The Knowledge Engineering Review, Vol. 14:4, 1999, 341–375
Printed in the United Kingdom

Agent technology in communications systems: an overview

ALEX L. G. HAYZELDEN and JOHN BIGHAM

Intelligent Systems Applications Group, Department of Electronic Engineering, Queen Mary and Westfield College, University of London, London E1 4NS, UK (email: {a.l.g.hayzelden, j.bigham}@elec.qmw.ac.uk)

Abstract

Telecommunications infrastructures are a natural application domain for the distributed software agent paradigm. The authors clarify the potential application of software agent technology in legacy and future communications systems, and provide an overview of publicly available research on software agents used for communications management. The authors focus on the intelligent agent type of software agent, although the paper also reviews the reasons why mobile agents have made an impact in this domain. The author's objective is to describe some of the intricacies of using the software agent approach for the management of communications systems. The paper is in four main sections. The first section provides a brief introduction to software agent technology. The second section considers general problems of network management and the reasons why software agents may provide a suitable solution. The third section reviews some selected research on agents in a telecommunications management framework. The final section concludes the paper by discussing some of the problems encountered and some future directions for further research.

1 Introduction

Employing software agents for network operator user interfaces has begun to make an impact on the communications industry allowing end management units to cope with the large amounts of control information that is generated from everyday operation of networked systems (Howlett, 1997). For example, telecommunications management control centres employ sophisticated software to aid decision making on how the network system should be configured and operate (Liebowitz and Prerau, 1995). However, the revolution where multiple software entities inhabit physical communication networks automating control actions upon network elements, without having to rely on human managers, has certainly not been overwhelmingly accepted by the telecommunications industry. In this paper, we will describe some of the reasons why the adoption of software agent technology will benefit network operators, service providers and the end users, so that communications systems can be; efficiently maintained; can benefit from increased business productivity; and provide improved Value Added Services (VAS). Before we delve into the latest developments in software agents it is important to provide an introduction to some of the problems the telecommunications domain is currently facing. The next few sections will describe some of the architectures and models that have been deployed within the network management area and how some of the changes disseminating from the computing arena, particularly Internet technologies, will hopefully lead to a "high speed, ubiquitous information highway" (Decina and Trecordi, 1997).

Telecommunications is in a transition stage where the telecommunications and computing industries are not only merging their main concepts, but working towards a similar goal, namely to have fully co-operative communication systems that help automate various everyday tasks or chores. Therefore, it is quite evident that there is a trend towards combining the advantages of traditional connection oriented telephony and Internet based communications. Of course another factor in providing a more combined communications infrastructure is how to deal with the huge amount of traffic that future networks will need to support, especially with the development of large data oriented applications like video-on-demand. One method for dealing with huge traffic demands and therefore being able to curtail congestion in network systems, is to adopt a model that focuses on the dedicated provisioning of Quality of Service (QoS). Although the term QoS is widely used as a subjective measure of how well a communications channel is performing, it also has a stronger meaning in technologies such as Asynchronous Transfer Mode (ATM). For example, QoS in ATM refers to specific metrics (such as delay, loss and minimum bandwidth reservation) that a network should provide once a connection has been accepted onto the network (Cuthbert and Sapanel, 1993). Of course the ability to provide these guarantees depends on the dynamics of the network state, something which is intimately related to the types of traffic that the end users generate as part of their communication dialogue. It also generally requires the implementation of sophisticated control and signalling techniques (Quendt, 1997). The dependency of QoS provisioning and the need for complex signalling techniques often presents a major obstacle to network integration. The need for various network operators (the owners of the underlying network hardware) and service providers (the authority that provides the application services on the network hardware) to agree on a common signalling system and control protocol, often conflict in interest.

The sheer complexity of inter-working these existing heterogeneous systems meant that the computing arena chose a quite different approach, i.e., the basis of the prosperous Internet. The Internet model is based on the concept of inter-working, where the network hardware is viewed as just a transmission platform and all the functionality lies in the software functions residing on the host servers. We and others, including (Sincoskie, 1997; Nwana et al., 1997; Gaïti, 1996) believe that this model will eventually prevail as the enabling communications platform of the future. In later sections we will describe how management solutions from the telecommunications world, coupled with concepts from the Internet and software agent technology could provide a more integrated, easier to automate and more flexible service enabled communications system.

Another important factor in achieving inter-operability of transmissions networks is how to maintain co-ordination of the distributed system, i.e., network management. Network management has traditionally concentrated on the physical level by categorisation into the specific classes of fault, configuration, accounting, performance and security management, often referred to as FCAPS. However, with the advent of technologies such as, ATM (Adams, 1995), IP over ATM (Schmid et al., 1997), mobile IP [URL 2], Intelligent Networks (Black, 1997), QoS based technologies (Wang and Crowcroft, 1996), Active Networks (Tennenhouse et al., 1997) [URL 20], mobile communications and advances in the underlying digital infrastructure (Halsall, 1997), network management will need to focus more on the logical level to allow for more flexibility and improved integration. Other factors pushing the communications industry forward are: competition between an ever increasing number of service providers; move towards a service-oriented or application level approach; increases in computational power; advances in switching hardware; standardisation of both software and hardware interfaces; and the bandwagon to provide "intelligence". Decina and Trecordi (1997) provide a comprehensive overview of communications and computer based convergence and review many of the underlying technologies, i.e., CORBA ((Pavón et al., 1998) provides a description of CORBA in a telecommunications context), Active Networks, IPv6 (Schmid et al., 1997) and TINA-C [URL 1]. Focusing attention on the logical level will aid this push and also provide the mechanism for increased system automation as a key contributor to achieving competitive market advantage.

This paper is in four main parts. The first discusses the notions of what constitutes a software agent and a multi-agent system (MAS), paying attention to software engineering concepts (Wooldridge and Jennings, 1998). We will describe how the Mobile Agent (MA) community have recognised the importance of implementation aspects and centred on computational agents at an early stage of development. The second section will discuss the main factors and problems that are found in the telecommunications domain and describe the research that has addressed these issues. The third section will be devoted to surveying selected and interesting research that conforms to our definition of software agents for telecommunications (see section 2). The final section summaries some of the issues highlighted and concludes the paper.

1.1 *Software agent related definitions*

It is widely recognised that the communication infrastructure of the future is fertile ground for software agents and also more generally AI (Artificial Intelligence) techniques (Lewis, 1995). This is illustrated by Decina and Trecordi (1997) who state that "The service network of the future will be populated by software agents capable of assisting the end user in the use of the ever growing information that is accessible on-line". We do not intend to provide a comprehensive review of agent technology as this would be beyond the scope of this paper and therefore we refer interested readers to Wooldridge and Jennings (1995) for aspects of intelligent agent *theory*, *languages* and *architectures*, Nwana (1996) for a detailed discussion on background and the application domains of software agents, and Müller (1998) for the latest review of agent architectures. Similarly, Chaib-draa et al. (1992), Russell and Norvig (1995), O'Hare and Jennings (1996) and Huhns and Singh (1998) provide comprehensive details of AI and DAI (Distributed Artificial Intelligence) techniques. Much of the next few sections is our interpretation of the descriptions provided in the above references.

For the context of this paper the term "software agent" has been adopted as the most general phrase to describe the concept of a software entity that automates some of the tasks regarded as mundane or laborious to a human agent. We use the term software agent to incorporate the two most commonly described software components that researchers have used to tackle complex decomposition problems, namely Mobile Agents (MA) and Intelligent Co-operative Agents, often referred to as just Intelligent Agents (IA). Until recently, there has been a distinct line drawn between these paradigms and the two research domains have been focussing on quite different problems. However, the two research areas have been relying on a similar conceptualisation, that of utilising autonomous software processes (they may be implemented as procedure calls, a thread or several threads, but look and behave conceptually as autonomous processes) for dealing with automating control tasks and it is also evident that the more specific attributes of each are merging towards a unified approach, e.g., Greenwood's (1998) paper discusses Intelligent Mobile Agents for future telecommunications systems. Foundation for Intelligent Physical Agents (FIPA) [URL 4] is also addressing the integration of mobile abilities into FIPA's original definition of an agent (cf. intelligent message passing agent). Also, commercially available mobile agent frameworks, such as Grasshopper [URL 3] are including the provision of Agent Communication Language (ACL) support, something that has traditionally provided a distinction between the two types of software agent (Smith, 1996).

This paper will focus mainly on software agents used for the enabling of automated management and control of telecommunications systems, rather than their use as providing 'intelligent' interfaces and Internet search type agents. This paper will discuss some of the reasons why distributed software agents can provide the underpinning software technology for current legacy systems and future telecommunications systems (Nwana et al., 1997). For readers interested in a wide variety of applications of agent technology to communications systems, then we refer them to a book entitled *Software Agents for Future Communications Systems* (Hayzelden and Bigham, 1999).

A software agent can be described as an independently executing program able to handle autonomously (i.e., without direct input at run time from a human) the selections of actions when expected or limited unexpected events occur. Software agents can be of differing abilities, but typically possess appropriate "expertise" to deal with their own world (operational environment or application domain). The term agent has also been used for many years in the field of distributed computing where it refers to specific (client or server) entities used in solving specific tasks in a distributed computing system, however for this paper we will concentrate on the former kind. The abilities of software agents have been described eloquently by Wooldridge and Jennings (1995) and can be classified into the possession of social ability, autonomy, reactivity and adaptability. Wooldridge and Jennings (1995) provide a diverse review of intelligent software agent research.

Magedanz (1996) has provided a broad taxonomy of agents in the telecommunications domain and identified the following groupings; local agents, networked agents, DAI based agents and mobile agents. These classifications apply to the vast array of work that is commonly grouped under the heading of "agents in telecommunications". However, in many research papers it is in fact often quite difficult to extract an author's interpretation of the term "agent", especially in telecommunications research. For example, "SNMP agents" in Stallings (1998) and "intelligent agent" in Howlett (1997) do not strictly relate to the agent definition that is encountered in the DAI works or comply to the definition given above.

1.2 Rationale for multiple software agents

At a high level the multi-agent systems approach is intuitively simple. Developers can draw on their experience in solving problems in real world co-operation with others. This gives software agent developers an advantage when constructing multi-agent-based systems, as there is some reliance on a natural methodology for isolating the different interaction modules that make up the complete problem solving agent system. Therefore, the distributed software agent idea is powerful in allowing the software developer to gain a solid conceptualisation of problem solving required in decentralised control. The agent approach also provides an appropriate analogy for the decomposition of problems and concurrent task delegation by supporting the abstraction of some of the complexities that are gradually emerging in the telecommunications domain (Goldszmidt and Yemini, 1998).

Some of the benefits cited in the literature of the MAS approach can be summarised as:

- to address problems that are too large for a centralised single agent, for example because of resource limitations or for robustness concerns (the ability to recover from fault conditions or unexpected events);
- to allow for the interconnecting and interoperation of multiple existing legacy systems, e.g., expert systems, decision support systems, legacy network protocols;
- to improve scalability – the organisational structure of the agents can dynamically change to reflect the dynamic environment, i.e., as the network grows in size the agent organisation can restructure by agents altering their roles, beliefs and actions that they perform (Lupu and Sloman, 1997);
- to provide solutions to inherently distributed problems, e.g., telecommunications control, air traffic control and workflow management;
- to provide solutions which draw from distributed information sources;
- to provide solutions where the expertise is distributed.

Using agent-based systems promises to have advantages over centrally controlled systems when achieving multiple goals and managing multiple sensory input. As a result, network management and control architectures exploiting agent technology are perceived as an important step forward in their own right.

1.3 Agent architectures

In this section we discuss two levels of architecture granularity, this will be important when making comparisons and distinctions in section 3 of this paper. The first level of architecture granularity defines the relationships between the components that constitutes the internal composition of the agent; we will call this the "agent internal architecture". Secondly the architecture of the MAS, which defines the relationships and interactions between each of the individual agents.

Wooldridge and Jennings (1995) classify agents by the type of agent internal architecture that the agent designer adopts, the three main architectural types being reactive, deliberative and hybrid. Wooldridge and Jennings' paper provides a detailed review of research that has adopted these

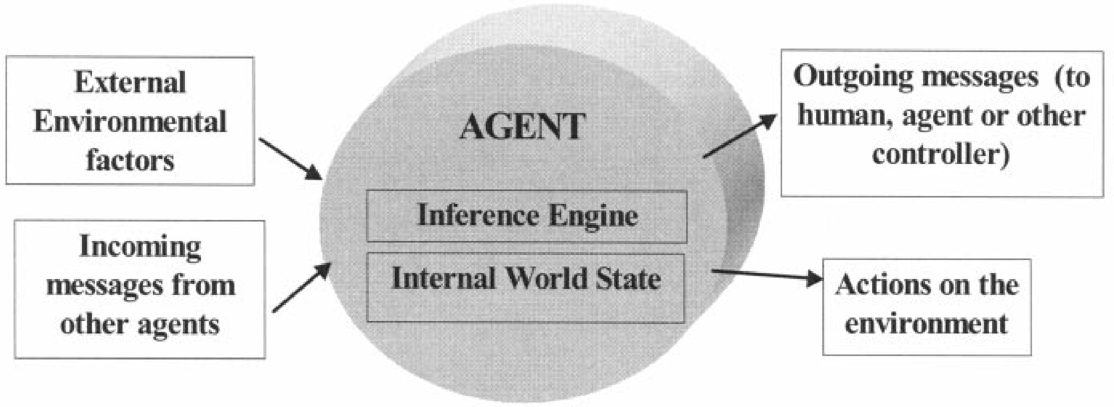

Figure 1 Agent interactions with external environment (adapted from Russell and Norvig, 1995).

different architectural approaches. Müller (1998) provides a discussion of where these different types of agent internal architectures have been applied and contrast situations where they are applicable.

Russell and Norvig (1995) describe differing classifications of agent internal architectures and also provide an excellent covering of AI techniques that have been used as an integral part of agent technology. Figure 1 shows common inputs, outputs and actions performed by a software agent.

Of course the choice of adopting a specific agent internal architecture as opposed to another internal architecture is useless unless the multi-agent system level architecture is also well defined. The multi-agent system level architecture can also be seen as the high level framework that the agents need to operate within and often constitutes the co-ordination mechanism that the total agent system should conform to. The next section will describe these system level architectures, where the importance is placed on the amount of collaboration that occurs between the agents.

As well as classification by architecture, software agents have been classified by their degree of agency. They may possess "weak" to "strong" agency. A software agent conforming to the "strong notion" (Wooldridge and Jennings, 1995) are those which reason based on mentalistic notions like belief, desires and intentions. Agency, which typically do not have "strong agency" (e.g., generally mobile agents) are more concerned with aspects of asynchronous task execution, mechanisms for hardware independence and the ability to distribute intelligence. Section 1.7 will describe mobile agents and their use in telecommunication systems.

1.4 Open systems and heterogeneous multi-agent systems

As mentioned previously, a central MAS concept is that of a society of agents sharing information. The problem solving that the MAS can achieve is often determined by the amount of collaboration that individual agents actively participate in and to what extent information is shared. The degree of co-operation or collaboration of the agency is generally governed by the adoption of a particular agent interaction strategy. There are three main interaction strategies that have been discussed in the literature, namely co-operative agents, self-interested agents and hostile agents. Self-interested behaviour is most appropriate when the agents are developed for competing parties operating in an open environment. Co-operation is often essential in critical system control as the equilibrium operational state of the system must be explicitly known (i.e., network fault control, nuclear power station control, etc.). The relevance of the different kinds of agent to network management is now illustrated.

