

UNIVERSITY OF SOUTHAMPTON

DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

**TOWARDS A FRAMEWORK FOR DEVELOPING
MOBILE AGENTS FOR MANAGING
DISTRIBUTED INFORMATION RESOURCES**

Jonathan Dale and David C. DeRoure

University of Southampton

Technical Report No. 97-1

February 1997

ISBN: 0-854-325-832

Towards a Framework for Developing Mobile Agents for Managing Distributed Information Resources

Jonathan Dale and David C. DeRoure

Multimedia Research Group

Department of Electronics and Computer Science

University of Southampton

UK

{jd94r, dder}@ecs.soton.ac.uk

Distributed information management tools allow users to author, disseminate, discover and manage information within large-scale networked environments, such as the Internet. Agent technology provides the flexibility and scalability necessary to develop such distributed information management applications. We present a layered organisation that is shared by the specific applications that we build. Within this organisation we describe an architecture where mobile agents can move across distributed environments, integrate with local resources and other mobile agents, and communicate their results back to the user.

1. Introduction

The concept of a global multimedia and hypermedia information space is starting to emerge. Inter-networked environments, such as the Internet, have made more electronic information available than ever before. Computer information, which was primarily text-based, is now branching into the areas of two and three-dimensional graphics, sound, video and video on demand and temporal media.

Yet networks of information present the electronic information community with a fundamental problem: how is information to be created, published, organised, managed, navigated and searched on a global scale? Information management has only been considered previously on relatively small scales and has dealt primarily with text-based media; for example, libraries, encyclopedias and dictionaries. For the most part, these sources of information have had to be painstakingly compiled and cross-referenced by hand.

Current technology in this area provides relatively few and generally unsophisticated tools for information management. Those tools that do exist tend to be aimed at a specific information resource and lie very much within the area of information retrieval. If we consider the World Wide Web as a case study, the prominent tool that exists is a simple search mechanism for retrieving information; few tools exist for keeping track of and managing documents and the hypermedia links between them. Recently, these types of information management activities have been termed *distributed information management* [24].

We believe that agent technology can be employed to build distributed information management tools. Agents provide flexibility and a convenient abstraction upon which to build applications; they answer distribution and scalability issues through mobility and their dynamic and modular nature addresses the problem of integration with legacy systems.

In this paper, we present a layered organisation for our agent-based applications. This organisation is derived from analysing the commonalities that are shared by our separate applications. Within this model, we describe an architecture for mobile agents that is suitable for distributed information management applications. From a software engineering perspective, the layered organisation is a recognised method for developing and maintaining large-scale applications and allows the mobile agent architecture to be independent of any underlying agent technology. Furthermore, the architecture is also independent of any particular application, which promotes reuse for our specific applications.

In section 2, we present four main tasks that we have identified as distributed information management activities. Section 3 describes the layered organisation for designing and integrating distributed

information management applications. Section 4 details the mobile agent architecture that we have developed. Section 5 presents a discussion on related work and is followed by our conclusions.

2. Distributed Information Management

Distributed information management is an initiative being promoted by the UK Technology Foresight panel on IT and Electronics to develop methods for managing change and evolution in distributed systems, particularly targeting multimedia applications across large scale, publicly accessible networks. The issues to be addressed by this programme of work include:

- The management of massive multimedia information stores distributed across large numbers of media-capable servers.
- Managing precision and consistency of information in federated systems.
- The extension of distributed technology to capture the semantic understanding of applications.
- Detached information handling, subsequent arbitration and reconciliation.
- Automatic content-based indexing, analysis and transformation and retrieval of information sources.

The following sections describe four issues which we have identified as being key to achieving successful distributed information management [8]. We see these tasks as being represented by cooperating communities of agents which interact in an intelligent manner to perform tasks, whether initiated by the user, or autonomously and pro-actively as a reaction to changes in the local environment or as a fulfilment of an agent's goal set [29].

2.1. Resource Discovery

The purpose of resource discovery is to search through distributed information systems that are known to the user and to present other sources of relevant information. Automatic resource discovery tools are needed because there are too many discovery information systems for the user to search manually. However, the searching algorithm must be accurate to ensure that relevant data is not overlooked and that irrelevant data is discarded before it reaches the user.

Another function of resource discovery is resource monitoring: the active task of notifying the user when the contents of a resource change. This is particularly useful if the user is monitoring temporal media, for example stock prices, but can also be used to indicate when a user should revisit information resources.

Most current information retrieval and resource discovery algorithms are text-based in nature, as with search engines for the World Wide Web, for example and are unsuited to multimedia data.

2.2. Information Integrity

As information becomes distributed across wide areas, information integrity becomes an issue to be addressed [7]. Due to the problems of packet delay and subsequent latency associated with networks, it is difficult to ensure that consistency updates are made in a timely fashion. Additionally, when considering collaborative working environments, versioning and update control needs to be implemented to ensure that alterations are not lost.

