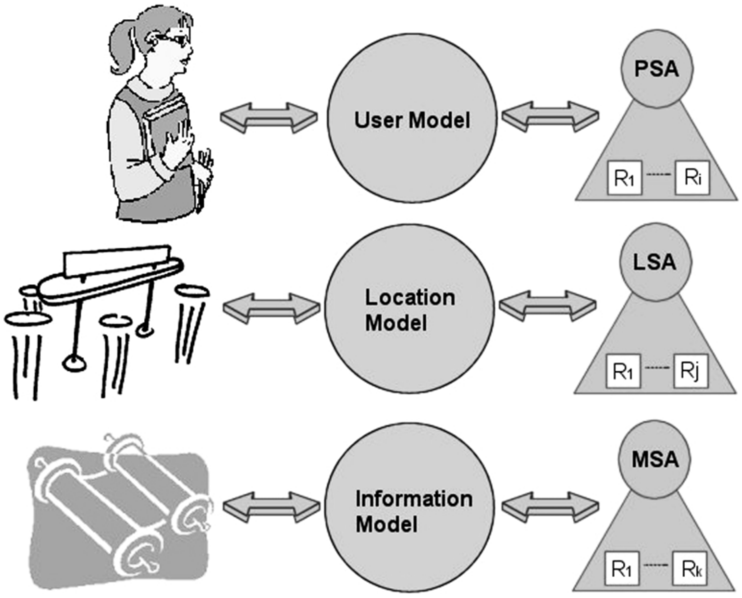

Fig. 3. Interactions of a particular user (community member) with a personal device are modelled by a personal service agent in the creation of a user model; interactions of users in a shared device at a particular location are modelled by an location service agent in the creation of a location model; and new communal content is modelled by a memory service agent in the creation of an information model. R1,…, R*i*,…, R*j*,…, R*k* refer to roles that the agents play in the community system.

static, mobile, personal and shared devices. The first type is a personal service agent (PSA) whose purpose is to model interactions of community members with personal devices, the second type is a location service agent (LSA) whose purpose is to model interactions of community members with shared devices placed in specific community locations, and the third type is a memory service agent (MSA) whose purpose is to model how the information content is stored and disseminated to a person or location. Models of people, locations and information are either described and updated through explicit interaction with devices, or deduced implicitly by agents from the content of these interactions. Next we describe the repertoire of roles that the three types of service agent need to posses as well as the corresponding service models.

### *3.1. Personal service agents*

Personal devices are inhabited by one or more PSAs that are designed to manage the storage of community information and its access for an individual member based on a model of the device's user.[1] We build upon the framework described by Mamdani et al., (1999) who allow either the user to update parts of the user model explicitly via the device interface or the PSA to infer parts of the model implicitly according to what the user has

[1] Personal service agents act on behalf of people by relying on the ownership model proposed by Pitt et al. (2001).

done. In this context, we identify three main roles for a PSA whose basic functionality is outlined briefly next.

### *3.1.1. Personal profile manager*

In the role of a personal profile (Mamdani et al., 1999) manager, the PSA manages information including personal details, preferences, interests and needs of an individual community member. A person's personal details are data that identify a person, such as their name, address, age, and gender, from which the PSA may draw conclusions based on stereotypes (Kobsa and Wahlster, 1989). A person's preferences are expressed as criteria and constraints that govern the interactions of the user with the device. For example, a PSA inhabiting a mobile phone may use knowledge about the times at which a person does not want to be disturbed to automatically divert calls to a voice-mail service. A person's interests are specified by a set of keywords corresponding to a person's likes and dislikes. For example, the PSA may use knowledge about a person's like for 'tennis' and dislike for 'badminton' to distinguish between relevant and irrelevant content. A person's needs define the goals that a PSA is required to pursue on behalf of its owner and indicate what the benefits are that a person expects to receive from using a community based information system. For example, a person might expect to establish new links to other members of the community, or receive recommendations about community services.

### *3.1.2. Acquaintance model manager*

In the role of acquaintance model (Wittig, 1992) manager, the PSA maintains a record of the social circle of a community member. Such a circle is described in terms of the community members known to an individual as well as the activities that a member pursues in different community locations (Mamdani et al., 1999). For example, 'playing chess in the Duke of Edinburgh pub with Arthur' is the kind of information that the acquaintance model manager maintains. In that role, the PSA is also responsible for updating the connections in the social circle. For example, the more you play chess with Arthur in the Duke of Edinburgh pub, the stronger the connections with chess, Arthur, and the pub will become.

### *3.1.3. Personal history manager*

In the role of a personal history manager, a PSA logs the user's interactions with communal content while using the personal device that the PSA inhabits. The PSA uses this log to deduce the emergence of new interests and acquaintances that arise from the interactions of the user with communal content. In addition, the PSA can use the log to speculate about events that take place in the community. For example, consider the situation in which a person A sends an announcement of a chess tournament to person B, which sets off an exchange of messages related to the chess tournament. If person B was not previously part of the acquaintance model of person A, then the PSA in the role of acquaintance model manager may infer that person B is a new acquaintance and should be added to the model. Similarly, if chess did not previously appear in the interest profile of person A, then the PSA in the role of personal profile manager may infer that chess represents a new interest and add this interest to the interest profile. Thus, the main responsibility of the PSA in this role is to keep a time frame specifying which part of the log remains uninterpreted, enabling the PSA to discriminate between inferences about what has happened, what is currently happening, and what might happen in the future.

### 3.2. Location service agents

As a consequence of being tied to a particular location shared static devices provide a view of the community that is characteristic of their location. In order to provide such location sensitivity, shared static devices are home to one or more LSAs that manage models of the locations in which these devices are situated (Fig. 3). To allow an LSA to construct a location model, we re-adjust the user model used to cater for locations. In this context, we identify three main roles, which we describe briefly below.

#### 3.2.1. Location profile manager

In the role of location profile manager, the agent manages location information, including location details, location topics and needs of the community members while using the device. Location details include data that identify a location, such as its name, address, and function, from which the agent may draw conclusions about the type of people that are likely to be using the static devices in that location. Location topics are specified by set of keywords corresponding to the likes and dislikes of the group of people interacting with the device. For example, the LSA of an interactive bus stop may infer a liking for 'news', while the LSA of a LiMe table situated in a pub may infer a liking of 'sport'. Similar to how PSAs use personal profiles, LSAs use these preferences to be able to distinguish between relevant and irrelevant content before the static device presents it. Needs express the goals that an agent is required to pursue on behalf of the community system-administrator of the location, and indicate what the benefits are, that a group of people might receive by using the community information system. The goal of the agent in this context is to enhance the kind of activities that people pursue while they are in a particular location. For example, a person waiting at a bus stop might like advice for alternative routes to a destination if the usual service is, for some reason, not available.

