On the Use of IP Multicast to Facilitate Group Communication between Mobile Agents

Syed Waqar Shah, P. Nixon and R.I. Ferguson
University of Strathclyde, Department of Computer and Information Sciences, Livingstone Tower, 25 Richmond Street, Glasgow G1 1XH, UK. Tel. +44 (0)141-548-3522, Fax. +44 (0)141552 5330.
{Syed.Shah, Paddy.Nixon, ian.ferguson}@cis.strath.ac.uk

Abstract

The use of multiple agents (in patterns such as master-slave and embassy)[10] is a common technique in agent-based systems. Maintaining coordination and communication between groups of agents, whilst relatively simple in the non-distributed paradigm, can impose a significant overhead on the developers of distributed/mobile agent systems. In this paper we report the design, implementation, and evaluation of a group communications mechanism for multiple agents systems.

1. Introduction

An architecture to simplify coordination and communication between groups of agents (known as AWMG – agentspace with multicast groups), based on IP multicast messaging, has been designed and implemented. It has been evaluated by employing it in the development of middleware (SeSaS - Sensor System with agentSpace) to support a real-time sensor environment intended for use in context –aware computing.

Throughout this paper applications that are built using mobile agents are referred to as mobile agent systems (MAs), those built with multiple agents as multiple agent systems (MAS) and those combining the two approaches as multiple-mobile agent systems (MMAS). In addition the underlying middleware that provides MA facilities is referred to as a mobile agent framework (MAF).

2. The Problem - Group communication and coordination

MAFs can trace their origins back to systems such as Obliq [2], Messenger [3] and Aglets [4]. The strengths of the MA approach in the development of distributed systems (e.g. resilience in the face of partial disconnection, circumventing network latencies, load balancing, dynamic deployment, etc.) have been enumerated by Lange and Oshima [1].

By combining the features of mobility and multiplicity, multiple mobile agent systems (MMAS) emerge as a particularly appropriate abstraction for the class of applications requiring autonomy and dynamic configuration/adaption. One such area is mobile and ubiquitous computing where small footprint devices, rapid evolution of the environment and unreliable connectivity demand dynamic and adaptive software.

In the field of MAS the problems of coordinating groups of agents are well understood. Indeed patterns describing standard solutions are part of the literature [10]. When, however, the agents participating in such cooperation are mobile agents, then a significant overhead is placed upon the system developer in terms of implementing some kind of message passing/state sharing scheme in order to maintain coordination.

Traditionally, mobile agents are meant to interact by migration followed by local interaction. This is not always the most appropriate (in terms of time and network traffic) means of interaction: When the amount of data being exchanged in a given interaction between two remote agents is significantly smaller than the size of the agents involved, it is more efficient to revert to a message passing or RMI-like solution.

In current MAFs the implementation of such non-local communication between agents is left largely to the application developer [7, 17]. In particular, current MAFs provide no built in support for the sharing of state between groups of MAs or for the distribution of messages amongst groups of dispersed agents. The ad-hoc solutions adopted are typically based on the use of multiple unicast transmissions. In addition to the obvious inefficiencies (in terms of duplicated messages and network traffic) the lack of built in support is a deficiency. In order to coordinate amongst a group of agents, the MAF needs the ability to: a) filter incoming messages and decide on which agents should receive them; and b) route the messages to the remotely located agents.

3. Solution - IP Multicast messages among mobile agent groups

Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)
0-7695-2101-0/04 $ 20.00 IEEE

The key abstraction in tackling this problem is the *agent group* and the transmission of messages to groups (using IP multicast) such that all constituent member agents receive a message sent to a single agent group. The filtering of messages can be based on membership of groups, (requiring the implementation of membership management). The routing of messages can be based on the idea of a 'group agent' acting as a proxy for its constituents to whom it routes all messages (requiring an agent location subsystem). Stated simply, an agent group needs to keep track of who should get a message and where those recipients are.

The approach of AWMG is to use IP multicast as the transmission mechanism (for reasons of network traffic efficiency), the location service and also the group membership management mechanism [6, 8].

3.1. Multicast operations in MMAS

We now consider how the use of IP multicast can aid in the design of a MAF to support the creation of MMAS.