In terms of hypermedia systems, for example Microcosm [11] and Hyper-G [15], the problems of hypermedia link and document consistency must also be handled. Link consistency deals with ensuring that the integrity of links is maintained, even if the source or destination anchor moves. In most cases, link inconsistency is relatively easy to deal with, since it either involves removing the link (if it is no longer valid), or re-pointing the start or end anchor to the new location.

Document consistency is a much more difficult problem, since it deals with the contents of a document changing and can also infer link inconsistency. If the end anchor of a link points to a keyword in the centre of a given document and that keyword is subsequently deleted, how is this resolved?

Unfortunately, in most instances of inconsistency, some form of user intervention will be required to ensure that the damage is repaired correctly. In all but the most trivial of cases, consistency algorithms

can do little more than highlight the problem for the attention of the user. However, in hypermedia systems that employ separate link databases, such as Microcosm and Hyper-G, the task of consistency checking is made simpler since interaction only occurs between the link databases. Where link information is embedded, as with the World Wide Web, interaction must occur between all documents within the system.

2.3. Navigation Assistance

Navigation assistance is the process of assisting the user in navigating some form of information resource. This resource could be the information contained within a distributed information system or could be the information generated by a number of resource discovery algorithms. Either way, a navigation assistance algorithm can be employed to protect the user from information overload.

Oren [25] sums up the role of this algorithm as “the human reference librarian who does not comprehend the material in articles being sought, but does understand the conventions of card catalogues, abstract collections, citation indexes and bibliographical references. Because these relations can be made explicitly in hypertext they can be utilised without, for instance, having any deep comprehension of the meaning of any article title.”

Wilkins [28] further describes navigation assistance as an algorithm that can fulfil the following requirements:

- Reduce the number of possible destinations for a user to choose from when too many are offered.
- Expand the number of destinations for a user to choose from if none are apparent.
- Gather and intelligently combine existing sources of information that already exist within the system and present the user with one combined source of information instead of many.

2.4. System Integration

A key aspect of distributed information management is the ability to manage information resources that are available across heterogeneous networks, heterogeneous platforms and are represented by heterogeneous protocols and data formats. To this end, there is a need for distributed information management tools to be able to integrate with a wide range of distributed information systems.

However, integration is more than protocol conversion since there is also a semantic problem to overcome. For example, how are hypermedia links translated between the World Wide Web and Hyper-G? If there is more information represented in a Hyper-G link, how is this stored within a World Wide Web link? Furthermore, is it possible to apply links across all distributed information systems? If so, where are these links stored and which system has the responsibility of resolving and managing them?

There are two complementary approaches to system integration: arming distributed information management tools with the necessary information to be able to converse with multiple distributed information systems, and equipping them with the necessary information to make consistent and meaningful semantic link and data conversions.

3. A Layered Organisation for Agents

At the Multimedia Research Laboratory, we have experience with developing multimedia-related applications. Current projects involve distributed hypermedia linking [11,1], content-based navigation and retrieval [18] and corporate information management and information linking across applications [3]. We recognise that there are advantages to using agents within and across these types of applications [13]. Upon reflection, we have identified the commonalities between our separate applications in order to promote flexibility, dynamism and modularity. To support these properties, we have designed a layered organisation [5] that is motivated from two different perspectives:

- For the architecture designer, a layered organisation facilitates the development of each layer in parallel by clearly identifying the interface between adjacent layers. Indeed, the code of a layer relies on the services provided by the layer below but is independent of its actual implementation.
- For the application developer, a layered organisation offers abstraction and a wide set of primitives