#### 3.2.2. Location neighbourhood manager

In the role of location neighbourhood manager, the agent maintains information describing other shared and static devices that are in the vicinity of the location. Such information is not simply about which other devices are located nearby, but also the kind of services these devices support. In this way, the agent may advise a person who is at a bus stop as to the closest news kiosk that may be located in the local pub. This role also allows the LSA to update its knowledge of the neighbourhood, for example when a new static shared device is introduced in the vicinity.

#### 3.2.3. Location history manager

In the role of a location history manager, an LSA logs users' interactions with communal content while using the shared device that the LSA inhabits. The LSA uses this log to make inferences about the profile of its users and its neighbourhood model. Similar to a PSA's role of personal history manager, the role of location history manager involves keeping a time frame specifying which part of the log remains uninterpreted, enabling the

LSA to discriminate between inferences about what has happened, what is currently happening, and what might happen in the future.

### *3.3. Memory service agents*

Typically, members of a connected community use the system in order to contribute content that may be of interest to others or to access existing content that is of interest to them. In LiMe we understand content as a set of multimedia documents, which following ethnographic input, we named electronic postcards. Electronic postcard consist of a textual part, a picture part, an audio part, a video part, and other information such as the name of the sender and a set of keywords describing the non-textual content (Pitt et al., 1998). However, for information management purposes, in this paper we focus only on the textual part combined with the keywords that describe the non-textual components.

LiMe documents, together with records of their history, form part of what is referred to as a collective memory of the community. Such documents have also their own life cycle, which starts at the time such a document is created and submitted to the system. For each document a record is made of when members of the community access and annotate it. The process of creation and subsequent annotation creates a history from which we can derive additional properties such as how popular a piece of information is, or how a piece of information relates to other pieces of information at any time.

Representing and managing community information (or collective memory) is a very complex task. To address this issue we use MSAs that are designed to produce an information model based on the four roles that we describe in the sections below.

#### *3.3.1. Information disseminator*

In the role of information disseminator, the MSA distributes communal content to the PSAs and LSAs registered with the services provided by the MSA. We further divide the information dissemination role into the following sub-roles:

- *retriever*—in this sub-role the MSA responds to additions to the personal interests of people and locations by retrieving and passing on content that matches those interests;
- *filterer*—in this sub-role the MSA responds to content that is newly registered with the MSA and, assuming that this content is suitably indexed, determines which PSAs and LSAs should be informed about this content.

A more detailed discussion of information dissemination in LiMe is provided later in the paper, in Section 4.1.

#### *3.3.2. Relevance monitor*

In the role of relevance monitor, the MSA checks and updates the relevance records of documents according to the popularity measures used in the system (discussed in detail in Section 4.2). One of the main tasks of the relevance monitor is to communicate with other MSAs to inform them about which documents have low relevance and need to be archived and eventually deleted.

### 3.3.3. Archivist

In the role of archivist, the MSA stores content in different storage devices (Chervenak et al., 1998), with the main criterion being how relevant a document is to the interests of the people in the community. Less relevant documents are put in less accessible storage devices, while more relevant documents are put either in a more accessible device or cached locally if the demand for these documents is very high.

### 3.3.4. Garbage collector

In the role of garbage collector, the MSA decides whether to remove content from the system's memory. Removal here is not like garbage collection in programming languages (Henriksson, 1997), but depends mainly on whether a document that has been made public has a time at which it expires. If so, a document can be deleted completely from the memory, only if certain additional conditions are met, as we shall see in Section 4.2, after which it will be accessible in archived versions of a particular memory instance.

### 3.3.5. Link manager

In the role of link manager, the MSA maintains the links across documents. The important aspect of this role is to keep the integrity of the information content currently in the system intact (Toni 2001), which is essential in case content is removed or made inaccessible by its owner.

## 4. Information management based on agents

The provision of communal content has to reflect what is relevant and of interest to the members of the community at any one time and in any community context. That is, the LiMe system needs to incorporate mechanisms for information management that allow it to provide content where and when it is most relevant. Therefore, we have based information management in LiMe firstly on a process that matches communal content with personal and location profiles and, secondly, on appropriate management of communal memory by taking into account the popularity and relevance of particular topics in the community. Each of these two mechanisms is described next followed by a discussion of related work in agent-based information management from the literature.

### 4.1. Information dissemination

Information dissemination in LiMe is based on the *vector space* method, which is well understood in terms of its advantages and disadvantages for matching documents and profiles (Belkin and Croft, 1987; Salton and McGill, 1983). We combine the vector space method with the measures of popularity, as discussed in Section 4.1.1, in order to add to what is basically a mechanical matching method a measure that relates relevance and popularity of documents to members of the community. Although the vector space model is not new, the novelty of our approach is based on integrating the vector space method with document popularity and implementing the resulting functionality in the information disseminator role played by MSAs. In this section, we specify how we conceptually

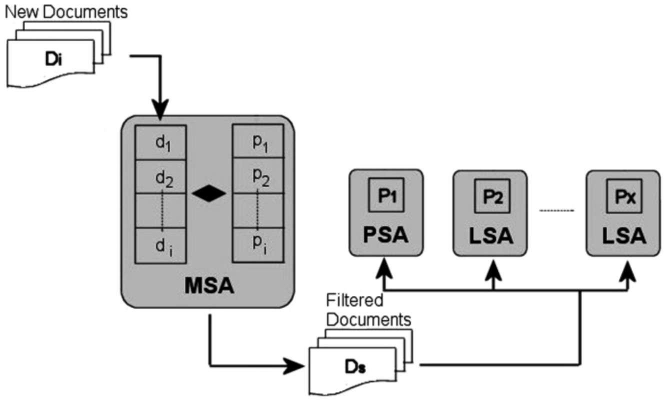

Fig. 4. Schematic representation of the communication between an MSA and one or more PSAs and LSAs during information filtering. For simplicity of presentation we show only one document source. In practice, however, the agent may receive documents from several different sources. On arrival, a new document ($D_i$) is indexed and stored in the indexed document list ($d_1$, $d_2$,…, $d_i$). Then, the indexed document is matched against the stored profiles list ($p_1$, $p_2$,…, $p_i$) to produce a set of user profiles (P1, P2,…, P$x$) and agent identifiers that the new document ($D_i$) should be sent to. $D_s$ is the list of documents that are filtered by the MSA for a given time-period.

organise this role's sub-roles to make MSAs disseminate information either due to arrival of new documents or due to changes in user profiles.