3.1.1. Transmission. By employing IP multicast as the means of message transmission, our solution gains the ability to have multiple hosts receive a single message thus avoiding the proliferation of network traffic inherent in the unicast methods commonly used. With a traditional multicast setup however, transmission and reception of messages (and group membership management) operates at an inter-host level. For multicast between mobile agents to function there must exist a second level of routing (within an individual host – inter-agent) to individual agents –see figure 1.

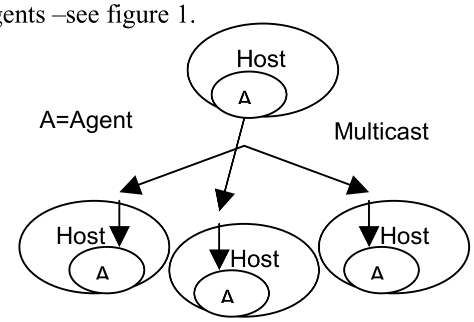

Figure 1. Two levels of message routing

3.1.2. Routing/Location service. At the inter-host level the task of routing messages is handled entirely by the multicast IP layer. A sending agent (or indeed the MAF) need not know the location of the agents in a group in order to transmit to them a multicast message. As long as the agent's current host has registered as a member of the appropriate IGMP multicast group it will (within the bounds of network reliability) receive any message multicast to the group address. Once a message has been received by a host it must then be routed to the individual agent that it is intended for. At this point the system can revert to the internal message routing/location scheme of the underlying MAF.

3.1.3. Group membership management. The group membership management services in AWM must integrate, seamlessly, the joining and leaving of IGMP multicast groups by hosts, with the joining and leaving of agent groups by individual agents in a manner that's transparent to the application programmer (who deals solely with the "group" abstraction).

3.1.4. Summary. By building group management and communication facilities into a MAF, the intention is that the amount of coding effort required to produce a multi-mobile agent based system is reduced. Having outlined the design we now turn to a discussion of the implementation of the technique followed by an evaluation of its performance.

4. Implementation

In this section the details of how multicast transmission, location services and group management have been made available to an agent whilst maintaining transparency to the developer are discussed.

The approach taken was to enhance an existing MAF - agentSpace [5] as it already provides inter-agent remote communication facilities in the form of asynchronous messaging and agentRMI (location transparent remote method invocation). These two facilities were designed for one to one agent-agent communication and are strictly unicast based.

There are two possible ways of achieving group communication with mobile agents: agent groups or messages groups. In the first case, clones of a mobile agent are multicast to a group of mobile and static agents using the multicast capable protocol, i.e. agent-mobility is implemented over a multicast connection [9]. AWMG supports the second type of group communication using IP multicast where a message is sent to a group of mobile and stationary agents using IP multicast and the group is identified by IP multicast address from class D address space.

The addition of group based multicast messaging involved the addition of several classes and requires the adoption of certain patterns by the MMAS developer. The following discussion first presents those classes and then examines their operation over the lifecycle of a group from both an MMAS developers view and the internal working view.

4.1. Basic abstractions

The agentSpace MAF [5] (in common with many other

Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)
0-7695-2101-0/04 $ 20.00 IEEE

MAFs) supports the concept of mobile objects/agents (via its MobileObject class), and an agentServer process space in which they exist.

4.1.1. MulticastGroup. The *MulticastGroup* is the primary abstraction and can be thought of as a proxy for a group of mobile agents. This class extends MobileObject so that it is itself potentially migrateable and has the attributes of an agent such as a globally unique identifer (GUID) and the ability to have its methods invoked remotely via agentRMI.

The *MulticastGroup* acts as a wrapper to the underlying Java multicast layer. At the application level, any agents which wish to send multicast messages must be of this type.

4.1.2. MulticastAgent. *MulticastAgent* extends the MobileObject class. At the application level, any agent which wishes to receive multicast messages must be of this type. Systems features include: group creation, joining, transmitting/receiving messages, migration, leaving a group, and group teardown. As can be seen from the preceding discussion, the classes MulticastGroup and MulticastAgent hide from the developer most of the complexity involved with implementing multicast communication between groups of agents. Together they present a higher-level abstraction to application developers and their use simplifies the development of MMAS applications. The following section presents an example of the use of AWMG in the development of some middleware to support ubiquitous computing is as evidence of the efficacy of the approach.

5. Evaluation of the approach

The implementation of AWMG has been evaluated by using it to build SeSaS (*Se*nsor *S*ystem with *a*gent*S*pace), a middleware system for collecting and distributing real-time data from a sensor system.