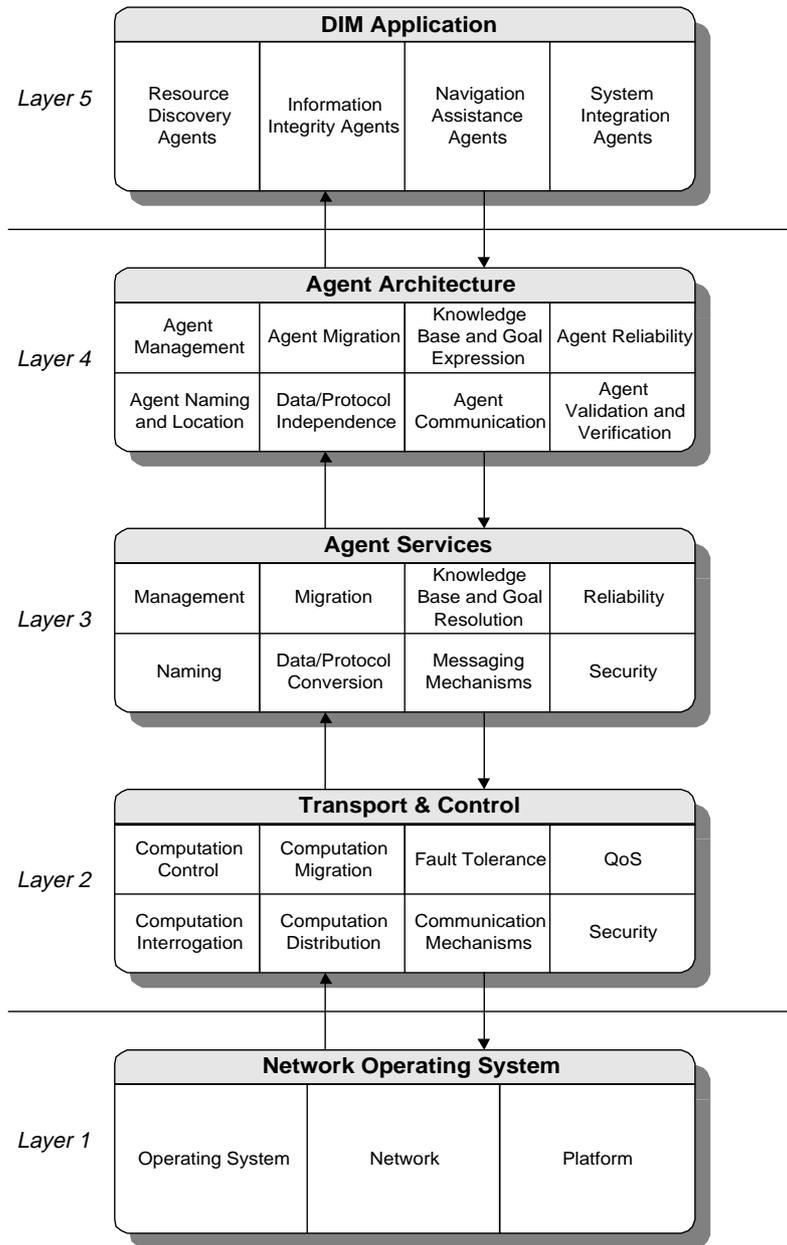


Figure 1 : Layered Organisation

within which a varied range of agent activities can be expressed. Additionally, it is a common starting point for applications to begin communicating and sharing information with each other.

Figure 1 illustrates the layered organisation that we are using to develop our agent-based applications.

Layer 1, the Network Operating System layer, is composed of a suitable transportation protocol and network operating system. This includes network-aware personal computers, workstations and servers, and the relevant software to support network access and multi-tasking.

The Transport and Control layer provides general facilities to build distributed applications with migratable components. By executing user programs, it creates *computations* which may be composed of several *tasks* running in parallel. It also offers distribution by allowing computations to migrate and as a result, computations can be regarded as a set of distributed tasks. Furthermore, the layer provides tasks with different mechanisms for communication, such as mailboxes, synchronous channels or even some form of distributed memory.

In addition, not only does the layer offer facilities for traditional distributed computing, like processor

farming, task stealing and transparent migration, but it also provides the means to *monitor* and *control* computations. The originality of the layer lies in the fact that distributed computations can be driven by mastering their resource consumption; a generic view is adopted which abstracts resources through a notion of *energy*. Any computation is allowed to proceed if it is given some energy. When a computation consumes all of its given energy, an asynchronous notification is posted which allows the user program to decide what action to perform next, for example, to give more energy to continue the computation. Two primitives are also provided to transfer more energy to a computation or to steal energy from a computation. This functionality is defined and implemented in a prototype language called Quantum [23]. Layer 2 also offers services that are reflected from the Network Operating System layer, for example, quality of service and security.

The essence of this layer is to provide the ability to integrate control into a distributed system at a very early stage; cost models can be developed on top of the energy scheme so that it can be tailored for the needs of individual agent-based applications.

The Agent Service layer, layer 3, is built upon the services offered by the Transport and Control layer and presents general facilities for managing, naming, controlling and migrating agents. This layer also provides knowledge-based, goal-directed reasoning. To make this layer dynamic, flexible and adaptable to the needs of the layers above, interchangeable modules allow different data formats, like PIF [17] and KIF [10], and communication protocols, like AgenTalk [16], Agent-K [6] and KQML [9], to be used and converted between. It is important to note that this layer is both application and platform independent, which allows us to reuse specific solutions that have already been developed.

The fourth layer, the Agent Architecture layer, gives form and structure to the nature of the agents that the application will be able to use. This layer mirrors some of the services provided by the Agent Service layer, but these have been refined and augmented within a specific context for a particular application or set of applications. The next section discusses the role of this layer in more detail.

The final layer is the Application layer where distributed information management agents operate.

The way in which these layers interact will be crucial; the lower layers must provide the core functionality required by the upper layers in a dynamic and generic fashion. We are using this layered organisation for developing distributed information management applications and we are investigating the possibility of using these ideas within other agent-based applications. In the remainder of the paper, we focus on layer 4 which offers an architecture within which mobile agents can operate.

