Cooperative Agents that Adapt for Seamless Messaging in Heterogeneous Communication Networks

Suhayya Abu-Hakima **Ramiro Liscano** **Roger Impey**
Seamless Personal Information Networking Group
Institute for Information Technology
National Research Council of Canada
Ottawa, Canada K1A 0R6
suhayya@ai.iit.nrc.ca
tel: 613-991-1231 fax: 613-952-7151

Abstract

This paper describes the difficult problem of seamless messaging in heterogeneous communication environments and the design of cooperative agents to address the problem. The seamless messaging agents include coarse-grained agents such as user agents, message transfer agents, server agents (for resources such as speech recognition and generation), application agents (interfacing to mail applications such as Microsoft and Unix mail) and device agents (for wireless phones, laptops, etc.). It also includes fine-grained agents, namely, user surrogate agents. Seamless messaging requires real-time performance which necessitates a tradeoff between agent communication, adaptability and intelligence to achieve a variety of tasks. In addition, seamless messaging requires user interface mediation between users and device agents to allow, for example, messages sent by users from a powerful graphic workstation to be received transparently by a cellular phone.

1. Introduction

Today's intensive communication environments require users to interact across heterogeneous networks and applications. Users send and receive voice mail, electronic mail (e-mail) and fax over a variety of wired and wireless networks. The paper addresses *Seamless Messaging (SM)* in heterogeneous environments and the design of cooperative agents to address the problem. Seamless messaging is defined here as the ability of the user to send and receive messages in a manner transparent to the modality (be it voice mail, email or fax mail), the networks (be it wired or wireless voice or data) and the devices (be it a cellular phone, a pager, a desktop computer, etc.). To achieve such transparency networks have to be used invisibly, users have to be located intelligently (through electronic calendars or active devices) and messages have to be tailored for delivery to diverse devices on demand (e.g. messages have to be intelligently converted and filtered from long multimedia desktop messages to short voice-only ones). Six types of SM agents are modelled. They include coarse-grained and fine-grained agents. The coarse-grained agents are more capable specialists than fine-grained agents and include user agents, message transfer agents, server agents, application agents and device agents. The SM fine-grained agents are user agent surrogates that are messengers from the user agent to the other coarse-grained agents.

The remainder of the paper includes: section 2.0 on relevant work, section 3.0 on seamless messaging agents, section 4.0 on the tradeoff between communication and agent capability, section 5.0 on user interface mediation for SM devices, and section 6.0 on future work and conclusions.

2. Related Work

Networks can be seen as a natural domain for the application of distributed artificial intelligence, and more particularly, agent-based computing technology [Reinhardt 94]. In particular, Weihmayer and Velthuijsen suggest a number of reasons for this, including their inherent distribution (e.g. along spatial, functional, and temporal lines), the proliferation of heterogeneous devices and services associated with them (this is particularly true of multi-vendor mixed computing-communications networks), the growing need for privacy, the sustained demands for high performance, and the increasing desire for "intelligence" in the network [Weihmayer and Velthuijsen 94].

Modelling messaging in organization services as a collection of coordinated agents results in a number of benefits. For example, a degree of virtual homogeneity is brought to otherwise heterogeneous networks of computer-telephony messaging services and devices (such as voice mail, e-mail or fax mail); relatedly, a more open network architecture facilitating more rapid and effective deployment of "plug and play" messaging services is made possible. All the same, the agent metaphor does not, in and of itself, directly resolve any of the technical issues related to system interoperability such as sharing remote resources, guaranteeing a particular quality of service or resolving the network feature interaction problem. Rather, as Laufmann points out in [Laufmann 94] "the metaphor provides a model of coordination that addresses real-world issues of the computing and communications marketplaces, and in so doing leverages the deployment of new technical solutions as they become available". In this respect, the *Seamless Messaging User, Server and Application agents* are analogous in scope and purpose to Laufmann's coarse-grained agents or CGAs [Laufmann 94].