### *4.1.1. Information filtering*

In the sub-role of information-filterer, an MSA is responsible for responding to the arrival of new documents. In our multi-agent architecture, a new document arrives when a link that identifies the document becomes public in one of the LiMe devices (such as a static or mobile device registered with the system). In this context, the activity of filtering consists of repeatedly taking incoming documents from different nodes, comparing these documents with all of the profiles currently held by the information dissemination system, and distributing them to those recipients whose profiles match with these documents.

The break down of tasks required to control the behaviour of the information-filtering sub-role is as follows:

1. Input a document $D_i$;
2. Standardise $D_i$ to produce $d_i$ and store $d_i$;
3. Match $d_i$ against all of the stored profiles to produce a list of profiles that match $d_i$;
4. Send $d_i$ to all the recipients mentioned in the list of matching profiles.

The execution of these tasks is done in a cyclic manner that is driven by new documents. The general picture of information filtering is shown in Fig. 4. This organisation of the interaction assumes that the MSA in the information-filtering role communicates with both the PSAs and LSAs. One of the advantages of such standardised communication between roles is that it supports the scaling up of the filtering service, in this case by having the agent spawning several information-filtering roles each accepting documents from a different source, but still comparing them to the same list of profiles.

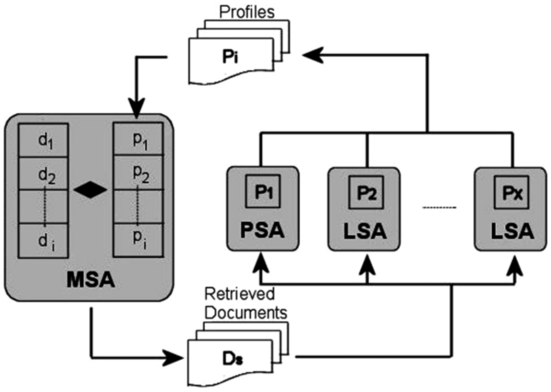

Fig. 5. Schematic representation of the communication between an MSA and one or more PSAs and LSAs during information retrieval. The arrival of a new profile or the update of an existing one ($P_i$) is indexed and stored in the indexed profile list ($p_1$, $p_2$,…, $p_i$). Then, the indexed profile is matched against the stored document list ($d_1$, $d_2$,…, $d_i$) to produce a set of documents that match the particular new or updated profile. $D_s$ is the list of documents that are retrieved by the MSA for a given time-period and sent to agent identifiers that manage the user profiles (P1, P2,…, P$x$).

### *4.1.2. Information retrieval*

The information-filtering role disseminates information about incoming documents by relying on having available personal profiles of members of the community and profiles of locations of the community. Members' profiles are not static, but dynamic, in that they evolve as a result of community members gaining new interests and having new needs. Similarly, location profiles may change, according to the policies specified by members of the community that administer these services. Consequently, changes in profiles prompt PSAs and LSAs to periodically request from MSAs the addition, update or removal of their profiles. MSAs playing the information-retrieval role must keep a record of all of these profiles in a profile index and match these profiles against a stored archive of documents.

We break down the behaviour of the information-retrieval role into its component activities based on the following cyclic behaviour:

1. Input a profile $P_n$;
2. Standardise $P_n$ to produce a new profile $p_n$;
3. Match $p_n$ against all the documents on record, to produce a list of all matching documents;
4. Send a copy of each of the documents to each of the recipients specified in $p_n$.

The behaviour of an agent engaged in the information-retrieval role is shown in Fig. 5. As before, the organisation of the interaction assumes that MSAs in the information-retrieval role communicates with both the PSAs and LSAs. Information retrieval is initiated by either a PSA or an LSA by requesting a profile update from the MSA.

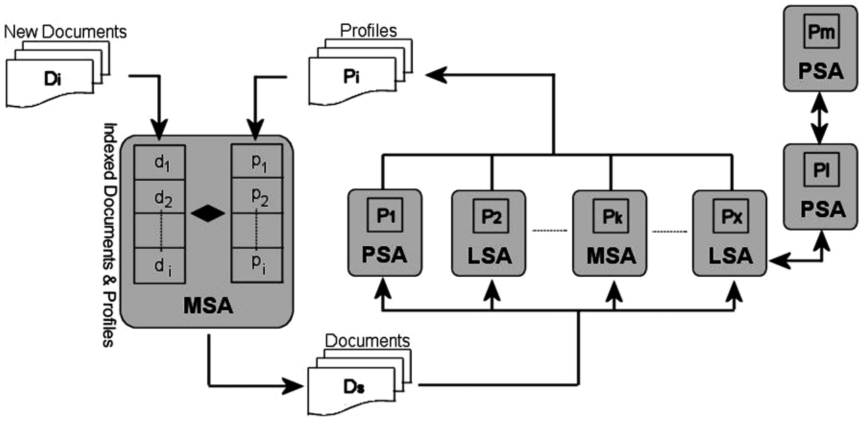

Fig. 6. Information dissemination in LiMe is achieved through the communication between MSAs, PSAs and LSAs. Note that documents and profiles received by an MSA may originate from another MSA in the system, thus allowing more complex organisations of information dissemination. On arrival, if the input is a document ($D_i$) it is added in the list of indexed documents ($d_1, d_2, \ldots, d_i$); otherwise, if it is a new/updated profile ($P_i$), it is added in the list of indexed profiles ($p_1, p_2, \ldots, p_i$). $D_s$ represents a list of documents that have to be delivered to the matching profiles (P1, P2,…, P$x$) for a given time-period. A delivered document may initiate additional communication between LSAs and PSAs due to matching interests in profiles (e.g. P$x$, P$m$, P$l$) that can be exchanged when these devices are in close proximity.

