Agent Cooperation and Monitoring on the Internet

R. Anane
MIS, Coventry University
Priory Street
Coventry CV1 5FB, UK
(+44) 02476 887761
r.anane@coventry.ac.uk

K-M. Chao
MIS, Coventry University
Priory Street
Coventry CV1 5FB, UK
(+44) 02476 888908
k.chao@coventry.ac.uk

M. Younas
MIS, Coventry University
Priory Street
Coventry CV1 5FB, UK
(+44) 02476 888901
m.younas@coventry.ac.uk

ABSTRACT

The combination of agent technology and mobility is viewed increasingly as an alternative to client-server models. This approach suffers from a lack of robustness, especially on the server side. In this paper we propose a decentralised approach. Static agents, mobile agents and Object Request Broker (ORB) mechanism are combined into a cooperation framework. Mobile agents possess some autonomy that enables them to observe situations locally and to filter data on site. The ORB mechanism is incorporated in order to provide a robust framework to support interoperability and to optimise processes by reducing unnecessary communication. Within the framework, control is distributed between agents and ORB through a dual mode of observation. This approach has the advantage that it enhances fault-tolerance, flexibility and the integration of heterogeneous systems.

Keywords

Mobile agent, ORB, observation, locality of reference

1. INTRODUCTION

One of the most pressing tasks facing software engineers today is how to leverage effectively the resources available on the Internet. Their endeavour is, however, hampered by its lack of homogeneity. Heterogeneous hardware and software systems co-exist on the Internet. Different types of protocols preside over the exchange of information between various systems that include legacy software, object-oriented systems and now agent based platforms. In this context the wider acceptance of the object-oriented technology has helped establish a comprehensive infrastructure. Besides the enhanced flexibility that it affords, this technology offers a migration path away from a large base of legacy systems. Most of these systems, however, follow the client-server model, with a strong bias towards centralisation.

The ubiquitous nature of the Internet and the availability of competing technology models raise a number of fundamental issues:

1. Heterogeneity. The existence of disparate systems requires explicit support for interoperability so that resources can be accessed and shared easily.

2. Network traffic. The growth in the intensity and variety of applications on the Internet has resulted in a significant increase in network latency. One way of reducing the large volume in traffic is to foster a policy a decentralisation in decision making [1].

3. Access to large databases. The retrieval of information from a large variety of data servers is one of the most popular applications on the Internet. Alternatives to client-server models should provide a framework for filtering data on remote sites in order to reduce Internet traffic.

4. Dynamic requirements. The dynamic nature of the Internet leads to changing requirements and the need to customise applications to reflect new interests. This requires support for flexibility and collaborative frameworks.

5. Lack of reliability. A heterogeneous and distributed system like the Internet may be inherently unreliable. New software technologies and platforms should aim at enhancing fault tolerance and maintaining integrity. Cooperative approaches to fault tolerance are being proposed as models [2].

In this paper we present a framework aimed at addressing many of the concerns identified above, by integrating agent technology and object-oriented technology in the form of ORB mechanism and active objects. This combination has the advantage that it maintains the integrity of the system whilst supporting effectively the functions of the agents. Environment monitoring is taken as a case study for this scheme as it provides a useful context for the deployment of mobile agents.

This paper is organised as follows. Section 2 gives an introduction to agent technology. Section 3 presents the context for the application domain. Section 4 describes a cooperation framework. Section 5 presents the dual mode of observation used in the framework. Section 6 discusses issues raised by the proposed approach. The last section concludes the paper.

2. AGENT TECHNOLOGY

Although the notion of agent has presented workers in AI with a major challenge there is, however some consensus around 'the weak notion of agency' given by Wooldridge and Jennings [3]. An agent is defined as a hardware-based or a software-based computer system that is autonomous, reactive, proactive, and has social ability. Reactivity is a necessary requirement for an agent to be able to respond to stimuli and to interact with the outside world by observing it and operating on it. Autonomy and pro-activity are the two features that distinguish an agent from a mere object, and underline the knowledge level at which agents operate. While autonomy enables an agent to function without direct intervention from other agents, pro-activity emphasises the fact that an agent is goal-directed and can take the initiative. Finally social ability refers to the communication and interaction between agents.