5.1. Background

SeSaS is intended to support the idea of a "smart room" – one that is replete with a variety of networked sensors and a wireless network. The network enables communication between all manner of (radio capable) computing devices from traditional desktop workstations through mobile devices such as handhelds, active badges and palmtops to small systems embedded in "smartifacts" and ultimately "smartdust". The envisaged scenario is one where devices are continuously arriving and leaving the environment, (hence the choice of mobile agents as the basic unit of decomposition) dynamically forming a constantly changing computing substrate. The role of the SeSaS architecture is to route real time data from the sensor system to whatever agents/devices exist in the environment at any given moment. Any agent interested in data from a particular sensor or sensor group may join (or dynamically create) a multicastGroup associated with that particular entity and thus receive appropriate multicast sensor data.

5.1.1. The SeSaS architecture. This architecture utilises AWMG in order to exploit the advantages of mobile agent technology [5]. Figure 2, below, shows a high level view of the SeSaS architecture.

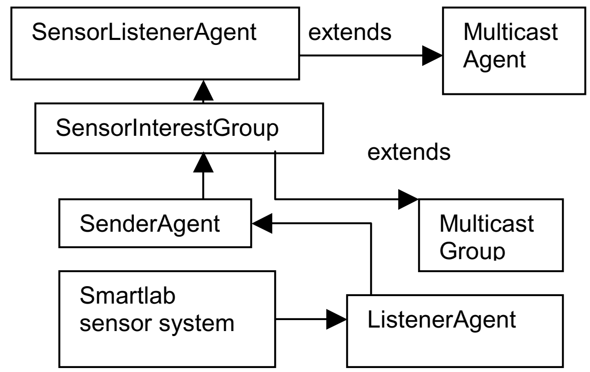

Figure 2. High-level view of SeSaS architecture

A layer of (multicast enabled) mobile agents is introduced over the sensor system layer. Whenever a sensor is triggered by some event, a message representing the sensor data is created in XML form. The message is sent to the *sensorInterestGroup* that represents the set of agents interested in data from that sensor. As mobile devices arrive and leave the smart room, their corresponding *sensorListenerAgents* dynamically join and leave *sensorInterestGroups*. The discovery protocol that introduces agents to groups is beyond the scope of this paper. A *sensorInterestGroup* may receive sensor events from more than one sensor – this allows the composition of sensors into logical groups such as *temperatureSensors* or *CoffeeAreaSensors*. Thus a group of agents may receive data from a group of sensors. The configuration of the sensors and groups is created at system start time from an XML file describing a static startup configuration. This can be dynamically altered during execution.

6. Related work

Work on agent communication systems is almost entirely concerned with the application level content of messages between agents usually in a non-distributed context. The infrastructure by which the messages travel is simply assumed to exist. Typical of this is the work on KQML [12], which has the field of artificial intelligence and speech act theory as its background. Similarly the KIF (knowledge interchange format) language [13] has been

Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)
0-7695-2101-0/04 $ 20.00 IEEE

widely adopted as a means of agents describing their knowledge to one another but again, mundane concerns such as how the descriptions are passed between agents are left unaddressed. An excellent review of other such high-level work can be found in Huhn and Stephens [14]. In comparison to such widely accepted protocols, standards for inter-mobile agent communication are in the nascent stages: Labrou et al. [11] recommends use of Agent Communication Languages whilst the Object Management Group's (OMG) Mobile Agent Facility [15] implicitly proposes the use of CORBA's communication framework. One of the FIPA draft specifications [16] requires that agent systems provide an Agent Communication Channel (ACC). Such a channel forms connections between agents both on a single host and across hosts. The specification is entirely at the logical level and makes no mention of how such a facility is to be implemented. LoudVoice [18] uses a multicast strategy but lacks the agent as proxy feature of AWMG. Barbeau [9] has discussed three mobile agent migration strategies in his paper from a bandwidth perspective. Whilst interesting, this work does not address the issue of using IP multicast for communication between agents. In [7], Baumann et al. discuss session-oriented and event-oriented communications. The system does not support any type of group communication.