These *Seamless Messaging (SM)* agents, like those supported by the Carnot project's Distributed Communicating Agents (DCA) tool [Huhns et al. 93], are essentially high-performance problem solvers which can be located anywhere within and among networked enterprise resources. These agents are intended to communicate and cooperate with each other, and with human agents. Through the use of models of other agents and resources within the enterprise, SM agents, like DCA agents, will be able to cooperate to provide integrated and coherent management of information in heterogeneous computing-communications environments.

The convergence of networks and the need for personal digital assistants with embedded agents that interface to all the information accessible through a web of networks (Internet, World Wide Web, etc.) has also been cited as a key reason to develop cooperative multi-agent systems [Rosenschein and Zlotkin 94]. SM agents are being developed to enable users to share valuable messaging services such as voice and e-mail as if in a single seamless network across a variety of devices that include cellular phones, PDAs, wired telephones and wireless laptop computers.

The Multi-Agent Network Architecture (MANA) described in [Weiss et al. 95; Abu-Hakima et al. 95] is designed to provide services across communication networks. In some ways, its objectives are similar to those of the concepts in Seamless Messaging. However, its agents are coarse-grained agents which are compiled at run-time and hence lack the flexibility to adapt as would be necessary in a seamless messaging environment. SM agents include surrogate user agents which are fine-grained agents that are light weight specifically designed for reactive situated behaviour.

3. Modelling Seamless Messaging Agents

What is SM?
Seamless Messaging provides users with the capability to work freely in distributed personal workspaces. It allows the creation, encoding, filtering and delivery of messages across heterogeneous networks. Thus, users can seamlessly deliver voice or electronic mail to wireless or wired mail environments. For example, a user can send an e-mail from a laptop computer and have it received by a cellular phone which is the recipient's currently active device. The SM paradigm requires that the recipient of the message be located through an intelligent calendaring facility and the message be tailored to the recipient's active device user interface. As such, SM is not an easy endeavour.

Why is SM important?
Seamless messaging allows users to work in distributed personal workspaces and have messages created and delivered how, when and where they wish. In today's distributed environments users are often faced with multiple messaging environments (e.g. voice mail, e-mail or fax mail) that do not interwork intelligently, if at all. The results are that users are overwhelmed with hundreds of messages in different messaging boxes. Many of these messages are unimportant or even considered junk mail. As a result, users are calling for methods of seamlessly integrating these message boxes and of intelligently filtering their contents.

Why are agents essential for SM?
Agents are active computational entities that are persistent, can perceive, reason and act in their environments and can communicate with other agents [Huhns and Singh 95]. Agents are ideal for applications that require some form of distributed intelligent cooperation. As such SM agents can be clients, servers or messengers in a distributed environment.

As illustrated in Figure 1, agents can form the backbone of seamless messaging. Specifically, six types of agents are required to achieve this application. They are user agents, user agent surrogates, message transfer agents, server agents, application agents and device agents. Each of these is described in detail below.

SM agents require communication mechanisms so that they may interact. They communicate by exchanging messages with a triplet of the form:

message(From_agent_id, To_agent_id, Body_of_message)

The body of the message can be a script or data to be acted on by the receiving agent. Specific examples of its use are included with the description of user agent surrogates (the SM messenger agents) below.