An agent is endowed with the ability to possess a mental model of the outside world and to reason about it. This deliberative behaviour is modelled by a set of mental categories. The Belief, Desire and Intention (BDI) mental model has become increasingly popular and has been used in many applications. BDI architectures provide a clear conceptual model that integrates seamlessly reactive and deliberative behaviours [4, 5]. The mental categories form the basis for a mental state: beliefs, desires and intentions. The beliefs represent the information that the agent believes is currently true. This information can be on the agent itself, other agents or the environment. The desires or goals are the states that the agent wishes to achieve. An intention involves the selection of a particular plan and a commitment to its execution.

Two types of agent can be distinguished, static and mobile. A mobile agent is a special type of agent that can move from one host to the other over the network to carry out the task that it was designed for. Locality of reference, flexibility and customisation are the main reasons for launching mobile agents. Flexibility and customisation allow for the dynamic deployment of software. Locality of reference applies to information retrieval situations, and especially when a large volume of data needs to be processed. The mobile agent can be sent to large data sources and filter through them locally. It is this particular application that is directly relevant to this framework [6].

3. CONTEXT

In this section we consider a case study based on the monitoring of the environment. It sets the context for the deployment of agents.

Concern about issues such as global warming, land erosion, deforestation and pollution is being addressed by a careful and more focused monitoring. The monitoring of the environment is a dynamic activity that brings into play various components: the sites being monitored, the remote monitoring stations, the central monitoring station (CMS) and the analysis and interpretation of the data collected and the communications infrastructure. The frequency and the level of the monitoring activity may be varied in response to new conditions or in order to address new concerns. One of the most prevalent approaches to monitoring involves the establishment of a network of stations for the generation and collection of data, which is then uploaded at regular intervals to a central system where it is analysed. Large volumes of data may be transferred and may transit via the Internet in order to reach their destination. The monitoring stations may be deployed in remote areas with difficult access such as the North Pole. The central monitoring system should be able to make specific one-off queries to the monitoring station.

Once the processing of the data is complete, the analysis and interpretation of the results may depend on the availability of experts on site or in other research institutions. Interpretations may also require a second opinion from other experts or corroboration from other central units that are monitoring other aspects of the environment. Stations can be added or removed from sites, different hardware units can be deployed and different analysis software systems can be installed. These tasks are often performed manually. These changes are either dictated by contingencies or by the introduction of more sophisticated tools. New developments may lead to incompatibilities in hardware and software systems.

From the scenario above it is possible to produce a summary of the main features of the underlying system:

1. Generation of large volume of data and its transmission across the network.
2. Exchange of information between experts.
3. Use of specialised software in the monitoring stations
4. Coexistence of diverse hardware and software systems

5. Dependence of the monitoring process on the level of reliability of the system.

It is evident that there is an overlap between these features and many of the characteristics of the Internet as identified in Section 1. An architectural framework for monitoring the environment should address all these issues. Below we present a framework that meets this requirement by involving agents, objects and ORB into a dual mode of observation.

4. COOPERATION FRAMEWORK

The proposed framework connects various remote monitoring stations (PC's), a central monitoring system and possibly some database systems. All these systems have the ability to support objects and agents. The framework is supported by an object oriented software component that includes an ORB. Active entities in the system are integrated by means of high-level protocols like HTTP or RMI, and by the use of an ORB mechanism.

Two levels of operation can be distinguished within the framework. The first level is defined by the interaction between agents. The overall monitoring process is controlled by a static agent, which resides in the central monitoring system. The deliberative behaviour of the agent is modelled by a BDI architecture. The BDI conceptual model offers a clear separation of concerns that makes it amenable to customisation. The static agent is able to specialise the functions of mobile agents through the components of their BDI model. The BDI reasoning mechanism in the static agent is tuned to its environment, and is therefore able to determine the functions of the mobile agent. Its meta-plans are designed to generate the BDI plans of a mobile agent to suit the needs of the remote site. Customised software is then migrated, as a mobile agent, to the remote PC's. The mobile agent is responsible for the monitoring of the remote site and for the analysis of the data. When the mobile agent detects an abnormal situation it notifies the static agent (referred to also as the CMS) and sends it the results of the analysis. This information is available centrally and may be sent to experts around the world for information or for interpretation. Another function of the mobile agents is that they may be required to move to other database systems to retrieve relevant information.