### 4.1.3. Information dissemination

One of the requirements of LiMe has been to combine information filtering and retrieval to develop an information dissemination mechanism that maintains a record of both profiles and documents. We have accomplished this task by designing the MSA's information dissemination role so as to treat both profiles and documents in the same way. We have also included for each profile or document an expiry date, so that a profile or document is removed from the index when it expires. Moreover, we ensure that a profile is checked against all the documents whose lifetime overlaps with that of the profile. This is achieved by comparing profiles to all the documents that are in the document index when the profile is received, and to those that are received before the profile expires and is removed from the profile index. The architecture developed to solve the general information dissemination problem is shown in Fig. 6.

## 4.2. Memory management

Information filtering and retrieval are essential functions of a community information system. However, satisfactory use of the system also depends on the provision of relevant and up-to-date information to the members of the community. For this we need a memory management mechanism that prioritises content according to its current relevance in the community.

We propose to make a distinction between a document's long-term relevance and its short-term relevance. A document's short-term relevance might be expressed simply in the form of an expiry date, which signifies the time at which the subject of the document's

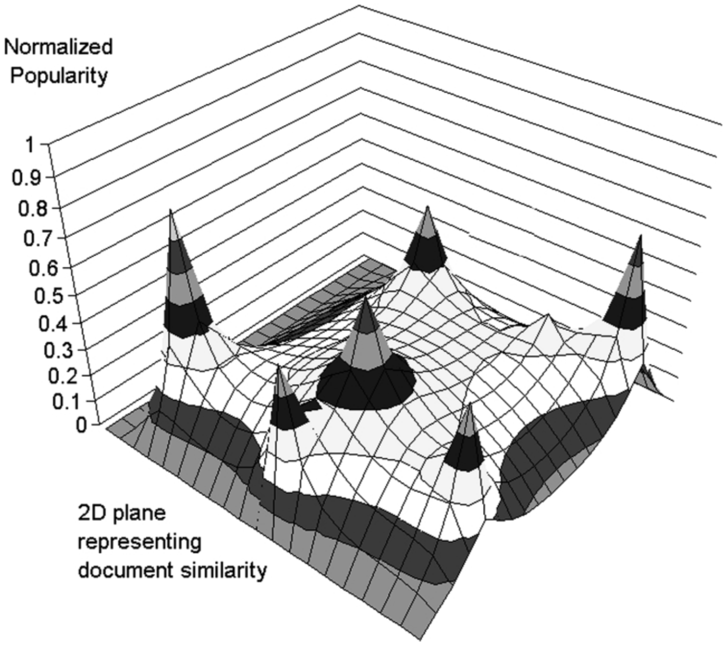

Fig. 7. Graphical representation of popularity across the document database.

content no longer has immediate relevance to life in the community. For example, an announcement of a chess tournament will expire as soon as the chess tournament is over. A document's long-term relevance, however, may continue long after its expiry date. By distinguishing between short-term and long-term relevance of documents we created a short-term memory mechanism that is sensitive to the daily activities of the community, and combined this with a long-term memory that provides the means for the community to preserve the awareness of a shared collective memory. Such an approach is unique to the LiMe system and differs from content storage in the Web or in video-on-demand (VOD) and other repositories (Little and Venkatesh, 1995). We now describe how we based the long-term and short-term relevance of documents on the notion of document popularity.

As depicted in Fig. 7, LiMe considers its document database as a network of documents based on links and similarity (Salton, 1989), each with a certain attributed popularity that is defined in Sections 4.2.1–4.2.3. In this surface, the peaks correspond to documents with high popularity, which pull related documents with them. With time, these peaks fade and eventually, if the documents they represent are not accessed, linked to, or commented on any more, they are archived by the MSA. To facilitate this, the LiMe system continuously keeps statistical measurements on accesses to its document database and knows how much time has expired since a document was first indexed. Furthermore, measurement of similarity and link analysis between documents provides LiMe with a notion of which documents are most representative for a certain topic.

In the following sections, we present three metrics for popularity and their use in the context of the LiMe system. The first and third metrics provide a measure of the long-term

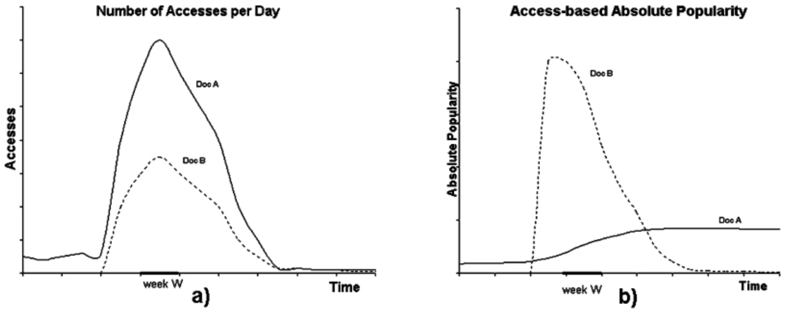

Fig. 8. Daily access count (a) and absolute popularity (b) for two documents.

relevance of documents, while the second metric provides a measure of short-term relevance.

### 4.2.1. Access-based absolute popularity

The MSA maintains an information model that includes the following details about each document:

- *arrival date*—the date that the document became public;
- *access date*—the date when the document was last accessed;
- *accesses*—the number of times the document has been accessed.

Every time a user interacts with a document the access counter for that document is incremented. Based on this measure, the popularity of a document can be computed by taking the quotient of the access count over the age of the document. This method of computing popularity is based on the observations of Laurier et al. (2000) suggesting that popularity is inversely proportional to the age of the documents, and is directly proportional to the number of access hits. Therefore, an access-based absolute popularity measure of documents is defined as:

$$\text{Absolute Popularity}_{\text{doc}} = \frac{\text{Accesses}_{\text{doc}}}{\text{Age}_{\text{doc}}}, \tag{1}$$

where $\text{Accesses}_{\text{doc}}$ and $\text{Age}_{\text{doc}}$ denote the number of access hits and the age of a document, respectively.