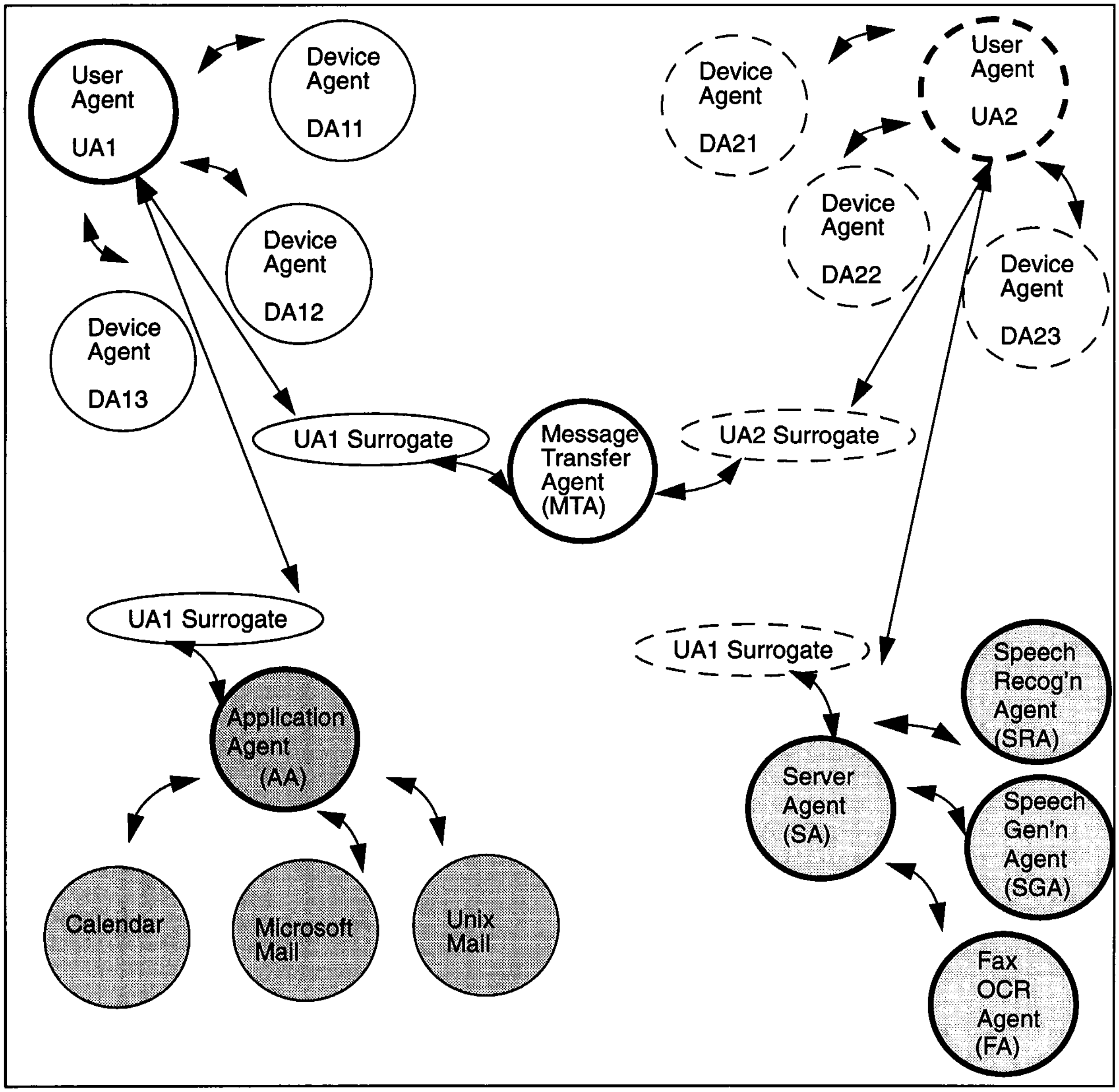

Figure 1: Seamless Messaging Agents

3.1 User Agents

User agents (UAs) manage the user's environment. These coarse-grained agents track user preferences in seamless messaging. This tracking can be based on the user explicitly instructing the agent with a rule or a script for some action. It can also result from the agent observing the user's actions and intelligently learning what the user is doing. For example, the agent could observe that the user deletes all messages with the subject line including "CFP" (call for papers). The agent could then propose to the user that it delete all CFP messages. In this manner the UAs are adaptive agents that are expected to learn from their environment.