Co-operative agents

Co-operative agents work together with the intention of solving a joint problem and can be an appropriate model for network management when viewed from the perspective of a single (usually large) network provider. A co-operative agent is prepared to suffer loss of individual utility so long as this is exchanged for the common good.

Self-interested agents

The introduction of independent service providers for example, means that automated negotiation systems with self-interested agents (i.e., those that try to maximise their own good without concern for the global good, and will only perform services for another agent for compensation, in terms of money for example) will become important in network management. The increasingly standardised and integrated communication infrastructure (see section 2) is making such interaction possible and arguably necessary. Such interactions have been the subject of analysis in microeconomics and game theory, but recent work has investigated the more practical case where computational resources are bounded and solution quality is traded for computational cost. Work in the area of electronic commerce (Sandholm and Lesser, 1995b) has also investigated exchange of goods (which in the context of network management could be bandwidth) for self-interested agents. The proposed method incorporates exchange that ensures at all times, (for both the supplier and the demander) future gains from carrying on with the exchange (i.e., co-operating according to the contract) are larger than the gains from defecting. This can be important for computational agents managing exchanges that cross national boundaries, as different laws may be applicable and whatever the law might be, it could be difficult to enforce.

Hostile agents

There is also the notion of hostile agents, which can be modelled by an agent which has a utility that increases with its own gains and also increases with competitor's losses. Whether this is an important class of agent in the network management area is not yet clear. In any case, the authors are not aware of work related to network management exploiting hostile agents.

There has been a considerable amount of work on distributed planning, scheduling, resource allocation and control problems for co-operative agents. The principle reasons for distribution have been either,

- to reduce communication costs associated with a central problem solver, or
- to gain speed and computational resource through parallelism, or
- to gain reactivity through not needing to consult a central agency, or
- to gain robustness through lack of dependence on a single computational node.

However, it is becoming understood that the case for distribution of problem solving in the co-operative agent case is not always strong (e.g. Sandholm and Lesser, 1995a), "to date, the arguments in favour of distributed co-operative scheduling systems have mostly been weak". The arguments certainly need to be examined in the context of network management problems. Let us first look at an example where distribution is effective. Take a simplified form of call admission control and a network service provider. If the admission policy problem is formulated as that of minimising utilisation (say) and we define utilisation of the network as the negation of the sum of the link residuals in the network (say) subject to the constraint that the residuals cannot be negative, then it is easy to show that the policy of allocating to the path with the smallest hop count, when feasible, is equivalent to applying a gradient based iterative approach for convergence to best utilisation. The same is true for different measures of utilisation. This can be used to provide a simple distributed allocation policy for the reactive agents sited at the entry points to the network. When a new connection request comes along we simply allocate to the path with the smallest hop count that also has adequate bandwidth. The distributed agents acting autonomously in the manner described interact to provide reactive control that normally converges to the optimal utilisation. The global optimal solution is in fact not guaranteed as a local optimum could be found. At a reactive level when controlling admission to the network there is a strong argument for distribution here as we are in fact obtaining all four advantages cited above (The problem of establishing adequate bandwidth makes the problem more complex, but the principle still holds). More strategic levels of control or planning often require more co-ordination.

In planning and scheduling much of the complexity is associated with the combinatorial nature of the problem. In distributed search, it can sometimes be argued that only abstractions of detailed local information need be passed between the distributed agents, rather than the detailed raw sensor data. So we may achieve co-ordination with only little bandwidth used for inter-agent communication. Counter arguments say that sometimes these abstractions introduce approximations, which makes the search less efficient. However, separate from such arguments, the communication between the distributed co-operating agents can still exceed that required if all the information were passed to a central problem solving agent. The large amount of communication can arise because the solution is some kind of search, messages may have to be sent several times and the distributed agents often need to know state information. The robustness argument often used to support distribution, can often be handled by replication rather than distribution.

Now let us illustrate a problem in the network management plane where distribution of co-operative agents is less effective. Suppose we want to optimise planned network flow under projected demand patterns at entry nodes of the network, using some appropriate criterion for the management plane such as tactical survivability, as defined for example by Bigham and Hayzelden (1998). This is a resource allocation problem that can be formulated in terms of a single central problem solving agent. For example, it can be formulated as a non-linear programming problem. It is also possible to distribute the computations of the desired changes to improve network survivability to co-operative autonomous agents situated at the nodes of the network and derive behaviour for each agent that should normally allow convergence to the global optimal planned allocation. This leads to a robust solution. If a node goes down the processing can still continue effectively. However, the distributed computational agents (there is one for each source destination pair) must know about the residual capacities of the network relevant for its source destination pair and so a lot of inter agent communication is required as each agent updates the plan. Alternatives such as replicating the whole calculation at different nodes and in the extreme at each node are more attractive. Robustness is then also assured, and furthermore a simple voting mechanism based on the value of the survivability metric makes it less likely that the solution is a local maximum.

1.5 Co-ordination in multi-agent systems

Problem solving performed by agents in a MAS is a form of Distributed Problem Solving (DPS) and this involves co-ordination, negotiation and communication. For a MAS to solve common problems coherently, the agents must share information about their activities, normally via explicit communication. They must also co-ordinate their activities or pursuits (effectively schedule their actions) and when conflicts arise they must be able to negotiate. Conflicts can result from simple limited resource contention to more complex issues where the agents disagree because of discrepancies between their domains of expertise. Co-ordination is thus required to determine organisational structure amongst a group of agents. There are a number of mechanisms to enable co-ordination, including structuring the agent system into hierarchies of authority (Nwana et al., 1996), limiting flexibility by pre-enumeration of actions and using high level finite state communication models to check for conversational consistency (Barbuceanu and Fox, 1995). There is a large collection of papers devoted to the co-ordination of socially interacting agents and we refer the reader to Jennings (1996) and Nwana, Lee and Jennings (1996) for a discussion and references. The following section will only discuss the co-ordination mechanisms that have been widely implemented and have been utilised in telecommunications problems.

The most influential co-ordination technique for task and resource allocation among a society of agents is the Contract-Net Protocol (CNP) (Smith and Davis, 1981). Its generic nature and robustness is acknowledged by its adoption into the FIPA (Foundation for Intelligent Physical Agents) specifications [URL 4] as a high level co-ordination mechanism for allocating tasks or resources within an agency. We have inserted this description of the CNP mechanism as some of the papers that we will review in section 3 have used it as the main co-ordination construct.

In the CNP a market structure is assumed and agents can take on either of two roles, a manager or a contractor. The basic premise of this form of co-ordination is that, if an agent cannot solve an assigned problem using local expertise, it will decompose the problem into sub-problems and try to find other agents with the necessary resources and expertise to solve these sub-problems. The assigning of the sub-problems is solved by a contracting mechanism. This consists of the manager creating a contract that is announced to other agents in the system (effectively the contract is broadcast). The contract agents each submit a bid for the contract on offer. The bids are evaluated. The contracts are awarded to the contractors with the most appropriate bid. If the bid is accepted the contract agent cannot revoke. The CNP has been used in many applications, such as Busuioc (1996) and Weihmayer and Brandau (1990) for service level network control.

Some of the CNP advantages include the ability to allocate tasks dynamically and to allow agents to be introduced and removed at will. Its limitations involve the fact that it does not detect or resolve conflicts and that the broadcasting of messages can possibly outweigh its advantages, particularly in a network management environment.

In section 7 of the FIPA'97 specifications they take as a reference application domain, network management and provisioning. Griffeth and Velthuijsen [URL 18] have suggested a negotiation protocol for a telecommunications service provisioning environment as a proposal to FIPA'97. This negotiation protocol builds on aspects of the CNP but incorporates an approach that keeps the number of service combinations manageable and interestingly also allows counter-proposals. This avoids the user having to be familiar with the sometimes complex nature of interactions between "features".

1.6 Software agent communication

Agents can pass messages to each other informing others of changes to the environment or information about "plans" to change the system's state, but this generally occurs at a higher level of abstraction than traditional software remote method calls. For example, agent communication normally involves the passing of information at the knowledge level (constructs of natural language syntax which has a high level meaning). Communication can occur via dynamic mechanisms such as shared memory, IPC facilities or via static mechanisms such as file system locks. There are two main categories of agent based communication. Proprietary Agent Communication Languages (ACLs) and Standardised ACLs. There are of course more specific subcategories for agent communication mechanisms, e.g., plan passing and stigmergy (see Schoonderwoerd, Holland and Bruten, 1997).

Communication enables agents in a multi-agent environment to exchange information (e.g., to allow an agent to inform another agent about its current beliefs, amount of resources available, additional information about its environment, etc.) on the basis of which they originate their action sequences and co-operate with each other. Software agents generally communicate with other agents in order to work more flexibly.

Most ACLs derive their inspiration from speech act theory (Searle, 1969), which was developed by linguists in an attempt to understand how humans use language in every day situations, to achieve everyday tasks, such as requests, orders, promises, etc. In multi-agent systems, the possible solutions to the communication problem range from those involving no direct communication (e.g., Appleby and Steward's BT mobile agent work), those involving ad hoc ACLs (Davison, Hardwicke and Cox, 1998) to those involving standard ACLs (Cohen and Levesque, 1997).

Proprietary ACLs have been chosen as the communications mechanism by most researchers, this is probably due to the designer's application requirements being specific and due to the lack of standardised ACLs being available. Priest and Siani (1997) have demonstrated the importance of the careful choice of an agent communication protocol. They demonstrated that in a multi-service provider situation the complexity of the message passing protocol could have severe impacts on the efficiency of the bandwidth utilisation of the physical communications system.

A standardised ACL is essential if agent-based network management systems are going to achieve their integration into legacy systems and inter-operate with future demands for regular network system upgrades. The FIPA organisation is attempting to define such standards. The FIPA ACL is mainly based on KQML (Knowledge Query and Manipulation Language). The KQML language can be viewed as consisting of three layers – content, message and communication layers (Labrou and Finin, 1998). The content layer specifies the actual propositions of the message (i.e., a fact). At a second level KQML provides a set of performatives which constitutes the message layer (e.g., ask, tell, reply, etc.). This layer enables agents to share their intentions with others (Barbuceanu and Fox, 1995). The communications layer defines the protocol for delivering the message and its encapsulated content. A simple construct example of a KQML message might be (ask <agent1> (allocate 200 Mbits of bandwidth)). More information about KQML specifications can be found at [URL 5].

KQML defines the specific set of performatives that can be sent between agents that support KQML. KQML defines the format of the messages. However, the contents of the messages should have some real world meaning to the agent recipient. It is therefore a common requirement that each agent engaging in a communication dialogue must specify a common database of terms and their associated real world meaning, this is called an ontology. An ontology gives meanings to symbols and expressions within a given domain language. The ontology allows the mapping from syntactical content (as in the simple construct example given above) to some well-understood meaning. For the communications between two or more agents to be useful each agent should use the same ontology (implicitly or explicitly), without it, co-ordination cannot be achieved.

However, KQML has a number of shortcomings, the most important of which is its lack of precise semantics. Cohen and Levesque (1996) believe this to be a real problem: "without a precise semantics, agent designers cannot be certain that the interpretation they are giving to a performative is in fact the same as the one some other designer intended to have". This problem has also been identified by FIPA. Therefore, FIPA have developed an ACL that reduces the number of supported performatives and in some respects defines semantics more precisely.

However, both KQML and FIPA ACL both lack well grounded semantics at the communication level and this can lead to problems, including the difficult task of "semantic verification" (Wooldridge, 1998). The ability to verify semantics becomes very important when developing large control systems based on ACL communication. One important point that FIPA ACL has tried to address is that of what to do when an agent X sends a request to agent Y, which does not know how to interpret the message content. FIPA specifies a performative "not-understood". This helps prevent deadlock situations where agent X is waiting for a response from agent Y when agent Y does not know how to respond. In fact to be FIPA compliant the implementation of the "not-understood" performative is mandatory.

Timing of information exchange is also of great importance. One consideration that must be addressed when designing and implementing a communicative society of agents, is that of message delay. Horikawa, Aida and Sugawara (1996) have studied the problems associated with agent message propagation delay. Propagation delay has been widely addressed in the telecommunications domain (Halsall, 1996), but the implications are exacerbated when considered within an agent context. For example, not only does propagation delay mean that network control can be delayed or even prevented, but agent systems are difficult to design and implement when the uncertainties of message delay are incorporated. Pitt and Prouskas (1998) have discussed some of these issues, including a discussion and methods for synchronising clocks for multi-agent interaction in a network load control context.

1.7 Mobile software agents

Until now we have focused on the intelligent agent paradigm. However, in the telecommunications arena placing new or pertinent control logic or additional functionality into a communications network is an extremely important factor, especially as it potentially allows for outdated systems to be updated in an efficient manner. Adopting this concept can provide network operators with a mechanism for dynamically updating switch control logic and performing other updates to system level functionality. As we will consider, there are various methods for achieving dynamic updates to process control, e.g., Active Network technologies (see section 2.6). Another method that conceptually extends the ability to deploy mobile code is the adoption of mobile agent technology. Mobile agent technology has been the subject of much attention in the last few years due mainly to the proliferation of technologies for platform integration, such as JAVA RMI (SUN, 1997), CORBA (Pavón et al., 1998), middleware (Pitt et al., 1997) and the increased availability of powerful processing hardware.

Mobile Agent (MA) technology was originally concerned with the ability to move executable code from one computer to another. The main benefit from adopting this approach was that available computational resources on another computer could be utilised when resources on the original computer became scarce. Even though the principles are quite intuitive the actual implementation led to many unforeseen problems such as maintaining the location of the software code in the network system, enforcing integrity and the more complex considerations of ensuring security and how to convert executable code to a generic platform independent format. Some of these points were considered relatively early in the evolution of the mobile agent history and before long, some interesting capabilities were realised. Researchers became more interested in dealing with the idea of moving so called "intelligence" from one place of execution to the next.

The impact that MA technology has had on the commercial world is conveyed by Nwana (1996) who mentions that "public perception of agent technology is synonymous with mobile agents". Since the publication of Nwana's overview paper the development in the mobile agent world has proliferated beyond belief. Whilst, Telescript (a mobile agent framework developed by General Magic) which Nwana (1996) cites as the "current leading mobile agent operating environment" is no longer supported (a more usable system called Odyssey is now available from General Magic [URL 14]), there are many mobile agent toolkits and frameworks now available, many of which are implemented in the Java language.

The concept of mobile agents further blurs the distinction between client and server type models, however it still remains the case that passing of messages is often more efficient than moving an agent (the executable code) from node to node and therefore, there has to be careful analysis of the benefits that each approach can provide in the particular application domain. However, since mobile agents provide a suitable mechanism for dynamically placing new control software within a telecommunications environment, its adoption will have an impact on the actual network architecture and the related signalling protocols of the telecommunications system (Quendt, 1997; Magedanz, Rothernel and Krause, 1996). For example, the metaphor itself – mobile, encapsulated programs provides a new framework for thinking about computer networks and the modularity of mobile agent approach encourages the design of more flexible and adaptive network architectures (Magedanz, Rothernel and Krause, 1996).