### 4.2.2. Access-based relative popularity

Although sufficient for long-term memory management purposes, the definition of absolute popularity as presented in Eq. (1) is inadequate for short-term memory management. For example, consider the example depicted in Fig. 8, where two documents Doc A and Doc B, are 6 months and 7 days old, respectively. Document Doc A receives twice the number of accesses than document Doc B in a particular week *W* (as in Fig. 8a). If we apply popularity as defined in Eq. (1), then Doc B will be considered more popular than Doc A due to the difference in age (as in Fig. 8b). Thus, while appropriate for long-term memory management, absolute popularity fails to identify peaks of interest in older documents. This example suggests that the concept of popularity needs to be measured relative to a specific window of time (i.e. what is hot today, this week or this month). Therefore, in the context of short-term memory management, a more accurate definition of popularity is given by the popularity for a document d relative to time interval $[t1, t2]$ as defined in:

$$\text{Relative Popularity}_{\text{doc};[t1,t2]} = \frac{\text{Accesses}_{\text{doc};t2} - \text{Accesses}_{\text{doc};t1}}{t2 - t1} = \frac{\Delta(\text{Accesses}_{\text{doc}})}{\Delta t} \tag{2}$$

### *4.2.3. Topic popularity*

The two metrics of popularity described in Sections 4.2.1 and 4.2.2 provide measures for short-term and long-term relevance of individual documents. However, LiMe supports and even encourages its users to add new documents that refer to other documents, either through explicit links, or as comments. Therefore, it may also be the case that the relevance of a document increases due to it being concerned with a topic that has a particular relevance to the interests of the community. For example, a document about the local pub may suddenly be more relevant to the community when it is announced that the pub's owner, a local brewery, is laying-off half of its work force. In order to include the relevance of a document's topics into the calculation of the popularity of that document, we add to its basic popularity the weighted sum of the popularity of all the documents that are related to it. As a consequence, the popularity of all the documents related to the same topic will increase whenever the popularity of one of these documents increases (as shown in Fig. 8). This leads to the following definition of topic popularity for document doc at time $t$:

$$\text{Topic Popularity}_{\text{doc};t} = \text{Popularity}_{\text{doc};t} + \alpha \sum_{j=1}^{n} \text{Topic Popularity}_{\text{doc}_j;t}, \quad \text{doc}_j \in \Omega_{\text{doc}} \tag{3}$$

where $\Omega_{\text{doc}}$ is the set of documents that link, or are linked to, by document doc, $n$ the cardinality of $\Omega_{\text{doc}}$, $\alpha$ represents the size of the considered neighbourhood and is given a low constant value between 0 and 1, to avoid instability. Note that due to the dynamic nature of LiMe's contents, a document's topic popularity changes with time, revealing the time awareness of LiMe memory management, a property unique to LiMe.

Using the link structure of hypertext documents in the Internet has proven to be very efficient as a means of adding relevance estimation to results returned by raw keyword matching based search algorithms (Brin and Page, 1998). Likewise, by considering the popularity of related documents, topic popularity of a document as defined in Eq. (3) is a more accurate measurement of the popularity of a document if the reactions of the community to its topic are to be considered.

## 5. Validation

Validation of the LiMe system was achieved first of all by an evaluation of the concepts underlying the multi-agent framework as outlined in the previous sections. Concept testing was carried out at an early stage of the LiMe project. The LiMe concepts were presented

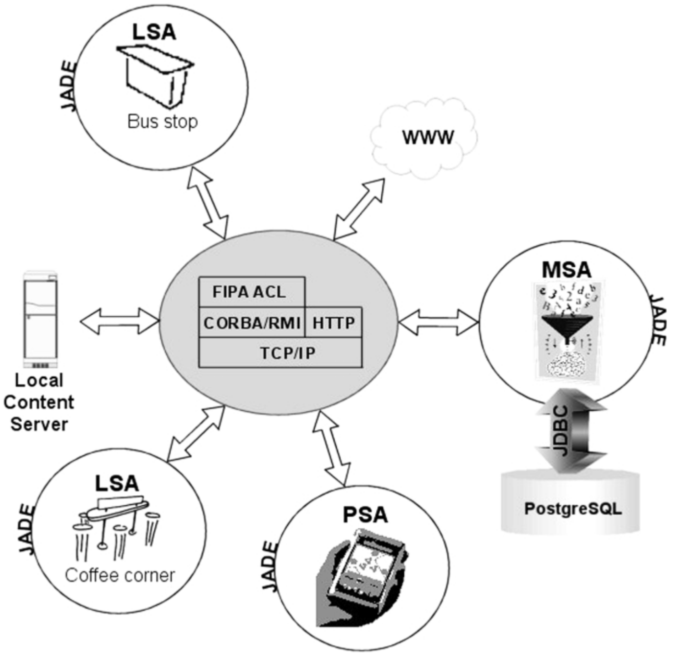

Fig. 9. Implementation of the LiMe multi-agent system with one MSA, one PSA, two LSAs, a local content server and access to the world wide web. The PSA was installed on a PC and linked to a PDA-emulator running on the same machine.

using ‘before-and-after’ scenarios to members of the community. However, these scenarios illustrated the issues discussed in our work only implicitly by describing what kind of information users would see at the interface. As a consequence, the evaluators were not asked explicitly to comment on the role of agents in the context of information dissemination. The results of concept testing are therefore beyond the scope of this paper and the interested reader is referred to Scholten and Rattle (1998).

The second part of the validation consisted of the implementation of a number of instances of the LiMe agents, and installing them on a number of devices that were linked together to form a network. In addition, a set of documents representing potential community content was created for the purpose of this evaluation. The implementation of LiMe agents allowed us to observe the flow of information resulting from the communication between LiMe agents. The installation of LiMe agents in LiMe devices also allowed us to perform a user trial in which members of the LiMe community were asked to interact with and comment on the LiMe system. How the system was implemented and the findings of each type of evaluation are described in the following sections, followed by a discussion of related work.

### *5.1. Implementation*

For the implementation of the LiMe agents described in Section 3 we used the JADE platform that is compliant with the foundations of intelligent physical agents (FIPA) standards for interoperable intelligent multi-agent systems (Bellifemine et al., 1999). JADE was chosen because it supports a rich set of agent platform services and provides tools that simplify agent development and system debugging. In particular, it allows the development of agent behaviour using the JAVA programming language, by extending a set of classes supported by the JADE environment. JADE deals with all those aspects of an application that are domain independent, including message transport, message encoding and parsing, and an agent's life cycle.

Fig. 9 shows a schematic representation of the LiMe agents as implemented in a working prototype. The LiMe agents communicate through the FIPA agent communication language (ACL, 2000). Thus, we were able to create a multi-agent system in which agents exchange information about profiles and references to communal content in the form of URLs. This content can be stored in local content servers dedicated to serving a particular community, or it can be part of the world wide web (WWW). Documents are transferred between devices using HTTP. Each LiMe MSA has access to a database where it stores the indexed documents and profiles. The database has been implemented using PostgreSQL (Momjian, 2000), and is accessed by the MSA through the JDBC interface.