User profiles include rules on how to manage the messaging inbox of voice, e-mail and fax. They may also include scripts for priority messages (e.g. "locate me urgently if the message is from my boss Ed"). The UA may also include scripts on how to handle specific message types. For example, a fax message would trigger a user agent surrogate to interact with the server that includes the optical character recognition (OCR) software. Once the fax is converted to an e-mail message, it can then be interpreted for priority or action (e.g. deletion, filing or forwarding).

In addition, UAs may include calendar coordination information. Thus, if an urgent message arrives, a UA surrogate interacts with the Calendar application agent to get a best guess on where the user is expected to be. This information is combined with the latest active device information polled from the user's Device Agents to attempt to deliver the message.

3.2 User Agent Surrogates

The UA Surrogates act as messengers or intermediaries of the UA and are given specific tasks. They are given scripts that direct them to other agents with specific goals. These UA surrogates are mobile and travel across networks in response to UA requests. These fine-grained agents are aimed at supporting wireless PCS (personal communication systems) messaging which interacts with mobility devices such as PDAs, cellular phones, wireless laptops and pagers.

As the user creates a message, the UA surrogate carries it to the Message Transfer Agent (MTA) which examines who the recipient is.The MTA identifies the recipient UA and the recipient UA sends a surrogate to take delivery of the message. If the message is in a format that is unacceptable to the UA, a UA surrogate is sent to the Application Agent (AA) with the message requiring translation. Users are assumed to be active on mobile devices (e.g. a cellular phone), stationary devices (e.g. a Sun workstation) or a combination. The user Device Agents (DAs) track user activity and are ready to report status to the UA on demand.

For example in figure 1, UA1 may have sent a Microsoft mail message to UA2. The MTA informs UA2 of the message using the following communication scheme:

message(MTA_id, UA2_id, Incoming_message)

UA2 sends its surrogate to receive the message. After UA2 receives the message, it recognizes from the MTA tag that it is a Microsoft e-mail. UA2 then sends another surrogate to AA to have the message translated to Unix mail. The communication would be:

message(UA2_id, AA_id, [Script[translate_email(Microsoft_m, to, Unix_m)]]))

The "Microsoft_m" variable is actually the file that is to be converted to Unix and returned by binding it to the "Unix_m" variable. Furthermore, AA acts as a gateway between the two environments and translates any attachments as instructed by the UA2 surrogate on what file formats its user can accept.

Another possibility is that UA1 sends UA2 a voice message that is tagged as urgent. UA2 polls its devices and finds that its user is at a text-only device. It then triggers a surrogate to interact with the Server Agent to have the voice message recognized by a speech recognizer. The text of the voice message is then received by the UA2 surrogate and delivered to the active device.

Yet another possibility is for UA1 to have sent UA2 an e-mail that is tagged urgent. Again, UA2 polls the device agents and finds its user only active on the cellular phone. A UA2 surrogate asks the server agent to generate speech from the text. This digitized speech is then transmitted to UA2's cellular phone device agent for playback to the user. Note that if the e-mail is too long, UA2 would have to first send the message to an AA specialist to filter the text to its key concepts.

3.3 Message Transfer Agents

Message Transfer Agents (MTAs) mediate in the delivery of messages between heterogeneous systems. These coarse-grained agents receive messages from the UAs and interpret the header information. The interpretation is then transmitted to the recipient UA surrogate. Thus, messages are tagged with a type (voice, e-mail or fax) and in some cases (e-mail) with a subtype. The subtype indicates the e-mail environment the message was created in (e.g. Unix mail or Microsoft mail). MTAs are similar in function to the MTAs proposed in the X.400 messaging standards.

3.4 Server Agents

Server Agents (SAs) are coarse-grained agents that mediate in the performance of a specialized function. Specialities in messaging include speech recognition (for voice to e-mail conversion), speech generation (for converting e-mail to voice), and fax optical character recognition (for converting fax to e-mail).

Server agents may also provide a seamless interface for hopping between networks. This is a key requirement in moving between public networks for wired or wireless voice and data. In most private networks movement between voice and data can be achieved locally. This will be made easier with the advent of voice cards and telephony interfaces that will reside in workstations and laptops.