MAs present a particularly interesting capability when coupled with the problem of having a non-continuous connection to networked resources. This situation becomes more pertinent as users are becoming more mobile and the increased use of laptop computers. We will discuss some of the requirements for mobile users in a later section. However, to summarise some of the advantages that MA can provide to this situation is that the MA can be created off-line and uploaded to the network (i.e., via say a wireless communications link) and dispatched so that it can execute on a network computer (server) to carry out some of its tasks. Later the mobile user can re-connect to the network and re-call the agent, hopefully with some useful information that it has managed to obtain. Sahai and Morin (1998) provide a more detailed description of this kind of scenario and describe how the MAGENTA framework overcomes some of the encountered problems. Figure 2 shows an agent being created at a Personal Digital Assistant (PDA) and being released into the fixed network via a temporary communications link, say over a wireless link.

Another example where mobile agents may be able to achieve an advantage over the message passing agent approach is when interaction between two nodes occurs, where the nodes are geographically far apart. Let us assume that node A (a management node) wants to re-configure a

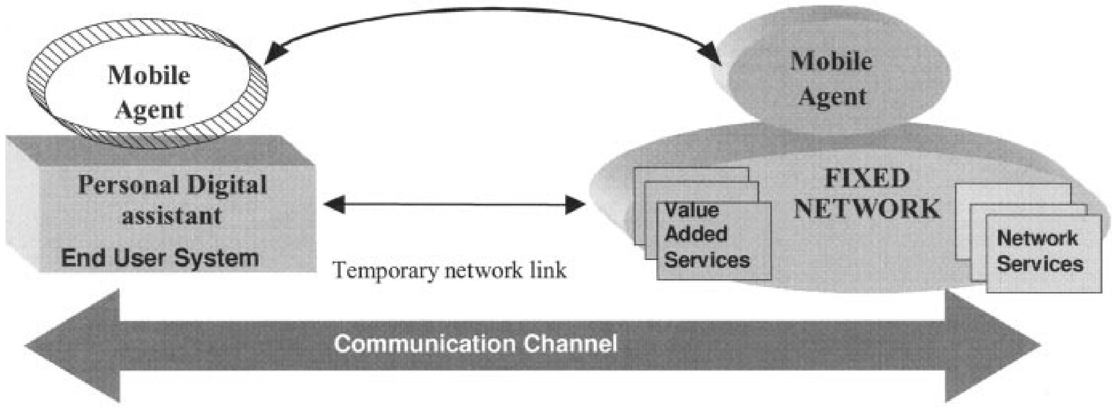

Figure 2 Mobile agent being released into fixed network.

node X by updating some of its control logic. In this situation, it may be possible for node A to send messages to node X to achieve the update, similarly it may be possible for the whole new control logic to be downloaded to node X and assuming that node X can accept the new logic this may be the most efficient mechanism. However, in this type of situation it is often the case that contingencies occur, for example node X requires additional data or a certain piece of operating system software that is not included in the new logic update. We can therefore envision situations where it may be more beneficial to have an "expert" agent move to the actual node and carry out the task of upgrade. Being specialised in carrying out this task the agent would be more likely to deal with contingencies that may occur. This approach is also more beneficial in terms of reducing bandwidth consumption between node A and node X, often more rapid (the sending of messages back and forth between the two nodes incurs propagation delay in both directions) and more encapsulated (the mobile agent has some knowledge of how to deal with contingencies of completing the task). Another advantage that can be obtained from using an expert mobile agent is that there is no need to have active processes at the two nodes (i.e., using CPU resources and having reserved bandwidth communications channels open), again this generally reduces network resource utilisation (White, 1997). Aglet technology (IBM Japan) [URL 8], has re-opened the interest in mobile agent technologies and provides a Java implemented autonomous mobile software agent technology. Aglets extend the model of network-mobile code by allowing the class files to maintain state when migrating from node to node. Aglets still rely on the presence of a Java Virtual Machine to execute their state and actions, so that security breeches are limited (Venners, 1997b).

A major concern with MA technology is the lack of security on the agent side of the relationship. It is often the case that the agent requires a specific piece of software to gain access to the resources on a host computer. In the IBM Aglet framework this software is called a "place" and provides some security mechanisms to prevent the server from being violated by the executing agent. However, there is considerable concern about users delegating confidential information (such as credit card details) to their MA and releasing them into an open network system where, without proper security considerations, server computers could gain access to confidential information (Venners, 1997a). This can be described as agent violation.

Mobile agent overview papers by Morreale (1998) and Harrison, Chess and Kershenbaum (1995) provide some interesting applications for mobile agent technology. Similarly, Green et al. (1997) contains a section on mobile agent technology that describes some of the mobile agent platforms that are currently available. For a mixed intelligent and mobile agent perspective on agents in communications systems we refer the reader to Hayzelden and Bigham (1999).

The widespread acceptance of the Java language and its ability to provide platform independence for the mobile agent technology and some clever innovations from MA researchers will surely mean that service providers will adopt the concept for use in all kinds of network applications, particularly in the Internet domain.

2 Communications systems

Following the discussion on some of the concerns, benefits and high level concepts that the software agent approach is constructed upon, we now turn our attention to the more specific problems that occur in the communications domain. We will describe some of the background and problems that are encountered in this domain and some of the solutions that the software agent paradigm has provided. We begin this section with a general description of some of the issues and complexities involved in network management and control.

2.1 General network complexity

Networks range in size and complexity from the small in-house local area networks (LAN) to the millions of nodes which make up the Internet. The larger the network, the more heterogeneous it tends to be. As network complexity grows, traditional embedded network management approaches will be unable to cope (Fuggetta et al., 1998), particularly when they scale to world wide communications networks. To tackle this problem the International Standards Organisation (ISO) developed the Open Systems Interconnection (OSI) standards, which are based on the idea that a manager-agent approach should be adopted for the management of complex network systems. The principle idea is that, if this approach is adopted some of the interfaces can be standardised and therefore reduce some of the network heterogeneity.

Network management in general comprises both network monitoring and control. A network management system thus performs, among other things, the tasks of network configuration, guaranteeing its proper functioning, maintenance, security control, collection and archiving of data and fault management. See Stevenson (1995) for an introduction to network management.

Extending some of the concepts of the OSI network model, Network Management Systems (NMS) have encompassed the ideas of task delegation. Here, management tasks are sub-divided amongst several entities, each being assigned specific tasks, which allows the distribution of processing and the reduction of bottleneck traffic. Unfortunately, many existing network management architectures are based on outdated concepts and due to the cost of re-implementation have not tracked some of the latest developments that enable things such as full distribution, conflict resolution and advanced software development techniques, such as OO design and of course agent technology. However, TINA-C has made significant progress towards supporting modular and flexible inter-working and in some respects replaces the OSI based communications by the implementation of a ubiquitous Distributed Processing Environment (DPE), thus in some ways conforming to the Internet type model. As such, it should be seen as a long-term architecture [URL 1]. The services provided by the DPE allow object interactions to occur via well-defined interfaces. Complexity transparencies are obtained over underlying network platforms and the typical transparencies are location, distribution and replication. This suggests that the TINA DPE is acting in a similar way to the CORBA architecture, which it is, but the CORBA architecture does not fully exhibit the requirement for this type of environment and lacks characteristics such as the availability of object stream interfaces. In TINA the DPE is implemented through the technique of creating a standard Application Programming Interface on all the concerned hardware. This idea is similar to running Java Virtual Machine software on all computer operating systems, and therefore demonstrates how the paradigms from each domain are converging on a generalised end-point framework (Nahrstedt and Smith, 1996).

2.2 Users and the need for mobility

User habits are also beginning to impact on the way new telecommunications networks must be managed and controlled. For example, the need for end users to move their communications devices (PDA, Laptops), to change their place of information gathering, point of banking etc, means that the equipment required to access these services must adapt. The list below defines some of the concepts that are beginning to emerge, as user mobility becomes a decisive factor influencing some of the changes taking place in telecommunications infrastructures.

- *Terminal mobility*: terminal equipment is identified by a unique terminal identifier. Equipment has a unique identification and is also independent of location. This idea allows the terminal equipment to be re-located and still be identifiable.
- *Personal mobility*: independence of network point of attachment and specific user equipment. A user has a unique identification. This enables "hop-swaps", where users can utilise different devices (e.g., laptops) taken from a common pool of resources and still have the information sent to them (Rizzo et al., 1995).
- *Session mobility*: allows an end user to be continually involved in a service session independent of the equipment. This provides the ability to move say an active spreadsheet session from UNIX to Windows environment transparently and without session termination.

Changes in user behaviour, such as new personal working habits, working profiles and place of connection, are all factors that are likely to impact on how the existing telecommunications paradigm will change. Why have a telephone number that identifies particular telephony equipment and location? It would be much more flexible if telecommunications network operators provided a personal identification mechanism. For example, like a user having a World Wide Web (WWW) homepage, you could search for a person who is interested in your hobbies and then contact them through a hyperlink type mechanism. The concept of Universal Personal Telecommunications has recently gained more popularity because it allows mobile users to be reached via a personal contact number. Calls made to a personal number are routed to the person's current location. As described in Rizzo et al. (1995), more sophisticated location mechanisms are possible, such as the use of a badge activating certain communications equipment in the vicinity of the mobile user.

Session mobility may be the concept that proves most essential to users changing needs. Envisage the ability to read your email on your work computer (say a UNIX workstation) and still be able to continue with the same session when you return home to your PC. This is a problem scenario where the agent approach is of the essence. The ability for an agent to collate the users preferences, types of computer equipment and negotiate with other agents that have the capabilities to convert the session format to the required format will give the user a transparent transition from one session to the next. Some research on the preliminary requirements for this sort of scenario are the relatively new proposals for Mobile IP [URL 2]. The principle behind Mobile IP is that, agents located at a key node of a sub-network allow an individual node (laptop computer) to roam the Internet space. The agent reconfigures IP address routing when the node is reattached to a different sub-network. This allows for the forwarding of appropriate information (e.g., email) to the new location [URL 2].

2.3 Centralised or distributed management

There are basically four different high level architectures for the control of complex domains, centralised, hierarchical, distributed local control and a hybrid architecture (Meyer et al., 1995; Lewis, 1995). Each architecture has advantages and disadvantages, but often one architecture has more applicability in a particular sector of telecommunications.

In the telecommunications domain, the decision on whether to have a centralised, hierarchical, distributed or hybrid network management architecture is a decisive factor on whether distributed agents will be effective or even appropriate. Rapid growth in the flexibility, diversity, and complexity of network systems has fuelled the debate over whether a centralised paradigm is scaleable. Unfortunately, polling (passing monitoring information, control information or acknowledgements) from switch to management station imposes a large overhead in the physical layer communications mechanism. Relying on the centralised approach also means that the system is more vulnerable to system collapse when failure occurs, i.e., there is no localised control policy in action and all control actions occur via the central management system. However, centralised management provides an effective mechanism for the instance where a small redundant number of entities require control activation. This is because the centralised approach has benefits in terms of providing global network state information, therefore reducing the probability that inconsistency or conflicts will arise. Of course, the benefits of this must be weighed against the need to provide scalability.

An obvious disadvantage of the (completely) centralised approach is that it does not allow control data to be processed where and when it is appropriate to do so. To enable application and network service diversity further points need to be addressed including, whether there is the need for global information to be disseminated to local control points, the need to provide common management interfaces, the need to reduce the expense of geographically spatial computation power.

However, the control of telecommunications networks is inherently distributed, the management system requires the ability to deal with run-time exceptions and be robust under these circumstances. As increases in service complexity arise (particularly multi-media networks), greater amounts of information need to be processed and partial system failures require rapid rectification, therefore network management is becoming increasingly difficult to handle by global centralised mechanisms (Chan et al., 1997). Having stated these facts, having distributed localised expert problem solvers can help automate the management process. For example, by:

- Reducing communications overhead between the management system and the control activation points via a reduction in the need for continuous system state polling and signalling, i.e., higher level communications language is used.
- Distributing tasks to the most appropriate and available computational resource.
- Improving robustness, both in terms of fault recovery (less vulnerable to global system collapse when isolated failures occur) and in dealing with uncertain information.
- Improving response time – localised control can achieve more rapid responses to local exceptions.
- Using AI techniques to deal with situations where there is a lack of knowledge about how to develop an optimal algorithm (i.e., when there is no well defined procedure or algorithm to deal with exceptions)
- Improving flexibility in encompassing changes (i.e., extensibility to provide new services). For example, the introduction of new complex services such as multi-cast video-on-demand.

Skarmeas and Clark (1996) describe a system where agents communicate and co-ordinate their activities using a central message-board. Hence, co-ordination is achieved via the master/slave approach. Advertisements for services (e.g., routing) are posted to the message-board, which then forwards (or broadcasts) them onto the relevant parties. Bidders respond directly to the sender (not through the message board), who then select the best offer. Again, this is the coupling of a central co-ordination centre to fully distributed control agents, where the robustness of the latter are dependent on the functioning of the former.

2.4 Software agent-based network management

Network management is often used to refer to the mechanism that acts in a global manner on many entities and tends to move the system configuration (e.g., reallocation of system resources) from one equilibrium state to another. Management operations normally act over relatively longer time scales. Control is about providing localised fine-grained actions necessary to maintain the system in a stable condition (Davison and Azmoodeh, 1997). However, the distinction is becoming blurred (IATA 1996) with the advent of mobile control entities and localised agent expertise enabling adaptation of node control logic and distributed co-ordination, e.g., where fine grained alterations to node functionality is invoked from a geographically distant location.

End-to-end service management and resource provisioning across multi-vendor heterogeneous environment is now mission critical for carriers in the competitive open market for network services. Agent researchers have focused upon the integration of control and management and have applied economic, and game theoretic concepts to resource allocation. Economic systems constitute one well-studied class of mechanisms for allocating resources among distributed decision makers (Clearwater, 1996). We will elaborate on the use of economic frameworks for resource allocation in section 2.7.

Resource allocation is a difficult problem to address in network management, primarily because network operational states are so dynamic in broadband technologies. Whilst, much research work has been conducted on optimal strategies for connection admission control (Pitts and Schormans, 1996), route topology or optimal capacity configurations, there is very little literature on the problems associated with successive topology reconfiguration requiring co-ordination of different aspects of network functionality. Hardwicke and Davison (1998) have conducted extensive research into ATM performance management using a peer-to-peer agent oriented approach. In contrast, Sommers (1996) and Frei and Faltings (1997) have chosen to rely on a hierarchical decomposition of the agent system framework, and Hayzelden and Bigham (1997) a layered approach. Gaïti and Boukhatem (1996) have chosen a co-operative agent approach for dealing with congestion control in ATM networks. They use agents located at switch buffers to provide feedback information to the source node agent so that connections that are additionally added to the network do not add to the congestion being obtained on that region of the network.