The second level of operation involves the interaction between the static agent and the ORB. The ORB offers a number of services such as interface definition, location and possible activation of objects and communication between clients and objects. Two functions are particularly relevant to this framework. The first function of the ORB is to enable different systems to inter-operate in a seamless environment regardless of the programming languages, operating systems, hardware platforms, or locations. This characteristic allows for the integration of various database systems in the central monitoring system and diverse analysis software tools. The second function of the ORB is to maintain the status of the participating entities in the system (i.e. the deletion and creation of objects) in order to support the roles of the static agent and the mobile agent.

A high level description of the main methods of the ORB observer mechanism is given in Fig. 1.

```
Module ORBObserver
{
  interface Observer
  {
    void update(observables,message);
  }
  interface Observable
  {
    void notifyObserver(data);
    void addObserver(reference);
    void deleteObserver (reference);
    long countObserver();
  }
}
```

Fig.1 ORB Observer Schema

The ORB observer mechanism contains a number of methods for adding and removing references to objects. Objects in remote stations are expected to connect to the CMS in order to enter the framework and initiate observation. The `addObserver` method adds references to the internal table of the CMS, whereas the `deleteObserver` method removes references to objects that withdraw from the framework. Changes in the status of objects are broadcast by the `notifyObserver` method.

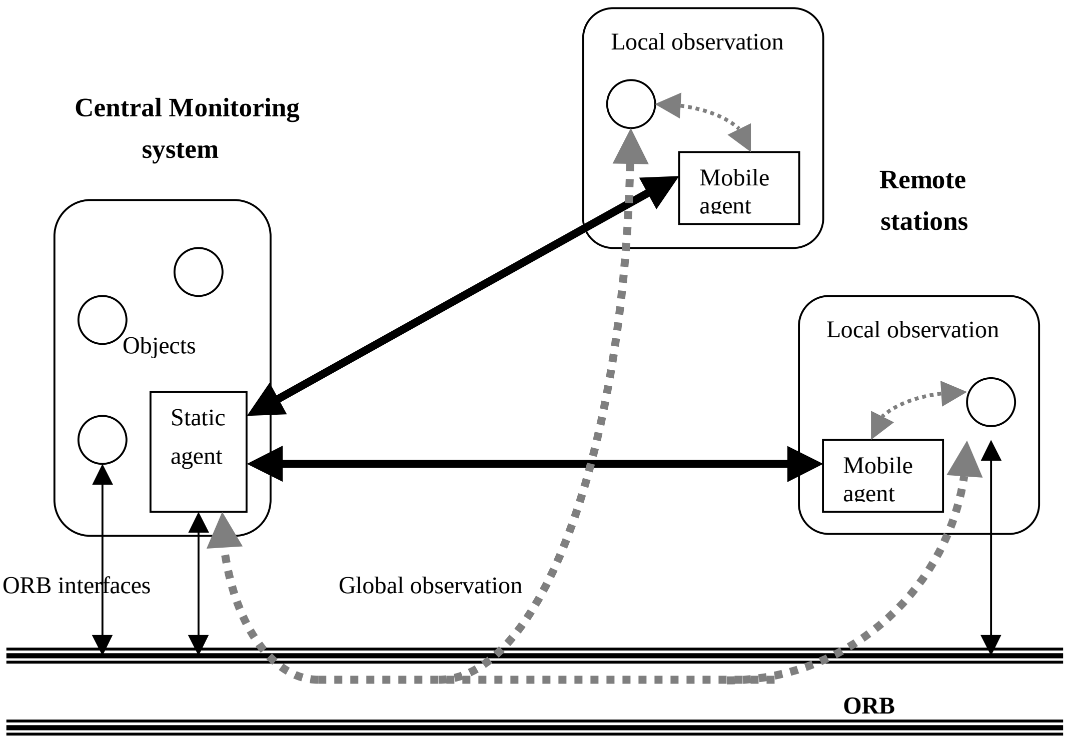

Fig. 2 Local and global observation

5. GLOBAL AND LOCAL OBSERVATION

The functions of the agents are supported by a dual mode of observation that reflects the relationship between the static agent resident in the CMS and the mobile agents: global and local observation. Global observation is the responsibility of the CMS whereas the mobile agent performs local observation. When a site requires monitoring, the CMS deploys a mobile agent to the remote site, using a high-level protocol. The diagram in Fig. 2 gives a snapshot of the interaction between various entities and the two modes of observation.