3.5 Application agents

Application Agents (AAs) are coarse-grained agents that mediate between UA surrogates and end user applications such as mail (e.g. Microsoft mail, Unix mail, Eudora, etc.) and calendaring. They enable UA surrogates to translate between mail environments and encode attachments as needed. They also allow UA surrogates to query electronic calendars that may be directly incompatible.

Application Agents may also provide an interface to specialized applications. Filtering of message content is such an application. Such filtering may allow the distillation of pages to a set of specific concepts. A primitive form of mail filtering occurs at the UA level where rules and scripts are triggered based on the message header that includes the sender and the subject. However, filtering out the concepts rather than only the keywords in a message is a more sophisticated speciality that is ideal for an AA.

Another specialized application is people finding. Here, a combination of calendar and active device information can be used to isolate users. Thus far, active device information has only been used successfully to find people [Ferguson and Davlouros 95].

3.6 Device Agents

Device Agents (DAs) reside in the devices that a user receives messages with. DAs include a cellular phone, a PDA, a laptop, a telephone, a workstation or a fax. DAs interact with UAs and may share the physical device with a UA (e.g. a workstation). DAs can be grouped into either personal (owned by an individual) or public (shared by several users) devices. For example, a cellular phone or a laptop may be a personal device whereas a workstation may be shared.

Device agents are coarse-grained and they have two key roles. They maintain a log of activity by the user so that they may respond to UA surrogates polling for active devices. They more importantly interact with the UA and the UA surrogates to create, send and receive messages. The UA surrogates interact with the Application Agents and the DAs as the user creates the messages.

4. Tradeoff Between Communication and Compilation in Seamless Messaging Agents

Seamless Messaging requires realtime performance. This is why it is essential for this distributed application to take advantage of agents that naturally partition the application. The SM specialists (such as the server and application agents) are called on when necessary by the UA surrogates with specific goals.

The Seamless Messaging agents mix coarse and fine-grained agents to take advantage of the tradeoff between communication and specialization. In the MANA agents [Weiss et al. 95], coarse-grained agents are programmed with pre-compiled scripts. To change the agent, it is necessary to take it off line and redefine its script and then re-compile it. This is unacceptable in any dynamic evolving application such as SM where the agent environment cannot be assumed to remain static.

SM agents by definition have a higher degree of communication so that they may coordinate their activities. They communicate their intentions, goals, results and state in the form of the message triplets described earlier. The messages may incorporate status, scripts (e.g. for goals or queries to the receiving agent). SM agents communicate more, thus raising the cost of performance. However, the UAs are adaptive and their functionality is expected to evolve with user preferences.

5. User Interface Mediation in Seamless Messaging

Mediating between the different user interfaces of the devices used in seamless messaging is a difficult problem. As illustrated in figure 2, this includes mediating between messages received on a full-screen (e.g. a 19 inch layout) workstation and a wireless phone (e.g. a 2 line character display). This is compounded by the preference of users for heterogeneous messaging applications that include a variety of e-mail applications, voice messaging and fax. This is even made more complex by mediating across wired and wireless networks which have differing user characteristics including noisy reception and high error rates in wireless networks.