Although allocation of resources is a major factor to consider when applying agents to telecommunications there are many others, including:

- What communication does the agent have with other agents – try to minimise inter-agent communication so that action resolution is reached more rapidly.
- Where should the agent(s) be located (the agent location granularity)? One agent per node, one per physical link, one to represent each connection?
- Should the agents be stationary or mobile?
- What view of the network does the agent have (the agent control granularity)? Is it beneficial to have the agent aware of other indirectly related events occurring in the agent's locality.
- What degree of dependency should there be between the agents in the system? If an agent "dies", is the system still robust?

Agent researchers have considered some of these matters, and section 3 will describe some of these works. However, there is still greater need to focus on the relationships each of these factors impose on one another. The proceedings of IATA (1996), Albayrak and Garijo (1998), and books by Albayrak (1998) and Hayzelden and Bigham (1999) are texts devoted explicitly to the factors mentioned above, and describe different approaches and implementations of "software agents for communications systems". Weihmayer and Velthuijsen (1998) and Plu (1998) describe aspects of agents in telecommunications from applications to software engineering perspectives, respectively.

FIPA has defined an informative specification document that addresses the issue of agent technology in network management and network provisioning. FIPA define a scenario where the importance is placed on combining the service elements from different network providers in order to obtain a single end-to-end service for the user. FIPA states that "Intelligent Agent technology is promising in the sense that it will facilitate automatic negotiation of appropriate deals and configuration of service at different levels" (FIPA'97 Document 7). The particular scenario in the specification uses co-operating specialised negotiating agents to generate a dynamic Virtual Private Network (VPN). There are three levels of agent granularity in the scenario: an agent for the representation of each human user, known as the Personal Communications Agent (PCA). The second type of agent, the Service Provider Agent (SPA) represents the interests and goals of the service provider and of course the Network Provider Agent (NPA). The process of setting up a user-defined service is initiated by the setting of some user defined constraints in the PCA, e.g., the quality of service (QoS) required, the maximum amount the user will pay for that particular service and the duration that the service is required for. The PCA negotiates with the various SPAs to determine the best deal available in terms of QoS and cost. Negotiating here is implemented as the agents passing FIPA ACL messages to one another. There is a specific VPN ontology specified that maps the agent ACL messages into the control parameters. See FIPA Document 7 for more information on the syntax and semantics [URL 4]. The negotiation mechanism continues with the SPAs dealing with the NPAs until a suitable configuration is obtained. It is, of course, necessary for the SPA to map the negotiated contract into network specific configuration settings and this is dependent on the bearer communications infrastructure (e.g., Internet, ATM, Frame Relay, etc.) that the VPN is being established on. Importantly in FIPA, some attention is paid to security issues, such as the how to prevent other competitive NPAs from discovering the high level policies of the organisation. In addition, exception instances and the policy for re-negotiating a new contract under physical network failures are mentioned, but are not a specific concern.

2.5 Service management

Telecommunications users wish to be provided with customised and diverse services using a variety of terminals and connected to the network of their choice. Therefore, despite service diversity on individual network platforms, competitive service delivery platforms are forced to interconnect. Distributed software agent technology provides one candidate solution for the required intelligent service management and integration techniques necessary to achieve this. Busuioc (1996) has indicated how agent technology can be applied to the telecommunications domain so that the services supplied by individual service providers can be coupled to give the user the combined services that they require. Individual services (e.g., video transmission capability, hotel booking service, telephony) combined together to generate a new service is often referred to as Value Added Services (VAS). An example of a VAS might be where the user wishes to have a video link, combined with a data connection and also a connection to a database with appropriate information about hotel accommodation.

Busuioc's (1996) paper describes agents that "wrap" the protocols that make up the individual services. Customers send service requests to their individual Customer Agents (CA), who then attempt to configure the value added service by sending a request to a number of relevant service agents (SA). These SAs negotiate with the service supplier agents (SSA). The SA selects the SSA with the highest bids that can provide the service content which best satisfies the CA's original request. A fuller description is given in section 3.

Ruzzo and Utting (1995) also consider how multiple service integration could be achieved through the use of software agents. They define a User agent that gathers preferences for the types of services they require, i.e., at what time of day they need the service, how long they require the service, etc. The User agent therefore negotiates with the service provider to seek a mutually acceptable service-level agreement. The User agent tries to gain the 'best deal' for its authority by trying to provide the required services at the cheapest price.

2.6 Current telecommunications management platforms

Until now we have constrained the discussion to that of how agents interact via co-ordination mechanisms, how they can employ communications protocols and how they can disseminate information and knowledge. However, there must also be some supporting infrastructure that the agents can operate in. Figure 3 shows the creation of a platform-independent agent environment generated by the execution of, say Java Virtual Machines at all node hardware systems. The message passing between the agents is depicted and the relationship to the underlying communication system is shown. The creation of a DPE provides physical communication transparency to the agents, e.g., via Java RMI or CORBA.

It is also important to realise that the choice of an appropriate agent environment can be a difficult decision to make and also means the difference between obtaining any gain in adopting the agent approach. A chosen platform must also integrate the agent support environment with the current telecommunications platforms, both in terms of system level architecture and interfacing.

The current telecommunications management environment is based on international standards, such as Intelligent Networks (IN) and Telecommunications Management Network (TMN). Both of

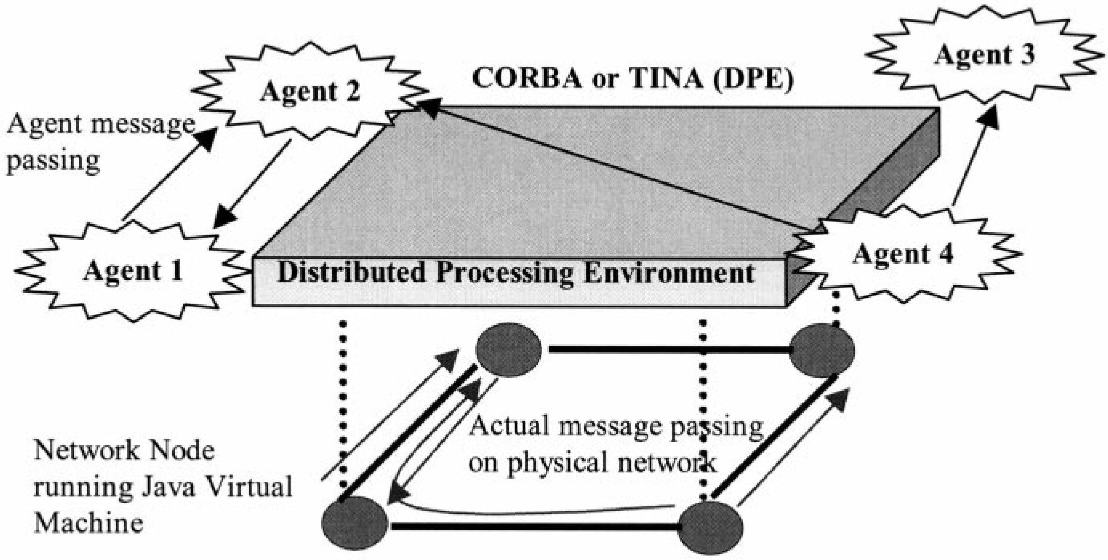

Figure 3 Agent operational environment in relation to underlying communications infrastructure.

these infrastructures can be seen as a stepping stone from traditional hardwired centralised management and implementation dependent service provision, (traditional telephony) to a more flexible and less restrictive framework that allows for management and control transparency. Although these two environments have advantages in their own right, they are both fundamentally conceived upon centralised control approaches and still rely on basic forms of the client/server paradigm.

The ITU's acknowledgement that management and control aspects of communications needs to be logically distinguishable has led to the most commonly used network management architecture available today, Telecommunications Management Network (TMN). The ITU M3010 standards describe the standard components that can be used to manage a network based on TMN. It should be noted that TMN is a semi-distributed architecture, where large parts of functionality reside on management nodes. These TMN nodes can be arranged hierarchically to avoid bottleneck situations. However, the main point is that TMN in hierarchical in functionality also. In essence it consists of functional layers including, business management, service management and network management. A TMN platform provides a common infrastructure enabling the management to communicate and facilitate their development through common building blocks and generic facilities [ICM Del 19].

Magedanz (1996) has identified some of the potential impacts that mobile agent concepts will induce on the evolution of the IN concept thus increasing the flexibility of service choice at run time. Similarly, Rizzo et al. (1995) have chosen an intelligent agent approach for the enhancement of IN to allow more flexibility in the services that the end subscriber can obtain. Rizzo et al. (1995) define two levels of agents. At one level, the User agent is concerned with negotiating the service requests for its subscriber by logging their preferences and their policies (e.g., no calls after 19:00). For the agent model to encompass the ability to support mobile users (generally users do not want to be coupled to specific end user equipment) they define another layer of agents that represent the end-point equipment (e.g., a telephone or a multi-media terminal), such that the subscriber cannot obtain capabilities that the end-point equipment does not support. For example, there would be no purpose in receiving a video message on a normal telephone.

With the advent of new telecommunication architectures which approach the communications industry from an object oriented viewpoint (and possibly later agent oriented design) there is a future potential for third parties to develop new and more advanced communication services. For example TINA-C (Telecommunications Information Networking Architecture – Consortium) is a collaborative project that defines a new telecommunications architecture based on well-defined object oriented common interfaces. In TINA, a Distributed Processing Environment (DPE) is generated upon different underlying heterogeneous network elements and provides a common 'platform' and interface for third party developers to adhere to. The objective of TINA is to provide an architecture based on distributed computing technologies that will enable telecommunications networks to support the rapid and flexible introduction of new services and the ability to manage both the services and the network infrastructure in an integrated way. In this way, TINA is often referred to as TMN + IN = TINA (Pavón et al., 1998). Services are constructed through re-usable software components developed by different vendors, while application interoperability will be provided by a distributed processing platform, possibly based on the emerging OMG CORBA technology enhancements (Banfield and Hutchinson, 1997) and (Pavón et al., 1998). Mobile software agents are not yet part of TINA-C. However, the concept of an agent operating at an intelligent middleware level is pertinent to TINA-C, and TINA-C is a likely candidate to extend itself to the paradigm of both mobile agents and intelligent agents. Standards that will emerge from projects such as TINA-C may mean that service providers will have the ability to create their own personalised services by interfacing with a common distributed processing environment (DPE).

Active networking [URL 20] is a new paradigm for the building of modern network architectures based on new capabilities of IPv6. Centred on routers and switches inside the network, nodes are given the ability to apply user-specified processes upon the packets that flow through them (Tennenhouse et al., 1997). The ideas of active networking are clearly comparable to mobile agent technology, where the packets flowing through the network can be viewed as small programs that can be executed on the router computers. One distinction that can be noted between most classifications of mobile agents (although we have read many papers that classify mobile code as mobile agents) and small packets of executable code, is the ability of a mobile agent to retain persistent state from one execution point to the next.

2.7 *Agents and economic theory for telecommunications management*

As Mullen and Wellman (1995) state, market-based price systems constitute one well-studied class of mechanisms for allocating resources among distributed decision-makers. The exploitation of economic theory, and its intrinsic ability to produce allocations with minimal communication, has probably been the most widely researched concept in application to agent-based control of telecommunications due to the fact that it is based on the large economic literature and theory available. However, in application to real world telecommunications, these ideas have not been fully exploited. This is perhaps surprising, since a telecommunications network is inherently distributed and appears to lend itself well to a economic models. The use of self-interested agents, where there is only the need for each of the individual agents to be aware of their own local state information and not aware of other agents' interests, fits the application domain of multi-party negotiation that is present in telecoms network management. The self-interested agents work in such a way that they achieve their own goals without having to explicitly consider the other agent's goals in the system. This means that the owners of the agents are able to keep market information confidential from the other authorities in the system. Therefore, the self-interested agent approach seems to be particularly appropriate for the allocation of services in a multi-service environment.

The negotiation techniques proposed by economic theory generally involve simple bidding mechanisms based on extensions to the contract net protocol (Smith, 1981a) (see section 1.5). For example, the FIPA specification defines Interaction Protocols based on the principles of market systems, such as the operational mechanics of English, Dutch or Double auctions. One of the main problems encountered with adopting an auction mechanism is how the virtual institution should be programmed to incorporate a fair bidding mechanism, i.e., keeping important information secure or isolated from competing agents (e.g., agents owned by competing service providers). Other problems are caused by deciding what the "currency" of the bundled commodities should be and what strategies the Producers and Consumer agents should use to optimise their objective(s). Some of these problems are encountered in general economic utility theory. Wellman's (1993) paper provides a description of the underlying economic theory in an agent context and a description of a system called WALRAS. Wellman's approach will be discussed briefly below.

There are a few papers that have used market oriented approaches to the management of networks (Yamaki et al., 1996; Rodriguez et al., 1997; Miller et al., 1996; Kuwabara et al., 1996). Kuwabara et al. (1996) define a case study that measures the effect of communication delay associated with disseminating the price information from sellers to buyers. A mechanism is proposed to reduce the degree of oscillation effects caused by agent communication delay. Kuwabara et al.'s work is discussed more fully in section 3.

Bigham and Hayzelden (1998) have described a mechanism that is analogous to a simple bidding mechanism to achieve the load balancing traffic through a virtual route network. Most other market-oriented resource allocation research uses metrics based solely on the variable cost for the resource of interest, e.g., the cost associated with bandwidth when the network is under high utilisation. These methods generally map into or directly relate to maximising the traffic throughput. In contrast, Bigham and Hayzelden (1998) have chosen to maximise the QoS provision of the network, by using a survivability objective function to guide resource allocation at the strategic level. Aeken and Demazeau (1996) have also pointed out the importance of survivability in a self-organising multi-agent system. They state that, "Emergent functionality only makes sense when there is an external observer attributing the functionality.... An alternative point of view is that functionality is a meaningless concept for a self-organising system. This interpretation promotes the concept of survival, or viability..."

Unlike the approach of Kuwabara et al., in Wellman's (1993) "Computational Market" there is no need to define a price function. Prices, such as the price for bandwidth along a link, are assumed to be globally known. Since prices are a compact representation they appear to be a good way of keeping the inter-agent communication requirements low. We will illustrate the approach for a simplified network flow problem. Suppose that we want to determine the flows, in cells per second say, to meet given demand requirements from source nodes to destination nodes under given capacity constraints on the links. We can imagine a buying agent dedicated to buying bandwidth for each source-destination pair and link agents selling bandwidth on each link. An agent bids for a bandwidth on links of the network. It will be bidding for bandwidth on many links. In Wellman's (1993) approach, rather than using a user-specified price function for the price of bandwidth on a link, the demand curve for each agent bids for a link's resource is estimated from the current flows in the network as follows. If we set the price of a link to zero and the link is not on a source destination pair's cheapest path, then we know that this link is not of interest, and therefore will not bid for that link. However, if when we set the price of a link to zero the link is on the source destination pair's cheapest path, then we know that this link is of interest. If we then set the price of a link to infinity so the link cannot be on a source destination pair's cheapest path, we will find the cheapest alternative path, and so an upper bound on the price that should be paid for bandwidth on this link. If the link is on the source destination's cheapest path, then it wants to use the link and buy some of its resource. The quantity bid for depends on the source destination pair's demand and link capacity constraints. Ideally, a source destination pair would like to meet its entire demand on a link if that link is on its cheapest path. In this way, the link will receive the supply and demand curves from interested source destination pairs. The link agent, or link auction, finds the clearing price (i.e., the price at which the demand for this link balances the supply), and sets the new price to be the clearing price. In Wellman's (1993) WALRAS system, this is done by a binary search using the points on the demand "curves" supplied by the source destination pair agents. After the demand has been allocated at the clearing price the demand from all agents is now zero. At this price the link is on no source destination pair's cheapest path. In this way, prices (i.e., clearing prices) are adjusted up or down, depending on whether there is excess demand or supply. As the excess demand approaches zero, the prices converge to the prices in competitive equilibrium.