4. A Mobile Agent Architecture

The first author [4] has designed a possible architecture for mobile agents which is suitable to operate in a heterogeneous, networked environment such as the Internet. It addresses some of the issues raised by Chess [2] regarding the safety and security of systems across the Internet as mobile agents become more prevalent. The architecture is not able to provide a complete solution to all of these issues, due to the fact that some specific solutions are required, but it can assist by being configured by the application.

In this section, we describe the architecture from the layered organisation perspective. The architecture is made independent of both the agent paradigm, by building it above the Agent Service layer, and the application, by allowing it to be parameterised by the Application layer.

The architecture, which appears in figure 2, is comprised of both *static* and *mobile agents*; static agents provide resources and facilities to mobile agents and mobile agents move between network domains taking advantage of these resources to fulfil their goals. A *domain* is a logical boundary used to delimit nodes, agents and resources into manageable and distinct entities.

It is envisaged that mobile agents will possess the ability to migrate to the most appropriate location (given the constraints of security, cost, distance and other such factors) and will comprise behaviour and some form of knowledge base. The behaviour segment of an agent will describe the functionality of the agent, for example a resource discovery agent or a link maintenance agent. The knowledge-base will contain the “intelligence” of the agent, that is, its experiences, its goals and its beliefs. This segment will also contain temporary information that the agent is working upon before it is returned to the user for appraisal.

The next sections take each component of the mobile agent architecture and describe its general

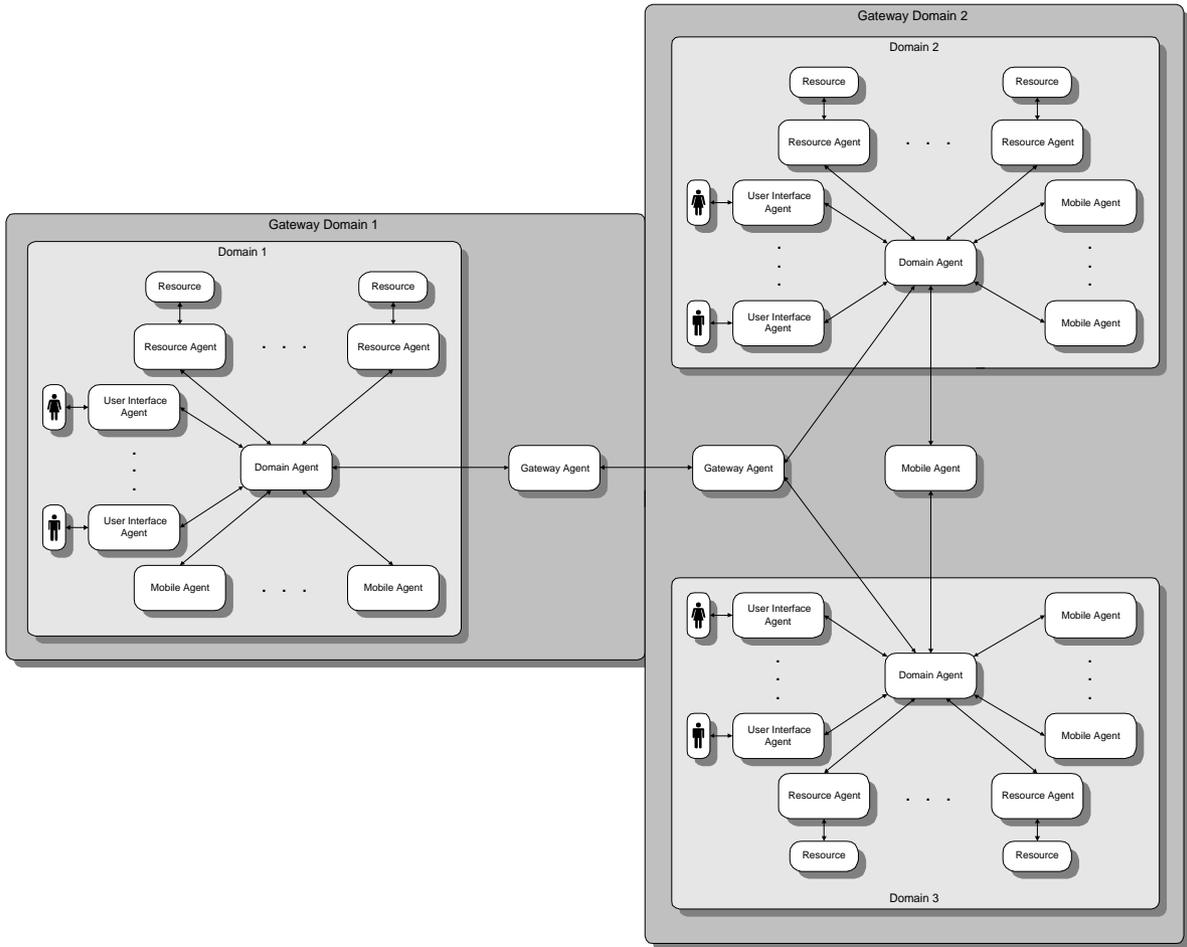


Figure 2 : Mobile Agent Architecture

activities and purpose within the infrastructure.