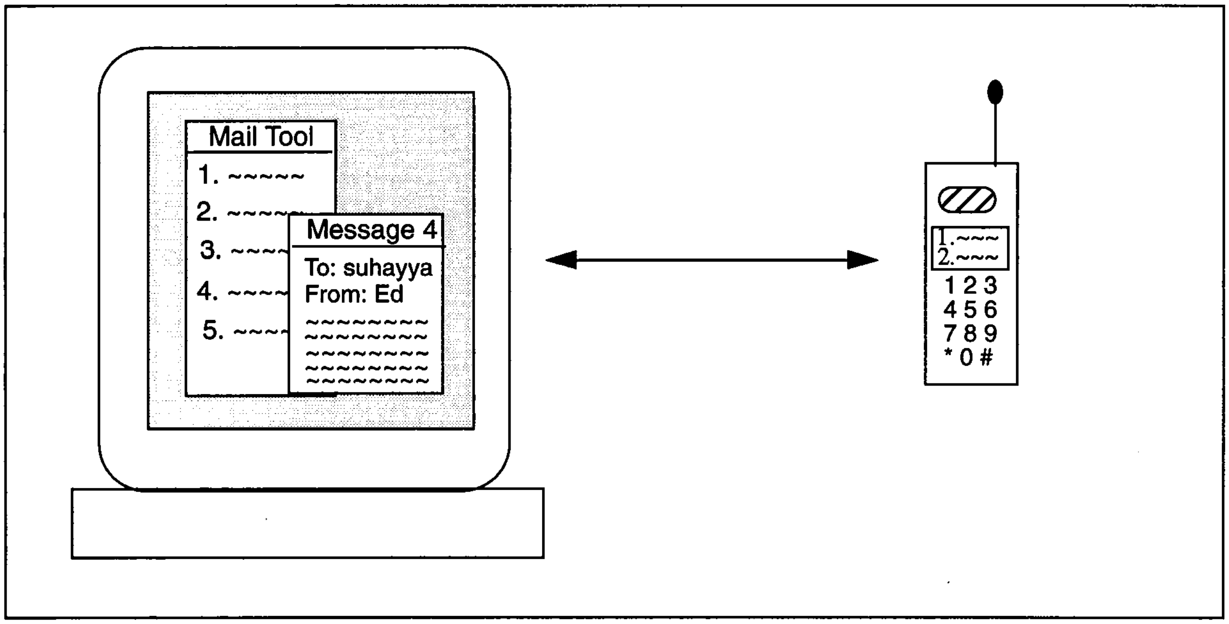

Figure 2: Required User Interface Mediation in Seamless Messaging

In Seamless Messaging this can be addressed somewhat through the use of preferences in user agents. A user agent profile could include layout and modality preferences, i.e. "always try and deliver to my laptop versus my cellular phone" and "do not convert voice mail from a cellular phone to text, just deliver it as digitized voice over the laptop speaker". These do not have to be made explicit by the user but may include defaults that are reasonable in cooperating between humans and their devices. A set of heuristics may be encoded in all user profiles that incorporates some of the key aesthetics and general usability factors in SM.

Device agents, message transfer agents and application agents can also help mediate between the conflicting user interface requirements. In the earlier example, the MTA was used to inform the user that an e-mail message targeted to them is not in the format of their mail environment. The message was then translated by an application agent. The device agent for a cellular phone could refuse to accept a text message that exceeds its buffer size thus avoiding the delivery of a long e-mail to a cellular phone. However, the UA has to be intelligent enough to ask the mail filtering specialist to distil the long e-mail to the key concepts so that an urgent e-mail can still be delivered to the cellular phone.

Thus, to attempt to meet user interface mediation requirements in seamless messaging one requires:

- user agent default and customizable profiles to include delivery and modality preferences
- device agents must include the ability to bounce messages back to UA surrogates based on logistic (e.g. pages of text to be delivered to a 2 line cellular phone) incompatibilities
- device agents must also include the ability to mediate conflicting modalities (e.g. delivering voice to a device that cannot process it returns an error to the UA that requests the text equivalent of the voice)

- message transfer agents must identify the content and required delivery medium of a message and transmit this information to the recipient's UA surrogate. The UA surrogate must then isolate the active device of the user and wait for instruction from the recipient UA. The recipient UA will then instruct the surrogate to have the message converted to text or voice as the context dictates.

6. Future Work and Conclusions

The Seamless Messaging application is currently being implemented with a combination of tools such as Lotus Notes, Phone Notes and Netscape. A UA mail filter has been implemented in Netscape and Lotus Notes. Furthermore, a prototype integrating email and voice mail preferences with user directed action for filtering has been implemented in Lotus Notes. A full implementation that would include messaging between a variety of e-mail environments and voice mail is expected to be completed in an 12 month timeframe.