Wellman argues that whilst his approach is "essentially equivalent to existing algorithms for distributed multi-commodity flow, the market framework offers an approach to extensions beyond the strict scope of the optimisation problem". It is perhaps worthwhile to repeat Wellman's observation about the current status of market oriented approaches, "speculations regarding the methodological advantages of the market-oriented framework are indeed just speculations at this point, and the relative flexibility of applications programming in this paradigm must ultimately be demonstrated empirically."

3 Software agents in communications systems

This section will review the work carried out by researchers and current projects that have used the software agent approach for communications management. The authors place the research into categories based on the type of agents that have been used. The categories include the MAS approach, Mobile Agents, the Ants based framework and Economic approach. Finally, the authors conclude with a summary and a few points about the future directions for this compelling domain. Note that this is not meant to be a complete and comprehensive survey.

3.1 Multi-agent systems approach

3.1.1 Cooperative problem solving for network management (Weihmayer and Brandau, 1990)

Weihmayer and Brandau (1990) describe the idea of coupling distributed expert systems to provide a tightly coupled distributed co-operative multi-agent customer control system. A reprint of this paper can be found in Hayzelden and Bigham (1999). Throughout the paper the authors use the metaphor of "Cook and an Architect", where each is an expert system that perceives the creation of a suitable kitchen design from a different perspective. The idea is for the Cook and the Architect to generate suitable plans for their individual requirements and then converge on a kitchen design that meets both of their requirements. Weihmayer and Brandau's paper continues with a scenario taken from network traffic control and dynamic reconfiguration.

Each of the expert systems is described as an agent. Weihmayer and Brandau state that the requirements of real world telecommunications network management appear to be better modelled as closely coupled co-operation between dissimilar agents. Some of their earlier work looked at domain level heterogeneity on the TEAM-CPS (telecommunications testbed) system, where the agents were different in their actions due to each agent being situated in a different domain. For this earlier work they concluded that in some cases agent diversity can lead to disadvantages particularly in the domain of telecommunications. "Diversity often leads to complexity increases for inter-agent communications as well as for the reasoning that agents must do about other agents" (Weihmayer and Brandau, 1990). To limit this complexity they suggest that agents should possess two sorts of common semantics, namely, knowledge level of action effects and goal intention knowledge.

3.1.2 HYBRID (Somers 1996)

HYBRID consists of a three layer agent system architecture. The HYBRID architecture adopts a hierarchy of autonomous control agents, which conduct network control based on local decision making capabilities.

HYBRID adopts three main features (Figure 4); control distribution, hierarchy and autonomous management. The aim is to provide a more scalable network management system. The hierarchy of layers is defined on the basis of statically defined regional parts of the whole network. This means that the agent architecture hierarchy is layered by local (layer 1), regional (layer 2) and national (layer 3) geographical areas. Each of the layers is therefore responsible for a particular region of the network. Within a layer of the architecture, there operates a number of "independent controllers" which are termed "authorities" (Somers, 1996). The authorities conduct the normal operations of a centralised management system, namely the FCAPS model (Fault, Configuration, Accounting, Performance and Security functional areas).

As Somers (1996) describes it, "...authorities also act like intelligent agents. That is, they speak KQML to each other. Each authority manages a set of resources (network elements). The root authority in the hierarchy is the ultimate owner of all resources, but it delegates direct management

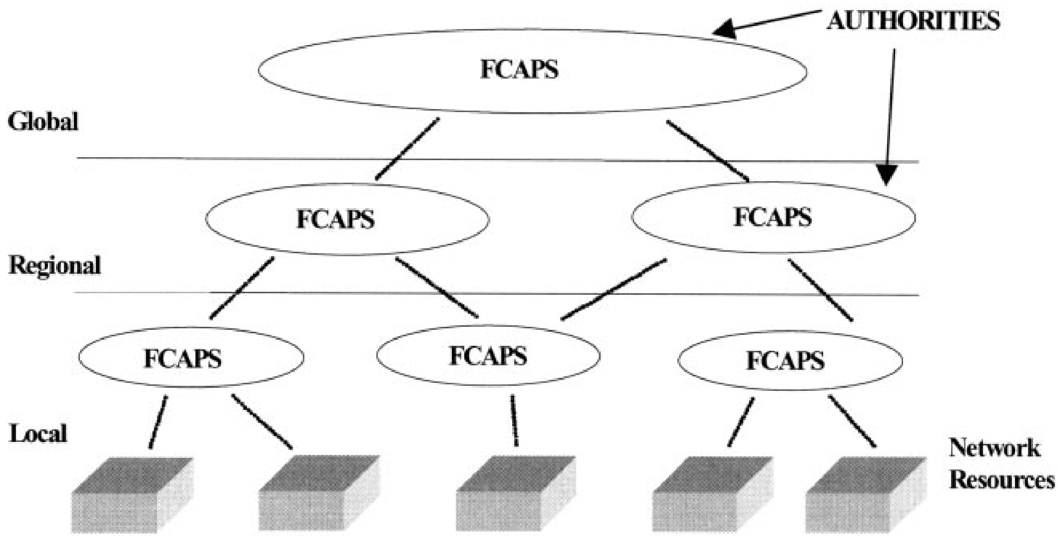

Figure 4 HYBRID system (adapted from Somers, 1996).

of some of its resources to lower levels – this recursive process is the definition of the hierarchy". HYBRID authorities are made up of multiple agents that each has a dedicated task in the network management system. These authorities run on top of an ATM configuration management system, which provide functions such as setting up connections and the ability to generate fault alarms. The authorities are co-ordinated by a single agent (Authority Controlling Agent) that conducts housekeeping tasks such as routing messages to different authorities.

Since negotiation is by virtue an iterate process, at least two message interactions has to take place before the communication can said to be a negotiation process. HYBRID uses negotiation to resolve problems that occur within the system. In HYBRID authorities can negotiate resource availability with other authorities in the network. Since there can potentially be a large number of authorities implemented on the network, negotiation for resources can be computationally expensive. To reduce the probability of this occurring, each authority has a proxy agent that updates information about other authorities, so that should re-configuration become necessary, negotiation can occur within a single authority. Of course this technique may lead to redundancy and means that control may be conducted based on out of date information.

Somers (1996) provides a description of the internal architecture that the different agents are based upon. The agents are all inherited from a basic agent architecture, which supports a slightly modified set of KQML performatives. Other agents inherit (from classes as in OO programming) other capabilities depending on the operating world environment in which they are intended to function. Functionality ranges from simple scripts that provide some reactive rules, to a "deliberative" search-based reasoning agent (Evans et al., 1999).

The authorities in HYBRID are built up from a number of agents. Similarly, Tele-MACS (see later) also adopts a layering of agents. The main difference is that Tele-MACS agents are in control of more dynamic resources, i.e. they are not so persistent within a logical grouping and do not conduct actions for a only static regions of the network. Thus, in Tele-MACS agents are in control of logical entities. If these logical entities terminate, then the agent that represents them will also terminate. This allows the agent management system to be more flexible, but possibly at the expense of complexity. Where the degree of complexity is related to the logical configuration of the network as opposed to the static physical configuration (nodes, links, etc.).

3.1.3 Management by delegation (Goldszmidt and Yemini, 1995)

Management by Delegation (M_bD) (Goldszmidt and Yemini, 1995) is an established distributed concept and technology which, is likely to incorporate the idea of Intelligent Agents. Goldszmidt and Yemini's (1998) paper describes the features of M_bD with an emphasis on the relationship to the intelligent agent paradigm.

M_bD is a decentralised network management architecture that is consistent with the incorporation of agents in the DAI sense. It has a pre-requisite that each network node in the network is of sufficient computational power (in fact today and almost certainly in the future, nodes in networks are likely to be powerful computer systems in their own right) to run distributed script processes. This enables the network management system to consist of "agents" operating at individual nodes that can receive scripts, which specify the tasks to do: hence the term Delegation. As Meyer et al. (1995) state, "At the highest level of abstraction, the decentralised M_bD paradigm and centralised SNMP (Simple Management Network Protocol) paradigm appear to be the same, as both have a Network Management System (NMS) communicating with agents via a protocol". However, there is a definite shift in paradigms, where M_bD is more concerned with the delegation of more high level goals than is obtained in the SNMP approach. The delegation of scripts as opposed to simple management messages is often more efficient in terms of bandwidth utilisation.

The idea that distinguishes M_bD from other network management paradigms is its ability to support full distribution of control to the node agents. Each node runs a program known as an Elastic Process (referred to as an "elastic agent"). The elastic agent allows new control functionality to be absorbed in the form of scripts. When there is an appropriate requirement, these newly absorbed script functions can be invoked. Other processes that require the functional services generally make this invocation. This is quite similar in concept to that of mobile agents, but it must be noted that the main idea in M_bD is to dynamically download script logic, and not to serialise mobile code and state.

3.1.4 Service management (Busuioc, 1996)

Busuioc is one of the earliest researchers to focus attention on applying agent technology to telecommunications problems. Busuioc's (1994) paper considers link bandwidth allocation and dynamic routing. In Busuioc's (1996) he pays more attention to the requirements of end users than specifically defining the tasks that network management agents need to carry out and also defines realistic scenarios of future types of networks, such as mixed fixed and mobile networks. Busuioc (1996) describes some of the requirements for the provision of an intelligent service management system that seeks to bridge the gap between the main service delivery platforms that are inevitable in the future. A taxonomy of the types of agent required to fulfil future service management systems is given. The classification includes three main types, Deterministic, Learning and Autonomous/Pro-active and Negotiating agents. The Negotiating agent type is the main concern of Busuioc's (1996) paper. Autonomous/Pro-active and Negotiating agents are defined as "...agents that break away from the traditional software model – namely command based control – and use the more natural humanistic metaphors such as prediction, decision making and negotiation, to control and manage processes. Each agent performs as a cost centre, negotiating with other agents when buying or selling resources. These agents are not mobile, but can communicate with others via messages". The service network problem is dissected into the particular domains of network, services and customers. Each domain has specific tasks to be performed. This is a similar compartmentalisation technique to assigning roles to the individual players in the system, as described by Lupu and Solman's (1997) paper. In Busuioc's paper the developed agent network architecture consists of a single flat layer. It is flat in terms of there being no hierarchy of control between the agents. One of the interesting scenarios that the agent architecture has been applied to, is that of a mixed fixed broadband and mobile type network. In this scenario, Customer Agents (CA) are the first point of call for someone requesting a service through the network. The CA attempts to configure services by co-operating with the Service Agent (SA). Similar processes propagate through the roles that the specific agents are subsuming. Trading begins (based on the Contract Net Protocol – see section 1.5) with the exchange of internal services available to other agents.

Busuioc states that further work is required on the mechanisms for allowing agents to make intelligent decisions via the information provided by other agents, that negotiation strategies require further investigation for conflict resolution and to enable the agents to be more flexible in their behaviour. A final point to mention is that Busuioc (1994) has also paid attention to the mechanism for graceful degradation in the case of agent failure. Busuioc (1994) describes a bidding mechanism that allows for self-organisation under agent failure conditions. The mechanism achieves system stability, but at the cost of reduced performance.

3.1.5 Tele-MACS (Hayzelden and Bigham, 1997, 1998b,c)

Hayzelden and Bigham (1997, 1998b,c) have applied a multi-agent control approach to the management of an ATM network. A system level management architecture which, capitalises on the architecture that the ATM reference model has defined, has been developed. The multi-layered agent control architecture (Tele-MACS) has been implemented in Java [URL 10]. The main contribution that Tele-MACS addresses is that it integrates distributed low-level reactive control agents and pro-active planning agents, which have higher levels of competence. Other factors such as dynamic bandwidth allocation have also been researched.

In the chosen application, Tele-MACS subsumes the proprietary control system. The proprietary control system functions in its intended fashion and Tele-MACS builds upon this system by providing more intelligent control. Part of the "intelligent control" comes about by the gathering of localised information from the operating network nodes. This information is processed to give a more abstract, but also more global view of the network dynamics, and thus provides the management system with an improved probability of dealing with unexpected events.

Tele-MACS consists of multiple layers of multi-agent systems, where each layer is defined to conduct the control of the network infrastructure to a certain level of competence. The adoption of a logically layered multi-agent control system appears to fit the requirements of broadband network control, namely inherent distribution, the need for intelligent control, moving intelligence from the switches to the edges, and robustness. Figure 5 depicts the interaction that occurs between the heterogeneous agents for control of avoiding network congestion and maintaining survivability of the network infrastructure. Others have discussed the importance of survivability and the process of physically dimensioning communications networks on this basis (Alevras et al., 1998). The arrows

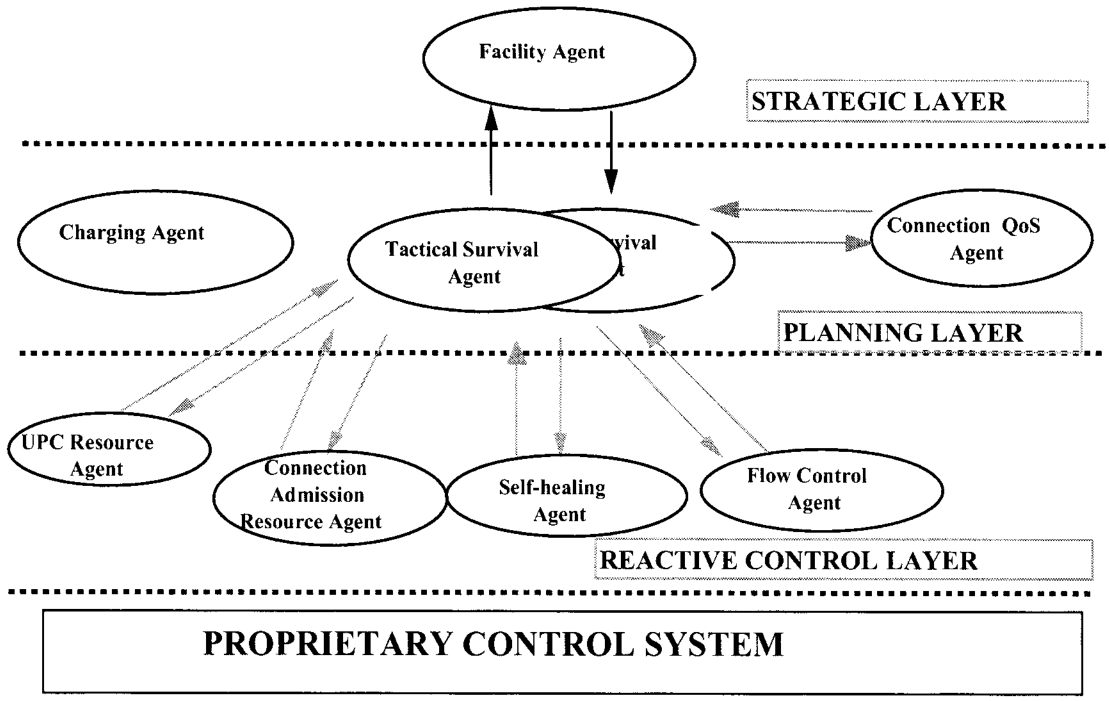

Figure 5 Tele-MACS architecture.

in Figure 5 show the co-operative relationships between the system level agents, which has been inspired by Brooks' (1986) Subsumption Architecture. Within each layer the agents co-operate on the basis of information about changes to the network state. This isolating co-operation to well-defined layers reduces the message passing overhead (constrains the broadcast of messages in situations when the network is in a failure state or stressed to limits) and sustains robustness. Further details of the operations conducted by the agents can be found in Hayzelden and Bigham (1998b). Similarly, Lewis (1995) and Evans et al. (1999) have made a case for the use of the Subsumption Architecture concepts in a telecommunications management context.