7. Conclusions

IP multicast based group communication between mobile agents represents an elegant solution to the problems of coordinating the operation of mobile agents. MA and MAS have gained acceptance in their respective fields of Artificial Intelligence and Distributed Systems. Their fusion in the form of MMAS has been hampered by the lack of support offered by existing MAFs for communication and coordination of groups of mobile agents. We have shown that by building-into a MAF the support for group communication and management based on IP multicast techniques we can encapsulate the complexity exhibited by the ad-hoc solutions currently employed. The novelty in our proposed technique lies in the application of IP multicast to the problems of communication amongst groups of migrating mobile agents. Future work will focus on a rigorous analysis of performance characteristics, on facilitating reliable channels, and on group based remote method invocation.

References

[1] Lange, D.B. and Oshima, M., "Seven good reasons for mobile agents", *Communications of ACM*, 42(3). March 1999. pp. 88-89.

[2] Cardelli, L.,"A Language with Distributed Scope", *Computing Systems*, 8, 1 27-59.

[3] Tschudin, C.F., "The Messenger Environment M0 – A Condensed Description", In Mobile Object Systems: Towards the Programmable Internet (MOS '96), Ed. Vitek, J. and Tschudin, C., *Vol. 122 of LNCS*, Springer-Verlag, Berlin, Germany, pp. 149-156.

[4] Lange, D.B., Oshima, M., Karjoth, G. and Kosaka, K., "Aglets: Programming Mobile Agents in Java", In 1st Intl. Conf on Worldwide Computing and its Applications (WWCA '97), Ed. Masuda, T., Masunaga, Y. and Tsukamoto, M., *Vol 1274 of LNSC*, Springer-Verlag, Berlin, Germany, pp. 253-266.

[5] Ferguson, R.I. and Grey, D.J., "The agentSpace mobile agent platform" (Online) http://www.agentspace.co.uk, downloaded Aug. 2003.

[6] Hartroth, J. and Hofmann, M., Using IP multicast to improve communication in large scale mobile agent systems", Thirty-First Annual Hawaii International Conference on System Sciences-Volume 7 January 06 – 09, 1998 Kohala Coast, HI.

[7] J. Baumann, F. Hohl, N. Radouniklis, K. Rothermel; and M. Straßer (1997), "Communication Concepts for Mobile Agent Systems", In: K. Rothermel and R. Popescu-Zeletin (eds.), 1st Int. Workshop on Mobile Agents (MA'97) *LNCS* 1219, Springer-Verlag, pp. 123-135.

[8] K.P. Birman, *Building secure and reliable network applications,* Manning Publications Co, USA, 1996.

[9] Barbeau, M.," Transfer of Mobile Agents Using Multicast: Why and How to Do It on Wireless Mobile Network." Technical Report TR-00-05, School of Computer Science, Carleton University, July 2000.

[10] Hayden, S.C.; Carrick, C.; &Yang, Q., "Architectural Design Patterns for Multi-Agent Coordination", In Proceedings of the International Conference on Agent Systems '99. (Agents'99). Seattle, WA, May 1999.

[11] Labrou,Y., Finin, T., and Peng, Y., "The Interoperability Problem: Bringing together Mobile Agents and Agent Communication Languages", In Proc. Of the 32nd Hawaii International Conference on System Sciences – 1999.

[12] Finan T. et al. (1993), "Specification of the KQML agent communication language", DARPA knowledge sharing initiative external interfaces working group.

[13] Genesereth, M.R., (1991), "Knowledge Interchange Format", In Proceedings of the Conference on the Principle of Knowledge Representation and Reasoning, pp599-600, 1991

[14] Huhns, M.N., and Stephens, L.M, (1999), " Multiagent Systems and Societies of Agents", In Multiagent Systems, ed. Weiss, G., MIT Press, 1999.

[15] Object Management Group, (2000), "Mobile Agent Facility",(Online)http://www.omg.org/technology/documents/formal/mobile_agent_facility.htm, downloaded Oct. 2002

[16] Foundation for Intelligent Physical Agents (FIPA), (2001), "FIPA Agent Management Specification - XC00023", (Online), http://www.fipa.org/specs/fipa00023/, downloaded Oct. 2002.

[17] S. Mishra, "Agent Fault Tolerance Using Group Communication", In the Proceedings of the 2001 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2001), Las Vegas, NV (June 2001).

[18] Busetta, P., Merzi, M., Rossi, S. & Legras, F. "Intra-role coordination using group communication: a preliminary report", In Proc. Workshop on ACL and Conversation Policies (AAMAS '03) Melbourne, Australia, (2003).