4.1. Domain Agent

The domain agent is a static agent that coordinates the activities that occur within a domain. The domain agent offers a number of services related to information resources, users and agents within a given domain:

1. It provides a migration service to mobile agents wishing to leave the domain. The domain agent is responsible for ensuring that the mobile agent is transferred successfully. At this level, there is negotiation, security and fault tolerance to take into account; by contrast, the Agent Service layer is concerned with merely transporting an agent from one location to another.
2. It authenticates and performs a validation check on mobile agents wishing to enter the domain. Agents that cannot be authenticated or fail the validation check are rejected.
3. It launches received mobile agents in a suitable run-time environment, which will be related to the amount of trust given to the mobile agent. Agents from trustworthy nodes may be allowed access to more information resources than agents from unknown nodes. A suitable run-time environment will depend upon the access level granted to the mobile agent and the functions that it wishes to perform.
4. It mediates access to information resources at a domain level. When a mobile agent enters a domain, the domain agent performs a security check upon it and allocates the mobile agent a domain-wide permission. This permission can subsequently be used by resource agents to determine whether to allow the mobile agent access to a resource, and by other mobile agents to decide whether to communicate and trade information with the mobile agent (cf. resource mediation in section 4.2).
5. It provides a central registration area within a domain where static and mobile agents can register

and advertise their resources and interests. Thus, a domain is a also meeting point which allows agents to gather and share information to resolve their goals.

6. It is the initial point of contact within the domain. All message interactions between static and mobile agents are initially routed through the domain agent. This allows agents to communicate in an asynchronous fashion. However, once two agents have become aware of each other's presence (via the domain agent), they can initiate a direct and synchronous communication between themselves, to transfer high-bandwidth data for example.
7. It ensures that the domain does not become overwhelmed by mobile agents. This can be achieved by limiting the amount of agents that can exist within the system at a given time or by implementing a charge for processor cycles and resource access. This way, even if a mobile agent attempted to monopolise a resource, it could only do so for as long as its credit lasted.
8. It advertises public information resources and the presence of mobile agents to enquiring agents from both inside and outside of the domain. In this way, mobile agents can interrogate the contents of a particular domain before moving, to ensure that it contains information resources that will help it to achieve its goals. It also provides a mechanism for agents to locate each other.
9. It periodically announces its presence (through broadcasting or multicasting) to the larger community of domain agents. This allows each domain agent to maintain a list of other domains that is available for mobile agents to use in determining their next jump point.

The domain agent, is the central manager within a domain and is ultimately responsible for ensuring that security is enforced within the domain and that agents can communicate with information resources and each other.

4.2. Resource Agent

Resource agents are static agents that exist within a domain to provide a level of abstraction between the information resource to which they provide access and the requesting mobile agents. The purpose of a resource agent is to mediate access to a particular information resource for a mobile agent; the resource agent understands how to access the resource and also understands the permission structures associated with the resource.

As such, the resource agent has a number of functions:

1. It is fully conversant in the protocols of the information resource. The information resource should be completely accessible by mobile agents, so that user intervention directly on the resource is not required. This implies that the user has a set of distributed information management mobile agents that are able to access, modify, locate and manage their information resources through the individual resource agents.
2. It provides an ontological description for each of the services offered by the resource. These services are the methods by which mobile agents interrogate, update and manage resources.
3. It mediates access to the resource at a resource level. When a mobile agent requests a service, the resource agent can either grant access according to the mobile agent's domain-wide permission (allocated by the domain agent), or it can impose further constraints and checks upon the mobile agent itself (cf. domain mediation in section 4.1).
4. It advertises the presence of the information resource by registering the services available to the domain agent. In this way, mobile agents can be aware of what information resources are present within a domain (by querying the domain agent) before they arrive at a domain.

The resource agent, through the Agent Service layer, provides conversion between the data formats and protocols that the individual resource is using, but the Application layer is responsible for providing the ontological mapping between them.

The interaction between resource agents and mobile agents will form the crux of the distributed information management aspect of the infrastructure. The flexibility that mobile agents are afforded in accessing these information resources and the manner in which they interpret the results will determine their usefulness to the user.

However, it is not envisaged that information resources are just distributed information systems; they

can be any system which presents an external interface through which it communicates or can be accessed. This form of legacy integration allows resource agents to be developed for any type of resource; electronic mail, USENET news, databases, the World Wide Web, Microsoft Word, and so on.

4.3. Mobile Agent

Mobile agents, as their name suggests, are the components within the framework which can migrate between domains. They are the mechanism by which the user exercises control over their own distributed information resources (through the appropriate resource agents) and gain access to other shared information resources (again, through the relevant resource agents).