Planning algorithms to balance the "flow" through the route topology has been developed and is based on survivability of the network. The importance of survivability at the planning level is that it ensures the network is well placed with as much time as possible to react to unexpected events. Reactive agents are constrained by survivability criteria, but are guided by others. For example in normal operation they are concerned with managing utilisation. To guide the flow adjustment, different survivability functions are being experimented with including, maximising the minimum residual capacity in the available routes.

3.1.6 IMMuNe Project (Frei and Faltings, 1997)

The IMMuNE (Integrated Management for Multimedia Networks) project is focussing on the integration of the vertical concepts of short, medium, and long-term aspects of network management. The project has firstly decomposed the problem into sections based on planning for resource allocation and a lower layer for implementation of the control policy. Frei and Faltings (1997, 1998) describe some of the work that has been conducted and the algorithm for simplifying network management via hierarchical routing strategies. Known as "Blocking Island Abstractions", they describe how agents are suitable for sharing information about available resources in a multi-party scenario.

An important difference between IMMuNE and HYBRID is that higher level agents arbitrate unresolved disputes between peer agents, i.e. in HYBRID an agent reins over a static region of the network. The blocking island approach gives an abstraction and summary of the resource availability for a particular region of the network based solely on the operational state of the network (the amount of residual capacity of the links in the system) but these abstractions are highly dynamic and change from one instant to the next. As Frei and Faltings (1998) mention, this abstraction technique can help alleviate agents needing to enter "substantial negotiations" when the agents do not have all relevant and important information to make decisions. However, it must be noted that in general, continuous updates to the logical state of the network leads to large amount of control messages flowing through the network when the mechanism is implemented. Of course this matter is of little or no problem if agent communication or negotiation occurs over dedicated communications media.

3.1.7 IMPACT, ABROSE, FACTS and MARINER

Under the ACTS (Advanced Communication Technologies and Services) set of projects comes the agent based cluster, called CLIMATE. CLIMATE consists of three defined sub-clusters (IN and Mobility, Connection Management, and Information Applications). Each of these sub-clusters are decomposed into projects [URL 17]. Four of these projects are focussing particularly on the applicability of MAS technology for telecommunications control.

IMPACT is one European funded project that will implement software agent technology on the EXPERT ATM test-bed. The long-term objective of the project is to demonstrate that agent based control can act within the constraints of traditional connection admission set-up procedures and also show that agent technology can provide a more holistic and flexible control method [URL 6]. This research addresses the problem of communicating between nodes, the knowledge of resources available on the network so that the connection admission no longer has to be a solely localised procedure, but can make decisions based on a wider picture. IMPACT has utilised some concepts from the Tele-MACS architecture (Hayzelden and Bigham, 1998b), however it extends the scenarios to incorporate a multi-service environment. The performance of the IMPACT system will be evaluated against a few realistic scenarios, one based on a multi-service provider scenario. The final stage of the project will be a fully distributed demonstration. The ABROSE project's objective is to develop a MAS for Information Brokerage. It has committed to using an existing Java based agent platform. More information on the ABROSE project can be found at [URL 26]. FACTS (FIPA Agent Communication Technologies and Services) is a project considering the interaction between differently implemented agents, rather than on agent technology itself, and be structured around three application areas, namely service provision, resource reservation and electronic commerce. Its goal is to validate the work of FIPA and other standards by the construction of demonstrator systems. MARINER [URL 21] is focussing on the use of intelligent agents for load balancing control and overload protection. The agents are based on the FIPA standards and (Pitt and Prouskas, 1998) examine the performance and general issues of using FIPA messaging in real-time constrained environments. Pitt and Prouskas (1998) considers some of the implementing issues encountered when building fully distributed control systems, such as methods for synchronising distributed clocks and conflict resolution. MARINER also considers the issues surrounding market based control strategies (see section 3.4 for further discussions on market based control research).

3.1.8 EURESCOM

The purpose of EURESCOM is to support the development and provision of harmonised Europe-wide public telecommunications networks and services by furthering science and research in the field of information and communication technologies. The general framework currently consists of group projects in the four main areas of: Strategic Studies, Telecommunication Services and Applications, Networks and Systems and Enabling Technologies. Although, there are a number of interesting projects of particular interest to telecommunications integration and future developments, one project, P712 "Intelligent and mobile agents and their applicability to service and network management" is of particular interest to the agent in telecommunications community. The project is paying attention to the problems associated with highly customised multimedia and mobile services and to new concepts for effective network and service management. OMG and TINA architectures will provide the starting point of interest and be the platforms to support the projects modular approach to control and management. The main objective of this project is to assess the maturity and implications of "Intelligent and Mobile Agents" applicable to service and network management (Corley et al., 1998). More information can be found at [URL 12].

3.2 Mobile agent approach

3.2.1 Mobile agents for route selection (Appleby and Steward, 1994)

Appleby and Steward (1994) describe the application of mobile agents to congestion control in a circuit switched environment. Their work demonstrates the feasibility of using agents to autonomously manage a network. They argue that distributed network control, where fixed co-operative message passing control entities are used, leads to complex co-ordination strategies when viewed as a single system. They therefore make a case for adopting a mobile agent strategy, stating that this leads to intrinsic robustness (system and program failure), i.e., is more intuitive to write well-structured software, and finally self-regulating software requires less direct co-operation between processes. In their work, they apply a large number of homogeneous agents to control routing problems in a simulated example of the UK proposed 30 node synchronous digital hierarchy (SDH) network (in fact, they used two types of agents for different levels of control – see later). They reason that this approach provides benefits in terms of control robustness. The type of agents used are "Telescript" mobile agents. Using an "ensemble of mobile agents" of two different kinds provides control. The "ensemble" came from the authors requirements that for robust control "agents should be present in reasonably large numbers" (Appleby and Steward, 1994). In fact, the number of agents used to control the network dynamically change in population number so as to alter the dynamics of the network-state. The two different kinds of agents that they describe are load agents to provide the lowest level of control and parent agents. The load agents are responsible for distributing the communications traffic evenly through the available circuits. The parent agents' tasks are to manage the population dynamics of the load agents. The parent agents possess the ability to "launch" new mobile load agents when they detect nodes that are in high states of utilisation.

In Appleby and Stewards' experiments, the mobile load agents measure the nodal utilisation (the amount of traffic throughput in a single network node). The idea being that load agents are released from a source node and move through the network making nodal utilisation measurements. The load agents operate in a backward node seeking manner (from destination node towards the source node), generating localised knowledge of the network node utilisation states. This behaviour is achieved via the use of a distributed implementation of Dijkstra's shortest path algorithm (Dijkstra, 1959). The load agents generate the best available routes (the route with the lowest total node utilisation). The mobile load agents can obtain a "global view" of the network, if they are allowed to traverse all of the network nodes, although this is not the objective of the research. Some of the disadvantages involved in this method of node searching is cyclic routes can sometimes be generated (see Schoonderwoerd et al., 1996).

3.2.2 Cooperating mobile agents for mapping networks (Minar, Kramer and Maes, 1998)

Minar, Kramer and Maes (1998) use mobile agents to discover information about network connectivity. The agents build a map of the links and node configuration so that dynamic routing decisions can be made. Within their scenario the nodes are radio-frequency transceivers distributed through two dimensional space. Typically in this type of scenario, routing problems are solved by collecting the data of network connectivity in a centralised store. Minar, Kramer and Maes (1998) argue that, particularly in cases where traditional centralised solutions are difficult to implement (i.e., the network is itself dynamic), it may be more efficient to utilise a fully distributed mobile agent control approach.

Each agent has a three-step control cycle, which consists of learning about the edges (links) at the executing node; learning about node information from any other agents operating at that node; and learning about the choice of the next node to re-locate to. The main objective of the research is to test different "movement" strategies. The first strategy is a control mechanism where the agents choose an adjacent node at each time event. The strategies become more sophisticated. The second strategy (denoted as "Conscientious") chooses adjacent nodes that have not been visited before or have been visited least recently. The agents using the Conscientious mechanism do not base their choice on any information learned from the other agents. The final strategy, ("Super Conscientious") base the choice of node visits on information learned from other peer agents in the system.

The experiments were conducted on a 250 mode (2522 link) network and run until the agents had obtained complete information about every node in the network. The time taken to reach the "perfect knowledge" state was used as the performance measure of the mobile agent's movement strategy. The experiments were re-run with varying numbers of agents.

Results obtained from the experiments showed that agents who co-operate by passing network topology information could map out the network configuration in less time than where there is limited or no co-operation. An interesting result is that increasing the number of agents in the system does not lead to an improvement in topology mapping. The "Super Conscientious" agents, as would be intuitively expected, performed the best at low populations sizes. However, the experiments showed that at higher population sizes (100 agents in the system), the "super-conscientious" agents did not perform as well as "conscientious" agents. The reason for this behavioural change is because the "Super Conscientious" agents tend to cluster at nodes and share information, thus taking a longer amount of time to reach the "perfect information" state. This is caused by the agents sharing similar path sets and therefore following similar paths through the network, rather than dispersing themselves evenly over the network. There is a Java-based applet available at Nelson Minar's web site [URL 9]. The applet uses logged data from the actual experiments conducted. The applet shows the movement of the mobile agents within the network topology and how the link information dynamically changes as the agents carry out their co-operation with other agents in the system.

3.2.3 MAGENTA

MAGENTA (Mobile AGENT environment for distributed Applications) framework. Sahai and Morin describe MAGENTA, which is an environment for the execution of mobile agents [URL 19]. It consists of "lieus" which are static programs that are capable of creating, sending, receiving and saving agents (cf. with "Places" in IBM Aglets) [URL 15]. A lieu has a unique identifier. The agents can hop from lieu to lieu and can interact with the system and other agents when executing at a lieu. Every agent carries a permit for authorisation. The agent is only allowed to carry out system resource changes once the permit is checked and verified. MAGENTA incorporates fault tolerance mechanisms and mechanisms to support mobile computing. The authors also describe an agent based network management system, termed Astrolog. The Astrolog network management system utilises mobile agents and client/server technology for implementing the network management functionality. The authors compare the client/server and mobile agent technologies quantitatively in this context. The comparison has been done in terms of response times and bandwidth usage (Sahai and Morin, 1998).

3.2.4 MAGNA

MAGNA stands for Mobile AGeNt Architecture [URL 13]. The project is developing a mobile agent approach for enhancing and extending the developments made in the TINA-C environment. A long-term objective of the project is to develop a generic architecture that will aid the needs of future communications systems.

MAGNA have recognised the possible benefits that mobile agent technology may inject into the telecommunications industry, particularly with respect to the service scalability problem inherent in Intelligent Networks. They have, however, also realised that the value of intelligent mobile agents for the provision of telecommunication services may be questionable for some application domains in which traditional Remote Procedure Call (RPC) mechanisms might be more adequate. Therefore, they are currently developing a mobile agent architecture (MAGNA), which considers Remote Programming and RPC as potentially complementary and not mutually exclusive technologies. It is their belief that this technology will enable telecommunication services to be provided instantly and customised directly at the locations where the intelligence is needed, namely it will enable "Intelligence on Demand" (Magadanz, 1996b).

3.2.5 MIAMI

ACTS MIAMI (Mobile Intelligent Agents for Managing the Information Infrastructure) is focusing on the creation of a uniform mobile intelligent agent framework. One objective amongst others is to augment the results of selected ACTS projects by the integration of mobile intelligent agent based solutions. Although, MIAMI uses primarily OMG MASIF [URL 11], a subset of FIPA ACL validation will be considered.

3.3 Ant-based approach

3.3.1 Ant-based telecommunications control (Schoonderwoerd et al., 1996)

An introduction to mobile agent technology was provided in the first section of this paper. A slightly different approach, but certainly inspired by the mobile agent concept, has been conducted by Hewlett Packard (HP). Rather than centre on the ability to move software code from node to node, they adopt an ant based vision, where multiple simple homogeneous software code (referred to as "Ants") are allowed to randomly roam the physical network configuration (Schoonderwoerd et al., 1996).

The behaviour of ant colonies and how they co-ordinate complex activities like foraging and nest building has fascinated researchers in animal behaviour. They have proposed many models to explain these capabilities. Recently, algorithms taking inspiration from the behaviour of real ant colonies have been applied to solve many types of optimisation problems. This new approach to distributed optimisation is known as "Ant Colony Optimisation". Ant Colony Optimisation has been applied successfully to a large number of combinatorial problems, including, the travelling salesman problem, routing in telecommunications networks, and clustering and sorting problems.

The classification of ants as a very simple mobile agent system is justified because although the "Ants" in the HP work were not directly involved in carrying out control changes in the network, their presence did indirectly influence the control of the network routing strategy.

Schoonderwoerd et al. (1996) focus on using the ant analogy for the control of route path finding in networks via stigmergy. The idea is to use the "world" or the agent's operational environment as a memory store. The type of routing considered in this work is known as hop-by-hop routing. Hop-by-hop routing is where each node has a routing table, which gives information about the next nearest neighbour nodes. The routing table in its most basic form consists of a table of node identifiers each with an associated priority. To route a call, information is given to the source node, which then checks for a node that is consistent with getting to the designated destination node. Once a node is chosen, the destination information is passed onto the chosen node and the process continues until the destination node is reached. The research considers simulating ants that randomly roamed the network laying a trail of "pheromones" at each node. The amount of pheromone deposited depends upon the amount of time each of the ants are delayed in the nodes that they traverse and is directly dependent on the utilisation of the node (amount of total capacity occupied by existing calls). The higher the utilisation the longer the ants are delayed. As the ants age, the amount of pheromone that they deposit decreases. The amount of pheromone is in effect, the priority of that route. Further ants follow the routes where higher levels of pheromones have been deposited, therefore re-enforcing that path. Paths with high levels of pheromone indicate a less congested route, because congested nodes delay the ants and therefore prevent the ants from depositing a greater amount of pheromone. Schoonderwoerd et al. (1996) extends Appleby and Steward's work by paying attention to the elimination of unnecessarily long routes and also the prevention of cyclic routes. The first problem is overcome by summing the squared utilisation of the nodes from the actual node back to the source node. The later problem is overcome by making sure that the ants only update the routes in the direction of their source node and limiting the number of ants that can update the node routing table, therefore eliminating inconsistencies. A description of the ants for telecommunications control research can be found in Hayzlden and Bigham (1999, Ch. 13).