### *5.2. Expert evaluation*

After the system was implemented the designers evaluated the dynamics of the system in terms of performance, robustness and scalability. A brief description of our findings related to each of these criteria is given next. At this point it has to be noted that some of the observations made during the expert evaluation reflect the rather limited set of content that was available for this evaluation.

The first observation of the experts was that the system responded adequately to changes in profiles and the arrival of new documents in that the MSA was able to disseminate relevant documents to the LSAs and PSA. However, they also noted that the response times were quite long and the disseminated documents did not always match the intention of the profiles. Nevertheless, the system proved to be robust and passed tests aimed at producing system errors.

Long response times and some lack of relevance of disseminated documents do not necessarily pose a problem, however, as the system is not intended to be used as a search engine. The system does not need to find the most relevant documents at every request because information dissemination in the LiMe system is an iterative process. This iterative process is achieved by the continuous feedback from document delivery into profile updates, causing additional documents to be retrieved. This process is kept under control by the loss of documents that are 'forgotten' because they are not being accessed. What our work indicates, therefore, is that such an iterative process in which communication of shared content generates additional activities, resulting in the creation of new content and changes in peoples profiles, is an integral part of the creation of an active communal memory.

The relative slowness of the system, discussed above, was also due to using a generic agent communication framework based on FIPA protocols. This particular framework scales up easily, but it required a lot more communication between components. In

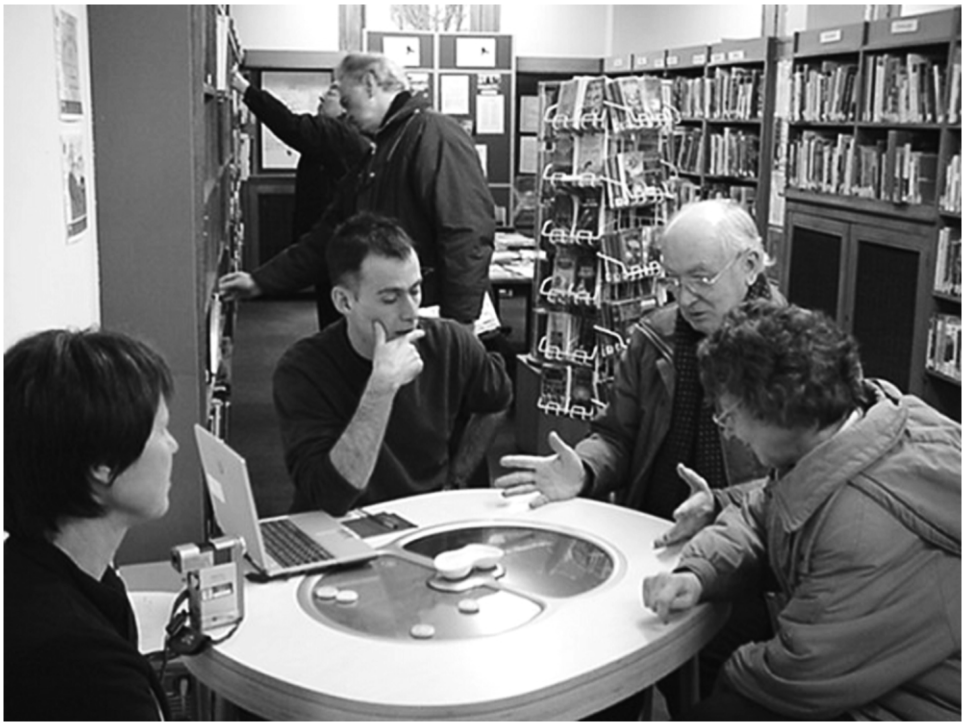

Fig. 10. The LiMe table is being evaluated in the local library by members of the target community. On the right are two users interacting with documents and giving their opinion. On the left are two researchers from the LiMe team recording the interactions and comments.

addition, access to the database proved to be a bottleneck that caused relatively long response times when many requests for document retrieval or filtering were received by many agents at the same time. This bottleneck would represent a severe limitation to the scalability of the system but for the fact that the task of document dissemination can be done by any number of MSAs serving a community. The lack of relevance of disseminated documents resulted from the rather simple matching process between documents and profiles (based on keywords) that was implemented. A more sophisticated matching technique would improve the relevance of disseminated documents.

### *5.3. User testing*

The LiMe multi-agent system has been tested in the LiMe target community, where the LiMe table (Section 2.1.1) with a local link to a laptop computer providing the services of an MSA was installed in the local library (Fig. 10). The evaluators were people from all ages including school children who happened to be in the test locations and were asked to participate. One of the goals of this evaluation was to show how our framework could produce a system that adapts dynamically to the interaction habits of people who visit a particular location in the community. We discuss only the results of the evaluation relevant to agent-based information dissemination. A detailed account of all the LiMe prototype evaluation results can be found in Laurier et al. (2000).

At the beginning of the evaluation session information dissemination was based on a default location profile maintained by the table LSA. During the session people were asked to interact with the content displayed on the table. These interactions caused the LSA to update the location profile so that more documents were retrieved related to the interactions. As a result of many different people interacting with content the table developed a profile that reflected the interests of all the different people, and as a consequence recorded a community-of-interests linked to a specific location. Such a community-of-interests is a complex and dynamic entity. It is precisely this kind of complexity referred to in the introduction, which is best managed by agents, providing the adaptability to allow everyday objects, such as a table, to serve the activities of a local community.

The downside of this implicitly collaborative approach to information dissemination was that some users expressed disappointment with the lack of responsiveness of the system when documents relevant to their interactions did not appear on the table display immediately. It was clear from conversations with these users that they expected that the LiMe system would provide a facility to see only items that match their interests, or to do ad hoc searches. It was explained to these users that the LiMe table was a shared device intended to expose users to emerging communities-of-interests rather than individual interests.

Users gave generally very positive reactions to the prototype and the idea that ‘articles’ could be entered and read from various locations outside the school. However, when it was explained to users that popularity of a document was based on age and the number of times it had been accessed, some took issue with this, asking ‘but can they [people] rate how much they like an item?’ Although these comments suggest that an explicit rating system, allowing users to rate items in terms of how interesting or relevant they are, might be considered to add a useful indication of popularity, research has shown that users are reluctant to provide explicit feedback of this kind (Schwab et al., 2002).