Initially, a user launches a mobile agent within a given domain, which is called the *host domain*. Mobile agents are equipped with a set of user-defined goals which describe the nature and limits of their functionality, for example, a resource discovery agent is a mobile agent with a different goal set than, say, a navigation assistant agent. In addition to the limits on the functionality that users place on their mobile agents, it is probable that the mobile agents themselves will encounter limits (in the form of security, authentication, validation, etc.) that exist within domains. In some cases, these limits (for example, lack of funds to pay for resources) will compromise the goals that have been given to the mobile agent.

The essential functions of mobile agents will be defined by the distributed information management tasks that they are allocated. However, they have the following interactions with the framework:

1. Mobile agents determine where to migrate to next by initially querying the local domain agent for a list of domains of which it is aware. The mobile agent can then use this information to contact each domain agent and determine which one offers a suitable set of information resources.
2. Mobile agents are authenticated by an electronic signature that they carry. This signature may be encrypted but must certainly be verifiable with the host domain of the mobile agent or a third-party domain. Additionally, mechanisms must be employed to ensure that mobile agents are not compromised during transit or within a domain, and that the behaviour of a given mobile agent is consistent with the behaviour that it was given when it was launched. As it is impossible to guarantee these requirements, since a malicious agent may be launched from an authentic domain, mobile agents should be controlled as tightly as possible and consequently given a minimum amount of permission.
3. Mobile agents possess the characteristic of persistence; they use migration as a mechanism to achieve longevity. In this way, they are not reliant on the host domain that launched them. This is particularly useful in nomadic computing, where the user is only connected to the network for short periods of time.
4. Mobile agents may communicate not only with resource agents, but also with other mobile agents to achieve their goals.
5. Mobile agents should transmit the results of their findings and actions regularly to their user on the host domain. This way, the user can ensure that the mobile agent is functioning correctly and also helps prevent the mobile agent from becoming too large (in terms of size) to move or to process.

Mobile agents are the ultimate effector of change within a distributed information management environment, and as such, should be treated cautiously. The possibility for mobile agents to effect changes or gain access to private information must be closely monitored by the domain agent and the individual resource agents. A comprehensive authentication, validation and access permission system is vital to ensure that malicious agents can do no harm to a system and that legitimate mobile agents can fulfil their goals. However, it is not the function of the Architecture layer to provide a complete solution to the security issue; this will need to be handled by the Application layer as well.

4.4. User Interface Agent

The user interface agent provides a level of abstraction for the user away from the details of the mobile agent framework. Maes [21] describes an interface agent as “...a *personal assistant* who is collaborating with the user in the same work environment.” It is essentially an interface agent that performs the following tasks:

1. It launches mobile agents on behalf of the user and tracks their progress and position.
2. It provides mobile agents with a communication point through which they can return the results of their tasks. User interface agents may need to possess local mobility to allow for node failures, since they need to be executing even when a user is disconnected from the system.
3. It organises and pre-processes information returned from mobile agents into a form that is suitable for the user. This may involve filtering out replicated information, presenting urgent information to the user quickly, rendering information using the tools that the user is most familiar with, etc.
4. It provides domain agents with the authentication credentials of a user's mobile agents. It is the task of the user interface agent to ensure that the requesting entity is itself an authentic domain agent, to make certain that authentication credentials are not given to unauthorised systems.

User interface agents provide the user with a window onto their agents, their status, their results and the mobile agent framework. The creation of new distributed resource management agents could either be the task of the user to write (or reuse from existing templates with different parameters), or the task of the user interface agent to generate automatically, based upon the interpreted requirements of the user. The latter approach is most desirable, but is still a long way off given the technology level that currently exists.

4.5. Gateway Agent

A gateway agent is a static agent that provides access to and from domains. It allows a number of domains, related perhaps by geographic, commercial or other commonalities, to be treated as a single domain, called a *gateway domain* (figure 2). Gateway agents offer the following services to the domains within its locale:

1. It provides firewalling for a corporation or organisation where information within the gateway domain is of a protected or secure nature. Mobile agents can be restricted both on entry to and exit from the collective domain.
2. It equips mobile agents entering a gateway domain with the suitable components to allow the mobile agent to communicate within the collective domain.
3. It provides a hierarchy of domains where queries from domains lower in the hierarchy can be dealt with and resolved by domains higher in the hierarchy. It is envisaged that the hierarchy will be similar in nature, though not restricted to an agent-like DNS, where gateway agents provide access to countries at the root level, regions at the next level and so on down to the constituent domains at the leaves.

At the Application layer, gateway agents are the ideal place at which corporations, governments, countries, etc. can ensure that local ordinances are observed within their domains. For example, the gateway domain for the Ministry of Defence will have tighter security restrictions than, say, the gateway domain of the United Kingdom. They also provide a mechanism for ensuring that undesirable material does not enter the domain and restricted information does not leave the domain.

Through the developed hierarchy of gateway domains, gateway agents add scalability to the system, since messages that are for agents within a given domain or gateway domain only travel within that domain. The burden of dealing with messages to destinations unknown can be spread across the hierarchy, thus alleviating the pressure on individual gateway and domain agents.