3.4 Economic approach

3.4.1 Automated auction in ATM bandwidth (Miller et al., 1996)

Miller et al. (1996) use an *Agoric* system to allocate bandwidth and quality allocations in a video conferencing situation. An agoric system is a computer based software system that encapsulates market based mechanisms such as those from economic theory. The basic set-up of the system is a centralised Auctioneer which runs a video server. The server is responsible for allocating bandwidth resources based on the bid price that the Auctioneer receives. At the client end of the system runs software that the end user interacts with. The user inputs their requirements for a video session, i.e., the QoS parameter and the price they are willing to pay for that level of service. Bids are received at the Auctioneer and the resources are allocated via a Bandwidth Deliverator. The Auctioneer and the Bandwidth Deliverator are segregated to ensure fairness is achieved, i.e., to prevent the Auctioneer having direct access to the allocation of resources.

3.4.2 MBC for distributed resource allocation (Kuwabara et al., 1996)

Kuwabara, Ishida, Nishibe and Suda (1996) use two case studies to demonstrate their use of applying a market based model to distributed resource allocation in communications network control. Kuwabara et al. (1996) describe their approach as an "equilibratory market based approach". They use a multi-agent system approach where the agents make "independent" decisions to maximise their own utilities. Despite the market based analogy we would classify the agents as co-operative rather than self-interested, as it might superficially appear. This is discussed after a brief description of the work.

Case study 1 – investigates a "sensitivity factor", which determines how changes are to be made in the resource requests in response to price changes.

Case study 2 – investigates at the effect of communications delay associated with disseminating the price information from the seller agents to the buying agents.

In the first case study they formulate a bandwidth management problem as a resource allocation problem. They define a metric for bandwidth utilisation and a global objective function that is a measure of the variability of the utilisation of the resources. Minimising the objective function spreads the resources over the network. In their solution to the problem they ignore the capacity constraints, using the fact that the objective function spreads the load across the resources. The route application cited in the paper can be interpreted in different meaningful ways depending on whether a "physical connection" is taken to be a path from a source to a destination or a link between nodes. Different approaches were considered in the paper ranging from making small increments to the flow in paths to making the maximum possible increment. The simulations showed, not unsurprisingly, that larger steps increase the rate of convergence, but can cause oscillations in the solution variables.

Are Kuwabara's agents co-operative or self-interested? The price function chosen by Kuwabara et al. was the utilisation metric, which is intuitively reasonable and indeed gives convergence to the global optimal resource allocation in the simulated market. It can be seen here that, the price function reflects the notion of common good; the price goes up with utilisation, hence penalising over utilisation.

A general question arises concerning how the price function should be chosen in a particular application. If the distributed resource allocation problem is formulated using a modification of the more conventional approach of optimising using the derivative of the global criterion, to compute the direction of steepest ascent, then we can see what price functions would correspond to the gradient function. If this is done for the objective function selected by Kuwabara et al. then the derivative of the global optimisation criterion they use is a monotonic function of their price function. The same is true of the price function chosen in a similar kind of formulation by Gibney and Jennings (see below). So the agents described by Kuwabara et al. (1996) and by Gibney and Jennings (1998) are arguably co-operative agents and so the issues for and against a distributed approach previously described, apply. However, if we are dealing with different service providers, the determiners of the bandwidth price for exchanges between service providers (at least for the duration of the service provider's contract with the network provider) have little relationship to overall network utilisation and so the co-operative model is almost certainly not applicable.

3.4.3 Market-based management of telecommunications networks (Gibney and Jennings, 1997, 1998)

Gibney and Jennings (1997) adopt a market-based control approach for the allocation of connections in an ATM-type network. They propose to use intelligent agents for the allocation of resources in a broadband environment and highlight some of the main benefits that general MAS can bring to distributed and dynamic domains. They state that since ATM networks are of an intrinsic high speed nature it can often be the case that management data can become out of date extremely rapidly. To overcome this fundamental problem encountered in high speed networks they suggest the use of economic theory to pro-actively couple network dynamics with localised control activity. They develop these ideas further in Gibney and Jennings (1998) and provide a detailed description of how market based control can lead to situations where, under high network utilisation, the market based strategy achieves better performance than experiments utilising basic fixed routing methods. Further extensions to Gibney and Jennings' work incorporates multi-service provider scenarios.

4 Conclusions

This paper has provided an introduction to the research literature on software agents for telecommunications systems. We have briefly described the differing classifications of software agents, including mobile agents and stationary intelligent co-operative agents. The first part of this paper described our perception of agent technology and some of the benefits of agent-based approaches have over existing technologies. Later sections considered some of the problems that are found in the communications industry.

An important point to note is that the research literature does not address the issues of agent granularity and its intimate relationship with the chosen architecture, both for the MAS as a whole and for the internal agent architectures. We have given some insight into this dependency. It is quite evident that agent internal architectures that pay attention to being reactive are far more suited for making real-time and local control decisions. This is because their internal architecture supports the requirement that they should be able to generate control actions within the available time interval. Similarly, adopting a deliberative internal agent architecture is logically more useful where pro-active and slow time scale planning is of importance. One way of overcoming the requirements for having a reactive element of behaviour with some pro-active planning behaviour is to adopt a MAS which is composed of both types of agents. This is why the authors believe in an approach which balances planning (where message propagation delay can be managed) and almost immediate reaction under policies generated by the planning components (see section 3.1.5). However, this means that unexpected occurrences must be handled using only local information until the strategic levels can asses the new situation. Note, however, that reactive behaviour is not fixed. It can and indeed should be made adaptive to previous events. In this way the reactive model can build up 'case histories' such as used in case-based reasoning, or constraint neural network classifiers, and learn appropriate responses to input triggers.

In the literature there still tends to be disagreement between the mobile agent work and the intelligent agent approach. It is often difficult to see where mobile agents have a clear application domain that cannot be achieved by the message-passing concept. We have detailed examples in the telecommunications control domain, where we believe that mobile agents will have an advantage over the message passing approach, namely the application of mobile agents for the mobile user disconnection problem. The second situation is where propagation delay becomes a significant factor for message passing when network nodes require excessive interaction or because they are geographically far apart. There also seems to be the debate over the trade-off in the amount of physical communication overhead the differing approaches generate in a particular scenario (Kotz et al., 1997). Although messages are generally just a few bytes, it may be that stationary agents, frequently passing messages to update the knowledge of the network when under stressed operational states, may create more traffic congestion than moving a single software process to the control point and beginning control execution there.

We therefore conclude that the adoption of the multi-agent approach for telecommunications management and control is largely dependent upon the ability to design and develop a "society of agents" which can be flexible enough to meet some of the future network requirements that we have described. Careful consideration is therefore required when de-coupling the management problem, something which agent researchers have not necessarily addressed to date.

The next stage for the deployment of agent technology into telecommunication systems will require further research and is still dependent upon enabling technologies such as Java, CORBA, AI research, and probably most importantly some further guidelines [URL 22] or suitable toolkits [URL 7] on how to construct agent-based systems.

Other points that we have tried to highlight in this paper are the major security issues involved in placing agents into communications systems, particularly with respect to the Mobile Agent paradigm. Some of the fears surrounding the introduction of agent technology into telecommunications systems may be ameliorated by fielded trails, some of which are now in progress. These trails, particularly the ACTS projects, will help the agent concept gain acceptance.

Acknowledgements

The authors thank BT Labs and EPSRC for partly funding this work. The authors would like to thank the people who provided comments on early versions of this paper, particularly those in our own department, including Shaw Green, Nick Jennings, Zhiyuan Luo, and especially Michael Wooldridge. Thanks to Simon Parsons (Queen Mary and Westfield College, UK) and the Network Management Research group at BT Labs (particularly Michael Cox and Rob Davison) for their encouragement to write this paper. Thanks Nelson Minar (MIT, USA) for comments on the mobile agent section. Specific thanks to Divine Ndumu (BT Labs, UK) for providing some additional information and interesting research references. Finally, the authors thank the anonymous reviewers who provided some useful comments on the original version of this paper.

References

Adams, JL, 1995, "Asynchronous transfer mode – an overview" *BT Technology Journal* **13**(3) 9–14.

Albayrak, S and Garijo, FJ, 1998, "Intelligent agents for telecommunications applications" *Lecture Notes in Artificial Intelligence 1437. Proceedings of the Second International Workshop, IATA'98* Paris, France. Springer-Verlag

Albayrak, S, 1998, "Intelligent Agents for Telecommunications Applications" *Frontiers in Artificial Intelligence and Applications 36* IOS Press.

Alevras, D, Grötschel, M, Jonas, P, Paul, U and Wessäly, R, 1998, "Survivable mobile phone network architecture: Models and solution method" *IEEE Communications Magazine* **36**(3) 88–93.

Appleby, S and Steward, S, 1994, "Mobile software agents for control in telecommunications networks" *BT Technology Journal* **12**(2) 104–113.

Banfield, M and Hutchison, D, 1997, "Providing QoS Based Routing Under TINA" *Proceedings of the 4th Communication Networks Symposium* Manchester.

Barbuceanu, M and Fox, M, 1995, "COOL: A language for describing co-ordination in multi-agent systems" *First International Conference on Multi-Agent Systems (ICMAS'95)* San Francisco, CA, pp 17–24.

Bigham, J and Hayzelden, A, 1998), "Subsumption and survivability in bidding for bandwidth" *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS'98)* Paris, France. IEEE Press.

Black, U, 1997, *Emerging Communications Technologies* Prentice Hall.

Bond, A and Gasser, L (eds), 1988, *Readings in Distributed Artificial Intelligence* Morgan Kaufman.

Bradshaw, JM (ed), 1997, *Software Agents* MIT Press.

Brooks, R, 1986, "A robust layered control system for a mobile robot" *IEEE Journal of Robotics and Automation* **2**(1) 14–23.

Busuioc, M AND Griffiths, D, 1993, "Co-operating intelligent agents for service management in communications networks" *Proceedings of CKBS-SIG* Keele, UK.

Busuioc, M, 1994, "Co-operating intelligent agents for service management in communications networks" *IEE Symposium* Cambridge, UK.

Busuioc, M, 1996, "Distributed intelligent agents — a solution for the management of complex services" *IATA'96 Workshop* held at *ECAI 96* Budapest, Hungary.

Chaib-draa, B, Moulin, B, Mandiau, R and Millot, P, 1992, "Trends in distributed artificial intelligence" *Artificial Intelligence Review* **6** 35–66.

Chan, MC, Pacifici, G and Stadler, R, 1997, "Managing multimedia network services" *Journal of Network and Systems Management* **5**(3).

Clearwater, SH, 1996, *Market-Based Control: A Paradigm for Distributed Resource Allocation* World Scientific.

Cohen, PR and Levesque, HJ, 1997, "Communicative actions for artificial agents" In: JM Bradshaw (ed) *Software Agents* MIT Press, pp 410–436.

Corley, S, Magro, D, Malabocchia, F and Meinköhn, J, 1998, "The application of intelligent and mobile agents to the management of software problems in telecommunications" *Intelligent Agents for Telecommunications Applications (IATA'98): LNAI 1437* Springer-Verlag, pp 118–129.

Cuthbert, LG and Sapanel, JC, 1993, *ATM The Broadband Telecommunications Solution* IEE Telecommunications Series 29.

Davison, R and Azmoodeh, M, 1997, "Performance management of public ATM networks – a scaleable and flexible approach" *Proceedings 5th IFIP/IEEE International Symposium on Integrated Network Management*.

Davison, R, Hardwicke, J and Cox, M, 1998, "Applying the agent paradigm to network management" *BT Technology Journal* **15**(3) (to appear).

Decina, M and Trecordi, V, 1997, "Convergence of telecommunications and computing to networking models for integrated services and applications" *Proc. IEEE* **85**(12).

Dijkstra, EW, 1959, "A note on two problems in connection with graphs" *Numerical Mathematics* **1** 269–271.

Evans, R, Somers, F, Kerr, D and O'Sullivan, D, 1999, "A multi-agent system architecture for scalable management of high performance networks: Applying arms length autonomy" In: A Hayzelden and J Bigham (eds) *Software Agents for Future Communications Systems* Springer-Verlag.

Frei, C and Faltings, B, 1997, "Intelligent agents for network management" *AI for Network Management Systems 1997* IEE Digest No. 97/094.

Frei, C and Faltings, B, 1998, "A dynamic hierarchy of intelligent agents for network management" *IATA'98* Paris. Springer-Verlag.

Fuggetta, A, Picco, GP and Vigna, G, 1998, "Understanding code mobility" *IEEE Trans. Software Engineering* **24**(5) 342–362.

Gaïti, D and Boukhatem, N, 1996, "Cooperative congestion control schemes in ATM networks" *IEEE Communications Magazine* **34**(11) 102–110.

Gaïti, D, 1996, "A proposal for integrating intelligent management in the intelligent network conceptual model" *Computer Networks & ISDN Systems* **28**(4).

Gibney, MA and Jennings NR, 1997, "Market based multi-agent systems for ATM network management" *Proc. 4th Communications Networks Symposium* Manchester, UK. (Also available at `http://www.elec.qmw.ac.uk/dai/projects/agentCAC/`)

Gibney, MA and Jennings, NR, 1998, "Dynamic resource allocation by market-based routing in telecommunications networks" In: S Albayrak and FJ Garijo (eds) *Intelligent Agents for Telecommunications Applications (IATA'98): LNAI 1437*. Springer Verlag, pp 102–117.

Glitho, R (ed), 1998, "Management of heterogeneous networks" *IEEE Communications Magazine* **36**(3).

Goldszmidt, G and Yemini, Y, 1995, "Distributed management by delegation" *15th International Conference on Distributed Computing Systems* IEEE Press.

Goldszmidt, G and Yemini, Y, 1998, "Delegated agents for network management" *IEEE Communications Magazine* **36**(3).

Green, S, Hurst, L, Nangle, B, Cunningham, P, Somers, F and Evans, R, 1997, "Software agents: A review" Technical Report TCD-CS-1997–06, Trinity College, University of Dublin. (Available at `http://www.cs.tcd.ie/research_groups/aig/iag/`)

Greenwood, DPA, 1998, "Intelligent mobile agents: An emerging technology for next generation telecommunications" *15th UK Teletraffic Symposium on Performance Engineering in Information Systems* Durham, UK.

Griffeth, N and Velthuijsen, H, 1997, "A negotiation protocol for telecommunications service provisioning environment" Proposal for FIPA'97. (Based on "The negotiating agents approach to run-time feature interaction resolution" In Bouma and Velthuijsen (eds) *Feature Interactions in Telecommunications Systems* IOS Press, 1994.) (Available at `http://fipa.comtec.co.jp/proxy/torino/cfp1/propos97_013.htm`)

Halsall, F, 1996, *Data Communications, Computer Networks and Open Systems* Addison-Wesley.

Hardwicke, J and Davison, R, 1998, "Software agents for ATM performance management" *NOMS'98*.