At the global level the CMS maintains a table of the active objects and in particular references to objects. Through its observer mechanism the CMS is able to keep the table up-to-date by being set to observe any object creation or deletion that takes place in the observed environment. This updating is essential, since the environment may hold objects of which the CMS is not aware. Furthermore, a lack of knowledge that an object has been deleted would lead to an exception if the CMS were to attempt to reference such a deleted object. When a connection to the CMS is made by any of the objects, the remote object reference is stored in the CMS table. References can be made to monitoring stations' PC, experts or other computer systems. In the case of a connection by the station's PC, the CMS responds by dispatching a mobile agent to the corresponding remote site. The MA is made aware of which site and data to monitor and where they are, through the remote object reference, in the table, obtained from the ORB Observer mechanism. The remote object references also provide the mobile agent with a list of locations of experts and database systems. When the connection to the CMS is about to terminate the remote object reference is removed from the table and the CMS will instruct the mobile agent to stop the monitoring activity to prevent exceptions from occurring. The CMS makes use of the information provided by the ORB to inform a mobile agent to withdraw from the remote site.

Under the local mode of observation, the mobile agent is observing the behaviour of the objects and monitoring the site. The mobile agent is able to apply the analysis software to analyse and interpret the significance of the data collected (e.g level of pollution). If an abnormal situation is detected in the summary data, the mobile agent notifies the CMS. The mobile agent sends the summary data to the agent in the CMS.

The integrity of the system is enhanced because the global observation mechanism insures that the agent resident in the CMS is aware of the status of objects in the environment. The mobile agent must check the object reference against the corresponding entry in the table held by the CMS.

Global observation is concerned with the existence of objects whereas local observation deals with their behaviour.

6. DISCUSSION

The main advantage of this scheme is that it is designed to work with existing technologies and that it integrate heterogeneous systems seamlessly into a useful system. It has also the merit that it addresses many of the issues presented in the Introduction.

Heterogeneity is dealt with by the adoption of high-level protocols and an ORB mechanism around which various technologies were integrated. Most systems are not 'agentised' [7] and it is therefore necessary to rely on existing technology. This approach enables 'objectified' legacy systems, native objects and agents to interoperate and cooperate in the collection and analysis of large volumes of data. Agent mobility is a significant departure from the client-server models in that it offers a lot of scope for a reduction in network traffic by filtering data on the site where it is generated. Specific requirements can be accommodated by the built-in flexibility afforded by the combination of BDI architecture and mobile agents.

It is possible to extend the model more widely in order to reach other resources and thus absorb more heterogeneous systems. This can be achieved by duplicating the framework, and by enabling various ORBs to interoperate via the IIOP protocol.

Enhancement of fault-tolerance is, we believe, one of the contributions of this paper. The main focus of this approach is to try to pre-empt specific exceptions from arising. It combines the sentinel approach [8] with the "citizen" approach [9] but with a focus on pre-emption. No single entity is responsible for exception handling. The robustness of the mobile agent and the reliability of the dispatching mechanism are enhanced by delegating to the ORB, a domain-independent service, the task of monitoring active objects. This aspect is critical since it is generally agreed that the non-deterministic nature of agents makes it difficult to predict their behaviour. Dealing with exceptions is even more complex for mobile agents because they usually operate in many different environments.

Fault-tolerance has also been addressed from a different perspective. In a large heterogeneous distributed system like the Internet, agents can die unexpectedly. The malfunction of the static agent in this framework will invalidate the whole set-up. A back-up mechanism, which implements a client-server model, can provide an alternative method for monitoring and analysis. It can use the ORB as a communication mechanism between objects in the CMS and other objects in the remote stations. Data can then be uploaded to the CMS (and later filtered there) by invoking specific methods from objects on the remote sites. This alternative route can be the obvious choice if the deployment of agents is too expensive or too complex.

The overall framework brings to the fore many issues related to the deployment of multi-agent systems, in general, and mobile agents in particular, especially in a heterogeneous setting. The agents operate in a hybrid environment populated by objects and legacy systems. The need to interact with 'non-agentised' components may however limit their ability at the social level. An object-oriented environment does not support autonomy let alone proactiveness. An object, for example, is deterministic to a large extent and has no control over the invocation of its methods.