### *5.4. Related work*

Much of the LiME information dissemination work is about combining in a single framework work on information filtering and retrieval. This builds on existing work on personalised delivery of information (Foltz and Dumais 1992), which leads to using agents to deal with the information overload problem (Maes 1994). The software agent paradigm in this context provides a framework that eases the consideration of user interests and context in the process of information filtering and retrieval (Rhodes and Maes 2000; Lieberman et al., 2001). Typically, in these systems, agents use machine learning techniques to gather information on their user’s browsing habits, interests and explicit feedback, in order to build a user profile that is then used for automatic filtering and retrieval of information.

In LiMe, adaptation is provided by PSAs and LSAs. LiMe agents acquire user interests in two ways. Firstly, users are provided with a user-friendly interface to view and edit their own profiles at any time, as this was thought to be a requirement to avoid fears of a black box controlling what a user does and does not see. Secondly, by observing which of the filtered documents get in fact the user's (or users') attention, LiMe agents are able to feedback this information into the corresponding filters, therefore creating a feedback loop that permanently tracks user interests. In addition to user information interests, LiMe agents are also aware of the user's current physical context, as well as his/her social relations with other community members, and this information is also used while filtering and ranking information.

Also relevant in the context of LiMe, is the idea of collaborative filtering (Goldberg et al., 1992). Collaborative filtering as usually proposed (Konstan et al., 1997) is applicable in scenarios where the same information is accessible by a group of users, and it is based on the principle that documents more relevant to a user are potentially more relevant to others in the same group. The use of community-based collaborative filtering also helps to avoid a content-based system falling in a 'local optima', where effectively the user becomes isolated from the rest of the community, only receiving information on a very limited context (Schwab et al., 2000). Work in the project Campiello (Grasso et al., 1998) looks also at agent-based collaborative filtering (Koch, 1998). In contrast to our approach, however, they took a knowledge management approach, assuming explicit user feedback, while LiMe followed an approach modelled after known dynamics of social interactions (as described in Section 1) and monitored relevance of information through observation of user interactions with the system, which was deemed more realistic (Schwab et al., 2002).

## 6. Conclusions

We have studied the application of multi-agent systems technology to develop intelligent information interfaces for connected communities, a class of applications aiming to enhance the way people interact and socialise in geographically co-located communities such as neighbourhoods. Our work has been focused on one of the main problems raised by connected community systems, viz., how to manage the complexity of the information generated by community members interacting with a wide range of every day objects with embedded electronic functionality.

Based on our contribution in the LiMe project, we have designed a multi-agent system for a connected community that makes use of three different types of agents: PSAs to manage information for people in the community, LSAs to manage the information accessibility and flow in physical locations of the community, and that store and manage the complex content that is generated by people interacting with each other and with content at different community locations.

We have designed the functionality of agents using roles, and for the purpose of information management we have identified different roles according to whether an agent acts on behalf of a person, a location or the institutions that may be managing the communal memory generated by and stored in the system. We have relied upon agent communication to support dissemination of the most popular information to people or locations and innovative human–computer interaction mechanisms to visualise this information in the community's environment. The resulting multi-agent system has been developed on a sophisticated software platform that supported agent communication and interaction based on well-defined agent development standards.

Building and testing the system in a real community has given us the opportunity to experiment with some of our ideas, and validate some of our original intuitions about the usefulness of agents in community settings. Most importantly, however, our work has opened up a range of issues, three of which have particular importance:

- *Agent-based interaction design*—is concerned with the lack of a generic framework for designing interactions between agents and between agents and users of the community-based system. Although in the human–computer interaction literature the concept of interaction design attracts more and more attention (Preece et al., 2002), the relationship between this new field and the design of agent functionality is not yet well established.
- *Community interaction models*—the modelling of interactions in connected communities begs for a systematic way of modelling the network of links resulting from the social activities of people at different places of a local community. An interaction model is needed here to act as a high-level specification for the computational and infrastructure requirements of the connected community system.
- *Verifiable agent-behaviour*—we are also concerned with the possibility that connected communities will move control and privacy away from people's hands to autonomous software agents. The associated social risk of connected communities based on agents requires methods and frameworks that allow a user to verify that the behaviour of an agent indeed complies with the intended functionality the agent was given at design time.

All three of the above concerns are shaping the future direction of our work with the aim to provide intelligent information interfaces for connected communities.

## Acknowledgments

The work reported in this paper was supported by the European Commission as part of the Esprit $I^3$ LTR programme, Project No. 25621. The co-author Silvio Macedo was co-sponsored by the Portuguese FCT PRAXIS Grants Programme. We thank all the partners in the LiMe consortium: Philips Design, Eindhoven, the Netherlands; Queen Margaret University College, Edinburgh, UK; Domus Academy, Milan, Italy. We also thank our colleagues Bob Spence and Patrick Purcell for their role in the LiMe project and their valuable comments on earlier versions of this document. We are also grateful for helpful discussion and technical advice in using JADE from Kaveh Kamyab and for the development of some of the information dissemination techniques by Richard Kennaugh. Finally, we thank the two reviewers of this paper for their comments and suggestions.

## References

ACL, 2000. The FIPA Agent Communication Language Specifications, available at http://www.fipa.org/repository/aclspecs.html (accessed 15/2/02).

Agostini, A., De Michelis, G., Divitini, M., Grasso, M.A., Snowdon, D., 2002. Community interactions and memories: the role of community systems. Interacting with Computers, Special Issue on Intelligence and Interaction in Community-based Systems, this volume.

Belkin, N.J., Croft, W.B., 1987. Retrieval Techniques. In: Williams, M.E., (Ed.), Annual Review of Information Science and Technology, pp. 109–145, Elsevier.

Bellifemine, F., Poggi, A., Rimassa, G., 1999. JADE—A FIPA-compliant Agent Framework, Proceedings of Fourth International Conference on the Practical Application of Intelligent Agent and Multi Agent Technology (PAAM99), pp. 97–108.

Brin, S., Page, L., 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine, Proceedings of the Seventh International World Wide Web Conference (Brisbane, Australia), Elsevier, pp. 107–117.

de Bruijn, O., Spence, R., 2001. Serendipity within a Ubiquitus Computing Environment: A Case for Opportunistic Browsing, Proceeding of International Conference of Ubiquitous Computing (Ubicomp 2001), LNCS 2201, Springer, Berlin, pp. 362–369.