Harrison, C, Chess, D and Kershenbaum, A, 1995, "Mobile agents: Are they a good idea?" Available at `http://www.research.ibm.com/massive/mobag.ps`

Hayzelden, A and Bigham, J, 1997, "Management of ATM resource configuration, automated by distributed agent control" *Proc. IEE 4th Network Communications Symposium* Manchester, UK. (Available at `http://www.agentcom.org/agentcom/`)

Hayzelden, A and Bigham, J, 1998a, "Intelligent route management in ATM networks using distributed software agents" *15th UK Teletraffic Symposium on Performance Engineering in Information Systems* Durham, UK. (Available at `http://www.agentcom.org/agentcom/`)

Hayzelden, A and Bigham, J, 1998b, "Heterogeneous multi-agent architecture for ATM Virtual Path Network resource configuration" In: S Albayrak and FJ Garijo (eds) *Intelligent Agents for Telecommunications Applications (IATA'98): LNAI 1437*. Springer-Verlag, pp 45–59.

Hayzelden, A, 1998c, "Telecommunications multi-agent control system (Tele-MACS)" *ECAI'98* Brighton, UK. (Available at `http://www.agentcom.org/agentcom/`)

Hayzelden, A and Bigham, J (eds), 1999, *Software Agents for Future Communications Systems* Springer-Verlag.

Horikawa, K, Aida, M and Sugawara, T, 1996, "Traffic control scheme under the communication delay of high-speed networks" *ICMAS-96* pp 111–117.

Howlett, D, 1997, "Intelligent agents" *UNIX and NT News* 104, pp 18–20.

Huhns, MN and Singh, MP (eds), 1998 *Readings in Agents* Morgan Kaufman.

IATA, 1996, "Intelligent agent for telecommunications applications" *Workshop Proceedings, ECAI'96* Budapest, Hungary.

ICM D16 , 1994, "Updated TMN Architecture and Functions" ICM Deliverable 16.

Jennings, NR, 1996, "Coordination techniques for distributed artificial intelligence" In: GMP O'Hare and N Jennings (eds) *Foundations of Distributed Artificial Intelligence* Wiley.

Jennings, NR and Wooldridge, MJ (eds), 1998, *Agent Technology: Foundations, Applications and Markets* Springer-Verlag.

Kotz, D, Gray, R, Nog, S, Rus, D, Chawla, S and Cybenko, G, 1997, "AGENT TCL: Targeting the needs of mobile computers" *IEEE Internet Computing* July–August, pp 58–67.

Kuwabara, K, Ishida, T, Nishibe, Y and Suda, T, 1996, "An equilibratory market-based approach for distributed resource allocation and its applications to communication network control'" In: SH Clearwater (ed) *Market-Based Control: A Paradigm for Distributed Resource Allocation* World Scientific.

Labrou, Y and Finin, T, 1998, "Semantics and conversions for an agent communication language" In: M Huhns and M Singh (eds) *Readings in Agents* Morgan Kaufmann, pp 235–242.

Lewis, L, 1995, "AI and intelligent networks in the 1990s and into the 21st Century" In: J Liebowitz and DS Prerau (eds) *Worldwide Intelligent Systems: AI Approaches to Telecommunications and Networks* IOS Press, pp 109–125.

Liebowitz, J and Prerau, DS, 1995, *Worldwide Intelligent Systems: AI Approaches to Telecommunications and Network Management* IOS Press.

Lupu, EC and Solman, M, 1997, "Towards a role-based framework for distributed systems management" *Journal of Network and Systems Management* **5**(1) 5–30.

Magedanz, T, Rothernel, K and Krause, S, 1996, "Intelligent agents: An emerging technology for next generation telecommunications?" *INFOCOM '96* San Francisco, CA.

Magedanz, T, 1996, "Mobile agent-based service provision in intelligent networks" *IATA'96* at *ECAI'96* Budapest, Hungary.

Meyer, K, Erlinger, M, Betser, J, Sunshine, C, Goldszmidt, G and Yemini, Y, 1995, "Decentralising control and intelligence in network management" *Integrated Network Management IV* Chapman & Hall, pp 4–15.

Miller, MS, Krieger, D, Hardy, N, Hibbert, C and Tribble, ED, 1996, "An automated auction in ATM network bandwidth" In: SH Clearwater (ed) *Market-Based Control: A Paradigm for Distributed Resource Allocation* World Scientific.

Minar, N, Kramer, KH and Maes, P, 1998, "Cooperating mobile agents for mapping networks" *Proc. 1st Hungarian National Conference on Agent Based Computing*. (`http://www.media.mit.edu/~nelson/research/routes-coopagents/`)

Morreale, P,1998, "Agents on the move" *IEEE Spectrum* **35**(4) 34–41.

Mullen, T and Wellman, MP, 1995, "A simple computational market for network information services" *Proc. 1st International Conference on Multi-agent Systems (ICMAS-95)* San Francisco, CA.

Müller, JP, 1996, *The Design of Intelligent Agents: A layered approach* Springer-Verlag.

Müller, JP, 1998, "Architectures and applications of intelligent agents: A survey" *The Knowledge Engineering Review* **13**(4) 353–380.

Nwana, HS, 1996, "Software agents: an overview" *The Knowledge Engineering Review* **11**(3) 205–244.

Nwana, HS, Lee, L and Jennings, NR, 1996, "Co-ordination in software agent systems" *BT Technology Journal* **14**(4).

Nwana, HS, Ndumu, DT and Azarmi, N, 1997, "Agents of change in future communication systems" *UNICOM Seminar series 'Agents in Telecommunications'* London. (`http://www.unicom.co.uk`)

Nwana, HS and Azarmi, N (eds), 1997, *Agents and Soft Computing: Towards Enhancing Machine Intelligence* Springer-Verlag.

Nahrstedt, K and Smith, J, 1996, "Design, implementation, and experiences of the OMEGA end-point architecture" *IEEE Journal on Selected Areas in Communications* **14**(7).

O'Hare, GMP and Jennings, N (eds), 1996, *Foundations of Distributed Artificial Intelligence* Wiley

Pavón, J, Tomás, J, Bardout, Y and Hauw, LH, 1998, "CORBA for network and service management in the TINA framework" *IEEE Communications Magazine* **36**(3) pp 72–79.

Pitt, JV, Mamdani, EH, Hadinham, RG and Tunnicliffe, AJ, 1997, "Agent-oriented middleware for telecommunications network & service management" *AI for Network Management Systems* IEE Digest No. 97/094

Pitt, J and Prouskas, K, 1998, "Initial Specification of Multi-Agent System for Realisation of Load Control and Overload Protection Strategy" Mariner deliverable 3. (Available at `http://www.teltec.dcu.ie/mariner/`)

Pitts, JM and Schormans, JA, 1996, *Introduction to ATM Design and Performance* Wiley.

Preist, C and Pearson, S, 1997, "An Adaptive Choice of Messaging Protocol in Multi Agent Systems" *Foundations of Multi-Agent Systems (FoMAS'97)* Warwick, UK.

Quendt, B, 1997, "An agent-based resource-control in a signaling system for the open telecommunication market" *Proceedings of PAAM '97* London, UK.

Rao, A and Georgeff, MP, 1995, "BDI agents: From theory to practice" *1st International Conference on Multi-agent Systems (ICMAS-95)* San Francisco, CA, pp 312–319.

Rizzo, M and Utting, IA, 1995, "An agent-based model for the provision of advanced telecommunications services" *Proceedings of the 5th Telecommunications Information Networking Architecture (TINA) Workshop* Melbourne, Australia, pp 205–218.

Rodriguez, J, Noriega, P, Sierra, C and Padget, J, 1997, "FM96.5 A Java-based electronic auction house" *Proceedings of PAAM 97* London, UK, pp 207–224.

Russell, S and Norvig, P, 1995, *Artificial Intelligence: A Modern Approach* Prentice Hall.

Sahai, A and Morin, C, 1998, "Mobile agents for enabling mobile user aware applications" *Proceedings 2nd International Conference on Autonomous Agents (Autonomous Agents '98)* Minneapolis/St. Paul.

Sandholm, T and Lesser, V, 1995a, "On automated contracting in multi-enterprise manufacturing" *Proceedings of the 'Improving Manufacturing Performance in a Distributed Enterprise: Advanced Systems and Tools' Conference"* Edinburgh, Scotland, pp 33–42.

Sandholm, T and Lesser, V, 1995b, "Issues in automated negotiation and electronic commerce: Extending the contract net framework" *1st International Conference on Multi-agent Systems (ICMAS-95)* San Francisco, CA, pp 328–335.

Sandholm, T and Lesser, V, 1996, "Advantages of a leveled commitment contracting protocol" *13th National Conference on Artificial Intelligence (AAAI-96)* Portland, OR, pp 126–133.

Schmid, RM, Beller, R, Giordano, S and Flinck, H, 1997, "IP and ATM – a position paper" EXPERT deliverable, AC094, WP 2.1.

Schoonderwoerd, R, Holland, O and Bruten, J, 1997, "Ant-like agents for load balancing in telecommunications networks" *Proceedings 1st International Conference on Autonomous Agents.* `http://www-uk.hpl.hp.cpm/people/ruud/`

Searle, J, 1969, *Speech Acts – An essay in the Philosophy of Language* Cambridge University Press.

Sincoskie, D, 1997, "Broadband ISDN is happening – except it's spelled IP" *IEEE Spectrum* pp 32–33.

Skarmeas, N and Clark, KL, 1997, "Intelligent Routing based on Active Patterns as the basis for the integration of distributed information systems" `http://www-lp.doc.ic.ac.uk/~klc/`

Smith, R, 1996, "Software agent technology" *Proceedings of PAAM '96* London, UK.

Smith, RG, 1981a, "The Contract Net protocol: High-level communication and control in a distributed problem solver" In: A Bond and L Gasser (eds) *Readings in Distributed Artificial Intelligence* Morgan Kaufman.

Smith, RG and Davis, R, 1981b, "Frameworks for cooperation in distributed problem solving" *IEEE Transactions on Systems, Man and Cybernetics* **11**(1) 61–70. (Also in A Bond and L Gasser (eds) *Readings in Distributed Artificial Intelligence* Morgan Kaufman.)

Somers, F, 1996, "HYBRID: Intelligent Agents for Distributed ATM Network Management" *IATA Workshop at ECAI'96* Budapest, Hungary.

Stallings, W, 1998, "SNMP and SNMPv2: The infrastructure for network management" *IEEE Communications Magazine* 36(3) pp 37–43.

Stevenson, D, 1995, "Network Management: What it is and what it isn't" White Paper (1995). (Available at: `http://netman.cit.buffalo.edu/Doc/Dstevenson`)

SUN, "1997, "Java Computing in Telecommunications: White Paper" Available from `http://www.sun.co.uk/publication/whitep/javawptelecom3.html`

Tennenhouse, D, Smith, J, Sincoskie, WD and Wetherall, D, 1997, "A survey of active network research" *IEEE Communications Magazine*.

Venners, B, 1997a, "Under the Hood: The architecture of aglets". `http://www.javaworld.com/javaworld/jw-04-1997/jw-04-hood.html`

Venners, B, 1997b, "Solve real problems with aglets, a type of mobile agent". `http://www.javaworld.com/javaworld/jw-05-1997/jw-05-hood.html`

Wang, Z and Crowcroft, J, 1996, "Quality-of-Service routing for supporting multimedia applications" *IEEE Journal on Selected Areas in Communications* **14**(7).

Weihmayer R and Brandau R, 1990, "Cooperative distributed problem solving for communication network management" *Computer Communications* **13**(9) 547–556.

Weihmayer, R and Velthuijsen, H, 1995, "Distributed AI and co-operative systems for telecommunications" In: J Liebowitz and DS Prerau (eds) *Worldwide Intelligent Systems: AI Approaches to Telecommunications and Network Management* IOS Press, pp 227–260.

Wellman, M, 1993, "A market-oriented programming environment and its application to distributed multicommodity flow problems" *Journal of Artificial Intelligence Research* **1** 1–23.

White, JE, 1997, "Mobile agents" In: JM Bradshaw (ed) *Software Agents* MIT Press, pp 437–472.

Wooldridge, M and Jennings, NR, 1995, "Intelligent agents: Theory and practice" *The Knowledge Engineering Review* **10**(2) 115–152.

Wooldridge, M and Jennings, NR, 1998, "Pitfalls of agent-oriented development" *Proceedings 2nd International Conference on Autonomous Agents (Autonomous Agents '98)* Minneapolis/St. Paul.

Wooldridge, M, 1998, "Verifiable semantics for agent communication languages" *Proceedings 3rd International Conference on Multi-Agent Systems (ICMAS-98)* Paris, France.

Yamaki, H, Wellman, M and Ishida, T, 1996, "A market-based approach to allocating QoS for multimedia applications" *Proceedings 2nd International Conference on Multi-agent Systems (ICMAS-96)* Kyoto, Japan.

Yamaki, H, Yamauchi, Y and Ishida, T, 1998, "Implementation issues on market-based QoS control" *Proceedings 3rd International Conference on Multi-Agent Systems (ICMAS-98)* Paris, pp 357–364.

Universal Resource Locators

[URL 1] TINA-C homepage: http://www.tinac.com
[URL 2] Perkins, CE, "Mobile Networking Through Mobile IP". http://www.computer.org/internet/v2n1/perkins.htm
[URL 3] Grasshopper platform: http://www.ikv.de
[URL 4] FIPA Specifications: http://drogo.cselt.it/fipa/ or http://www.fipa.com
[URL 5] KQML Overview: http://www.cs.umbc.edu/kqml/kqml97.pdf
[URL 6] ACTS IMPACT Project: http://www.acts-impact.org/impact/
[URL 7] AgentBuilder Framework: http://www.agentbuilder.com
[URL 8] IBM Aglet Framework: http://www.trl.ibm.co.jp/aglets
[URL 9] Nelson Minar's Web page: http://www.media.mit.edu/~nelson/research/routes-coopagents/
[URL 10] Tele-MACS Project: Web page: http://www.agentcom.org/agentcom/
[URL 11] OMG Standards: http://www.omg.org/
[URL 12] Eurescom P712 Project: 'Intelligent agents and their application to service and network management. http://www.eurescom.de/public/Projects/
[URL 13] MAGNA Project. http://www.fokus.gmd.de/oks/research/magna_e.html
[URL 14] General Magic Corp. http://www.generalmagic.com/
[URL 15] Programming Mobile Agents in Java – With the Java Aglet API. DB Lange and M Oshima, IBM Research. http://www.trl.ibm.co.jp/aglets/aglet-book/index.html
[URL 16] ABROSE Project; http://b5www.berkom.de/ABROSE/
[URL 17] CLIMATE ACTS Cluster. http://www.fokus.gmd.de/research/cc/ima/acts-d5-ac/
[URL 18] http://fipa.comtec.co.jp/proxy/torino/cfp1/propos97_013.htm
[URL 19] Akhil Sahai's homepage: http://www.irisa.fr/prive/sahai/
[URL 20] Telemedia, Networks and Systems Group: http://www.tns.lcs.mit.edu/
[URL 21] MARINER Project http://www.teltec.dcu.ie/mariner/
[URL 22] AgentLink: http://www.agentlink.org/