5. Discussion and Related Work

There are many systems being presented that offer services similar to layers 2 and 3 of our layered organisation, such as Agent-TCL [12] and TACOMA [14]. Such systems are concentrating on the concepts of agent communication and migration; issues such as agent control, security and reliability are the responsibility of the layers above to provide. It is interesting to note that although these systems offer the facilities of spawning and migrating new agents, they offer few facilities for providing subsequent control of these agents and no intrinsic support for multi-agent systems. The Transport and Control layer of our layered organisation provides such basic services through the energy model presented and this system can be utilised by the Agent Services and Agent Architecture layers to

provide high-level control over agents.

Telescript [27] is an architecture to support mobile agents for electronic commerce. It is oriented around providing high levels of security, places for agents to meet and mechanisms for allowing agents to communicate. There is a close analogy between places and resource agents, as well as between engines and domains, but the purpose of each in Telescript is not as well defined. Moreover, the Telescript architecture is tied to its particular implementation and platform (to ensure conformity and to enforce security) and has been criticised for not being more open and modular. However, Telescript has begun to tackle the questions of security and resource protection across public access networks by enforcing very stringent restrictions upon mobile agents.

The itinerant agents system [2] is a generic framework for implementing secure, remote applications in situations such as the Internet. There are a number of similarities between the architecture that they describe for supporting mobile agents and the one detailed in this paper; the need for ontologies, high security features, resource access, etc. However, the framework lacks the layered organisation that we present and as such, there is a functional crossover inside the individual components of their system. Moreover, it is not clear how the framework is to be realised, and how the issues of flexibility, modularity and extensibility are to be tackled. The itinerant agents framework does discuss ideas such as security, resource registration and matching, etc. which would make useful starting points in the development of our distributed information management applications.

Many of the agents that we are proposing to undertake distributed information management tasks already have parallels within the agent community. For example, we envisage the user interface agent to be an expression of the work undertaken by Pattie Maes [20] and Henry Lieberman [19] in developing agents that learn from the user and adapt to their needs. Additionally, information retrieval agents are being employed to utilise the information that they receive from many different resources, such as the Information Retrieval Agent [26] which can search for loosely specified articles from differing document repositories.

6. Conclusion

In this paper, we present a layer organisation for our agent-based applications and we introduce an architecture for mobile agents. We are using this architecture to build distributed information management tools and are investigating using an agent programming language, such as APRIL [22].

The tools that we wish to develop at the Application layer will be built around the architecture that we have developed. For example, a resource discovery tool can be implemented as a mobile agent which travels between domains and interacts with resource agents and other mobile agents. An information integrity tool can also be implemented using a mobile agent. However, it is the responsibility of the resource agent managing a particular resource to launch and control these agents to ensure that the integrity of documents and hypermedia links is maintained.

We believe that the development of the composite layers of our layered organisation will provide our agent-based applications with a common starting point, which includes and is based around the concept of a mobile agent. This will also allow them to begin exploiting distributed information management related tasks to help the user discover, navigate and manage information resources that exist within the global information space that is developing.

Acknowledgements

This research has been funded by an EPSRC Research Studentship and partially supported by EPSRC Research Grants number GR/K73060 and number GR/K30773. The authors would particularly like to thank Luc Moreau and Stuart Goose for their help in developing the layered organisation, and the members of the 'Voyager' group for their general contribution to this research.

References

- [1] CARR, L. A., DeROURE, D. C., HALL, W. and HILL, G. J., The Distributed Link Service: A Tool for Publishers, Authors and Readers. *In*: Proceedings of the Fourth International World Wide Web