Chervenak, A., Vellanki, V., Kurmas, Z., 1998. Protecting File Systems: A Survey of Backup Techniques, Proceedings Joint NASA and IEEE Mass Storage Conference.

Davenport, E.R., Buckner, K., Whyte, A., Bates, K., 1998. Profiles of Interaction Habits. LiMe deliverable 2.1. Living Memory/ESPRIT/I3/25621 Doc no. LIME/QMC/ERD + KaBu + AW + KiBa/98001.

Foltz, P.W., Dumais, S.T., 1992. Personalized information delivery—an analysis of information filtering methods. Communications of the ACM 35 (12), 51–60.

Gaver, W., Martin, H., 2000. Alternatives: exploring information appliances through conceptual design proposals. CHI Letters 2 (1), 209–216.

Goldberg, D., Nichols, D., Oki, B.M., Terry, D., 1992. Using collaborative filtering to weave an information tapestry. Communications of the ACM 35 (12), 61–70.

Grasso, M.A., Koch, M., Snowdon, D., 1998. Campiello—New user interface approaches for community networks. Position paper at Workshop Designing Across Borders: The Community Design of Community Networks CSCW'98, Seattle, WA.

Henriksson, R., 1997. Predictable automatic memory management for embedded systems. OOPSLA97 Workshop on Garbage Collection and Memory Management Atlanta, Georgia. Also available in http://www.dcs.gla.ac.uk/(huw/oopsla97/gc/papers.html).

Heugeneden, H., (Ed), 1993. IMAGINE final report, Eprit project 5362, Siemens, Munich, Germany.

ITN, 2002, New Media. Available at http://www.itn.co.uk/company/newmedia/portfolio/contracts02.html, accessed 02/01/2002.

Jennings, N.R., Sycara, K.P., Wooldridge, M., 1998. A roadmap of agent research and development. Journal of Autonomous Agents and Multi-Agent Systems 1 (1), 7–36.

Kobsa, A., Wahlster, W. (Eds.), 1989. User Models in Dialog Systems, Springer, New York.

Koch, M., 1998. Knowledge Management and Knowledge Agents in Campiello, Proceedings of Workshop on Intelligent Agents in CSCW Fachbereich Informatik, Universität Dortmund, Germany, pp. 44–52. Also available at http://www11.informatik.tu-muenchen.de/publications/pdf/Koch1998a.pdf.

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), 77–87.

Laurier, E., Whyte, A., Dennison, G., 2000. LP4 Field Test Report—Community Interaction with

Prototypes. Living Memory/ESPRIT/I3/25621 Document no. LIME/QMUC + ICSTM + PD/AW + KaBu + EL + LS + HvdW + PM + AA + Odb/0014.

Lieberman, H., Fry, C., Weitzman, L., 2001. Exploring the web with reconnaissance agents. Communications of the ACM 44 (8), 69–75.

Little, T., Venkatesh, D., 1995. Popularity-based assignment of movies to storage devices in a video-on-demand system. Multimedia Systems 2, 280–287.

Maes, P., 1994. Agents that reduce work and information overload. Communications of the ACM 37 (7), 31–40.

Mamdani, E., Pitt, J.V., Stathis, K., 1999. Connected communities from the standpoint of multi-agent systems. Journal of New Generation Computing 14 (4).

Matthews, P., Nikolovska, L., Thursfield, P., Celik, N., 1999. Living Interfaces: Interacting with the LiMe System. LiMe/908020/PGM/99002/LL.

Momjian, B., 2000. PostgreSQL: Introduction and Concepts, Addison-Wesley, Reading, MA.

Noack, D., 1994. Blacksburg Electronic Village: Testbed for the Future. Government Technology 7 (5), 20–76.See also http://www.bev.com/.

Norman, A.D., 1998. The Invisible Computer, MIT Press, Cambridge.

Pitt, J.V., Mamdani, E., Stathis, K., 1998. A Cooperative Information System for the Connected Community. In: Darses, F., Zarate, P. (Eds.), Proceedings of the Third International Conference on the Design of Cooperative Systems COOP'98, vol. 2., pp. 72–76, Cannes, France.

Pitt, J.V., Mamdani, E., Charlton, P., 2001. The Open Agent Society and Its Enemies. Journal of Telematics and Informatics 18 (1), 71–87.

Preece, J., Rogers, Y., Sharp, H., 2002. Interaction Design: Beyond HCI, Wiley, New York.

Rhodes, B.J., Maes, P., 2000. Just-in-time information retrieval agents. IBM Systems Journal 39 (3–4), 685–704.

Salton, G., 1989. Automatic Text Processing. The Transformation, Analysis and Retrieval of Information by Computer, Addison-Wesley, Reading, MA.

Salton, G., McGill, M.J., 1983. Introduction to Modern Information Retrieval, McGraw-Hill, New York.

Scholten, L., Rattle, J., 1998. Living Memory Report on Concept Testing, Living Memory/ESPRIT/I3/25621, Document no. LIME/PH/LS + JR/98001/JR.

Schwab, I., Pohl, W., Koychev, I., 2000. Learning to Recommend from Positive Evidence, Proceedings of International Conference on Intelligent User Interfaces, New Orleans, LA, pp. 241–247.

Schwab, I., Kobsa, A., Koychev, I., 2002. Learning user interests through positive examples using content analysis and collaborative filtering. Available at http://citeseer.nj.nec.com/398141.html (accessed 25/03/02).

Skarmeas, N., 1995. Organizations through roles and agents. International Workshop on the Design of Cooperative Systems COOP'95. Nice, January.

Sumi, Y., Mase, K., 2002a. Supporting the awareness of shared interests and experiences in communities. International Journal on Human–Computer Studies 56, 127–146.

Sumi, Y., Mase, K., 2002b. Conference Assistant for Supporting Knowledge Sharing in Academic Communities. Interacting with Computers. Special Issue on Intelligence and Interaction in Community-based Systems, this volume.

Toni, F., 2001. Automated information management via abductive logic agents. Journal of Telematics and Informatics 18 (1), 89–104. Elsevier.

Wittig, T. (Ed.), 1992. ARCHON: An Architecture for Multi-agent Systems, Series in Artificial Intelligence, Ellis Horwood, Chichester.

Wooldridge, M., Jennings, N.R., Kinny, D., 2000. The Gaia methodology for agent-oriented analysis and design. Journal of Autonomous Agents and Multi-Agent Systems 3 (3), 285–312.