Adaptability is an essential characteristic of the SM agents. Three forms have been described in the paper. The first mechanism requires the user agents to observe what the user does with their messages (be they voice, email or fax). Based on these observations, action scripts will be generated and sent to the user for approval on actions to be taken for incoming messages. The second mechanism is the filtering of message content so that messages may be delivered to a very different device than the one where the message originated (e.g. a long multimedia desktop message has to be decomposed and filtered so that a short equivalent can be delivered to a cellular phone). The third adaptability mechanism deals with the assumption that the user is mobile and must be located for urgent messages. The user is located using a combination of electronic calendaring information and latest active device information.

This paper has described the difficult problem of seamless messaging in heterogeneous communication environments and the design of cooperative agents to address the problem. The seamless messaging agents include coarse-grained and fine-grained agents. Seamless messaging requires real-time performance which necessitates a tradeoff between agent communication, adaptability and intelligence to achieve a variety of tasks. User Agent Surrogates are the fine-grained agents used to lighten the communication load between the User and Device Agents and the distributed specialist agents such as the Server, Application and Message Transfer agents. User interface mediation between users and device agents is also being studied to enhance the implementation.

References

[Abu-Hakima et al. 95] S. Abu-Hakima, I. Ferguson, N. Stonelake, E. Bijman, and R. Deadman. A Help Desk Application for Sharing Resources Across High Speed Networks Using a Multi-Agent architecture. Proceedings of the workshop on AI in Distributed Information Networks, IJCAI-95, Montreal, Quebec, August 19, 1995. pp. 1-9. (available from editorial@ai.iit.nrc.ca)

[Ferguson and Davlouros 95] I. Ferguson and J. Davlouros, On Establishing Multi-Sensory Multi-Channel Communications Among Networked Users. Proceedings of the workshop on AI in Distributed Information Networks, IJCAI-95, Montreal, Quebec, August 19, 1995. pp. 1-9. (available from editorial@ai.iit.nrc.ca)

[Gray et al. 94] T. Gray, E. Peres, D. Pinard, S. Abu-Hakima, A. Diaz, and I. Ferguson. A Multi-Agent Architecture for Enterprise Applications. In *Working Notes AAAI-94 Workshop on Artificial Intelligence in Business Process Re-engineering*, Seattle WA, July 1994, pp. 65-72. NRC 37140.

[Huhns and Singh 95] M.N. Huhns, and M.P. Singh, Cooperative Information Systems, IJCAI-95 tutorial Notes, August 21, 1995, Montreal, Quebec, Canada, 68 pages. Published by IJCAI.

[Huhns et al. 93] M.N. Huhns, N. Jacobs, T. Ksiezyk, W. Shen, M. Singh, and P.E. Cannata. Integrating Enterprise Information Models in Carnot. In *Proceedings of the First International Conference on Intelligent and Cooperative Information Systems*, 1993.

[Laufmann 94] Laufmann, S. The Information Marketplace: The Challenge of Information Commerce. In *Proc. of the Second International Conference on Cooperative Information Systems*, May, 1994, pp. 147-157.

[Reinhardt94] A. Reinhardt. The Network with Smarts. *Byte*, October, 1994 pp. 51-64.

[Rosenschein and Zlotkin 94] J.S. Rosenschein and G. Zlotkin. Designing Conventions for Automated Negotiation. *AI Magazine* 15(3), 1994, pp. 29-46.

[Weihmayer and Velthuijsen 94] R. Weihmayer and H. Velthuijsen. Application of Distributed AI and Cooperative Problem Solving to Telecommunications. In *Proc. of the International Workshop on Distributed Artificial Intelligence*, Lake Quinault WA, July 1994.

[Weiss et al. 95] M. Weiss, T. Gray, and A. Diaz. A Multi-agent Architecture for Networking Applications. Submitted to *First International Conference on Multiagent Systems*, San Francisco CA, June, 1995.