- Conference: The Web Revolution, Boston, USA, December, 1995.
- [2] CHESS, D., GROSOFF, B., HARRISON, C., LEVINE, D., PARRIS, C. and TSUDIK, G., Itinerant Agents for Mobile Computing, IBM Research Report RC 20010, 1995.
 - [3] CROWDER, R. M. and HALL, W., The Use of Interactive Media as a Training and Operational Interface in the Advanced Factory. *In: Proceedings of the Third International Conference on Factory 2000—Competitive Performance Through Advanced Technology*, York, UK, pages 106-110, 1992.
 - [4] DALE, J., A Mobile Agent Architecture to Support Distributed Resource Information Management, PhD Mini-Thesis, University of Southampton, UK, May 1996.
 - [5] DALE, J., MOREAU, L. A. V. and GOOSE, S., A Layered Model for Supporting Agent-Based Applications, Technical Note, University of Southampton, UK, November 1996.
 - [6] DAVIES, W. H. E and EDWARDS, P., Agent-K: An Integration of APOL and KQML. *In: Proceedings of the CIKM-94 Workshop on Intelligent Agents*, Finin, T. and Labrou, Y., Eds., Maryland, USA, 1994.
 - [7] DAVIS, H. C., Data Integrity Problems in an Open Hypermedia Link Service, PhD Thesis, University of Southampton, UK, 1996.
 - [8] DeROURE, D. C., HALL, W., DAVIS, H. C. and DALE, J., Agents for Distributed Multimedia Information Management. *In: Proceedings of the First International Conference on the Practical Application of Agents and Multi-Agent Technology*, London, UK, pages 91-102, 1996.
 - [9] FININ, T., LABROU, Y. and MAYFIELD, J., KQML as an Agent Communication Language. *In: Software Agents*, Bradshaw, J., Ed., MIT Press, 1995.
 - [10] GENESERETH, M. R. and FIKES, R. E., Knowledge Interchange Format Version 3.0 Reference Manual, Technical Report Logic 92-1, Stanford University, USA, January 1992.
 - [11] GOOSE, S., DALE, J., HILL, G. J., DeROURE, D. C. and HALL, W., An Open Framework for Integrating Widely Distributed Hypermedia Resources. *In: Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Hiroshima, Japan, 1996.
 - [12] GRAY, R. S., Agent TCL: A Transportable Agent System. *In: Proceedings of the CIKM Workshop on Intelligent Information Agents, Fourth International Conference on Knowledge Management*, Mayfield, J. and Finin, T., Eds., Baltimore, USA, 1995.
 - [13] JENNINGS, N. R., FARATIN, P., JOHNSON, M. J., O'BRIEN, P. and WIEGAND, M. E., Using Intelligent Agents to Manage Business Processes. *In: Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK, pages 345-360, 1996.
 - [14] JOHANSEN, D., van RENESSE, R. and SCHNEIDER, F. B., An Introduction to the TACOMA Distributed System, Computer Science Technical Report 95-23, University of Tromsø, 1995.
 - [15] KAPPE, F., PANI, G. and SCHNABEL, F., The Architecture of a Massively Distributed Hypermedia System. *In: Internet Research: Electronic Networking Applications and Policy*, 3(1), pages 10-24, 1993.
 - [16] KUWABARA, K., ISHIDA, T. and OSATO, N., AgenTalk: Coordination Protocol Description for Multi-Agent Systems. *In: Proceedings of the First International Conference on Multi-Agent Systems*, page 455, San Francisco, USA, 1995.
 - [17] LEE, J., GRUNNINGER, M., JIN, Y., MALONE, T., TATE, A. and YOST, G., The PIF Process Interchange Format and Framework Version 1.1, Working Paper #194, MIT Center for Coordination Science, 1996.
 - [18] LEWIS, P. H., DAVIS, H. C., GRIFFITHS, S. R., HALL, W. and WILKINS, R. J., Media-based Navigation with Generic Links. *In: Proceedings of the Seventh ACM Conference on Hypertext*, Washington DC, USA, pages 215-223, 1996.
 - [19] LIEBERMAN, H., Letizia: An Agent that Assists Web Browsing. *In: International Joint Conference on Artificial Intelligence*, Montreal, Canada, August 1995.
 - [20] MAES, P., Social Interface Agents: Acquiring Competence by Learning from Users and Other

- Agents. *In: Software Agents—Papers from the 1994 Spring Symposium, Technical Report SS-94-03, Etzioni, O., Ed., pages 71-78, 1994.*
- [21] MAES, P., Agents that Reduce Work and Information Overload. *In: Communications of the ACM, 37(7), pages 31-40, 1995.*
- [22] McCABE, F. G. and CLARK, K. L., April—Agent Process Interaction Language. *In: Intelligent Agents, Lecture Notes in Artificial Intelligence, 890, Wooldridge, M. J. and Jennings, N. R., Eds., Springer-Verlag, pages 324-340, 1995.*
- [23] MOREAU, L. A. V. and QUEINNEC, C., On the Finiteness of Resources in Distributed Computing, Technical Report, University of Southampton, UK, November 1996.
- [24] Office of Science and Technology, Technology Foresight Panel on IT and Electronics, Volume 8, HMSO Publications, 1996.
- [25] OREN, T., The Architecture of Static Hypertexts. *In: Proceedings of Hypertext'87, Association for Computing Machinery, pages 291-306, 1987.*
- [26] VOORHEES, E. M., Software Agents for Information Retrieval. *In: Software Agents—Papers from the 1994 Spring Symposium, Technical Report SS-94-03, Etzioni, O., Ed., pages 126-129, 1994.*
- [27] WHITE, J. E., Mobile Agents. *In: Software Agents, Bradshaw, J., Ed., AAAI Press/MIT Press, 1996.*
- [28] WILKINS, R. J., The Advisor Agent: A Model for the Dynamic Integration of Navigation Information within an Open Hypermedia System, PhD Thesis, University of Southampton, UK, 1994.
- [29] WOOLDRIDGE, M. and JENNINGS, N. R., Intelligent Agents: Theory and Practice. *In: Knowledge Engineering Review, 10(2), June 1995.*