With respect to the cooperation between agents, the type of relationship in which the main static agent and the mobile agent are involved highlights two aspects in the management of agents: dependence and role [10]. Within the organisation the main agent acts as the manager whereas the mobile agent is the delegate. This hierarchical organisation is relatively easy to implement and it provides an example of cooperation and social ability that agents are expected to display. It does however constrain the autonomy of the mobile agent. The agents do not engage in negotiation [11] and their roles are asymmetric. The relationship is also reflected in the way plans and meta-plans are used. When it is performing its monitoring task the mobile agent is mostly in reactive mode. Although the mobile agent may perform some reasoning it is the main agent that displays the full range of proactiveness, reactivity, autonomy and social ability. The mobile agent is cast into a structural relationship that is designed to guarantee a specific behaviour. This reduction in autonomy is, however, offset by the advantages that accrue from an enhancement in fault tolerance and flexibility.

The concern over the lack of security on the Internet is another issue that is impeding the widespread adoption of agent technology. One common view is that the Internet is inherently insecure because control is distributed. More specifically, a faulty or malicious agent can cause considerable damage to a host. Ensuring that mobile agents are subject to some authentication scheme is a critical factor in their adoption [12].

The relatively open architecture of the mobile agent is aimed at supporting adaptability and flexibility. Open architectures may, however, be subject to breaches of security. One fundamental assumption within the above framework is that the monitoring stations and other computer systems are trusted hosts. This assumption may limit the application of such a scheme in e-commerce, for example. Researchers are working on mechanisms for protecting mobile agents from attacks by malicious hosts. These attacks may affect the information that the agent is holding as well as its functionality, namely the reasoning mechanism. Many schemes are being considered to protect agents, including encryption, traces and execution of encrypted polynomial functions [13].

7. CONCLUSION

The proposed framework has underlined the fact that the integrity of a distributed system may require two levels of observation. It has also shown that a combination of agent technology and ORB mechanism can reduce communication traffic considerably by exploiting the proactive, autonomous and social nature of agents.

The context in which mobile agents were to be deployed has highlighted the need for a careful management of agency and a clear identification of roles. Furthermore, the development of sophisticated security methods to protect mobile agents from malicious hosts and vice versa was identified as one of the requirements for a wider adoption of schemes based on mobile agents.

Overall this particular framework offers significant benefits. Leveraging resources on the Internet can be achieved by addressing some fundamental issues raised by its distributed nature. This hybrid approach increases the responsiveness of the environment and manages also to reduce communication traffic through delegation and specialisation. The framework is generic and can be applied in many situations.

8. REFERENCES

[1] Ferber J., *Multi-Agent Systems*, Addison-Wesley, 1999.

[2] Klein M. and C. Dellarocas, Exception Handling in Agent Systems, *Proceedings of the Third Annual Conference on Autonomous Agents*, Seattle, USA, 1999, 62-68.

[3] Wooldridge and Jennings, Intelligent agents: Theory and practice. *The knowledge Engineering Review*, 10(2), 1995,115-152.

[4] Rao, S. A., & Georgeff. M. P., BDI Agents: From Theory to Practice, *Conference Proceedings of 1st international conference on multiple agent system,* 1995, 312-319.

[5] Huber M. J., JAM: a BDI-theoretic mobile agent architecture, *Proceedings of the Third International Conference on Autonomous Agents. ACM.,* 1999, 236-43.

[6] D. Milojicic, “Mobile agent applications”, *IEEE Concurrency*, 7(3), 1999, 80-89.

[7] Nwana H.S., Software Agents: an overview. *The knowledge Engineering Review* 11(3), 1996, 48-53.

[8] Hägg S., A Sentinel Approach to Fault Handling in Multi-agent systems, *Proceedings of the Second Australian Workshop on Distributed AI*, Cairns, Australia, 1996.

[9] Klein M., J. Rodriguez-Aguilar, C. Dellarocas, An exception handling architecture for Open Electronic Marketplaces of Contract Net Software Agents, *Proceedings of the 2nd ACM conference on Electronic Commerce*, Minneapolis, 2000.

[10] Jennings N.R., On Agent-based Software Engineering, *Artificial Intelligence*, 117(2000), 277-296.

[11] Chao K-M., R. Anane, J-H. Chen, and R. Gatward, Negotiating Agents in a Market-Oriented Grid, *Proceedings of the Second IEEE/ACM International Symposium on Cluster Computing and the Grid*, Berlin, May 2002, 436-437.

[12] Ono K. and H. Tai, A Security Scheme for Aglets, *Software-Practice and Experience*, 32(2002), 97-514.

[13] Sander T. Tschudin CF, Protecting Mobile Agents Against Malicious Hosts, in Vigna G., (ed) *Mobile Agents and Security*, LNCS 1419, Springer-Verlag, 1998.