16 RELIABLE COMMUNICATIONS FOR MOBILE AGENTS – THE TELECARE SOLUTION

Octavio Castolo, Luis M. Camarinha-Matos
New University of Lisbon / Uninova
Monte Caparica, 2829-516 Caparica, PORTUGAL
{lgc, cam}@uninova.pt

The mobile agents area represents an emerging technology that extends the distributed computing mechanisms and shows a high potential of applicability. However, the problem of reliable communications among mobile agents persists. In this paper a practical solution for reliable communication among mobile agents is described and its characteristics and limitations are discussed. The suggested approach is implemented in the TeleCARE platform for elderly care.

1. INTRODUCTION

Progress in agent development platforms is making this technology a serious approach for developments in complex distributed systems. While an agent can be considered as an independent software entity that shows several degrees of autonomy, running on behalf of a network user (Hendler, 1999), the mobile agent is often seen as an executing program that can migrate autonomously, from machine to machine in a heterogeneous network (Gray et al., 2000); that is, an agent with mobility property. The mobile agent concept has been applied to a variety of application domains including electronic commerce, network management, information dissemination, spacecraft, and network games (Kotz et al., 2002). Applications to remote operation (Vieira et al., 2001), including remotely operated robots, manufacturing systems, remote surveillance systems and remote elderly care, have been suggested. Another interesting area is the use of mobile agents in virtual laboratories (Camarinha-Matos et al., 2002), to share expensive equipment, or to operate machines in hazardous environments, among others.

A fundamental issue in the development of mobile agent systems is the reliability of agent-based applications. The need for reliable inter-agent communications is one of the key requirements. Since mobile agents roam on different hosts (or machines), communication among them is not a trivial issue.

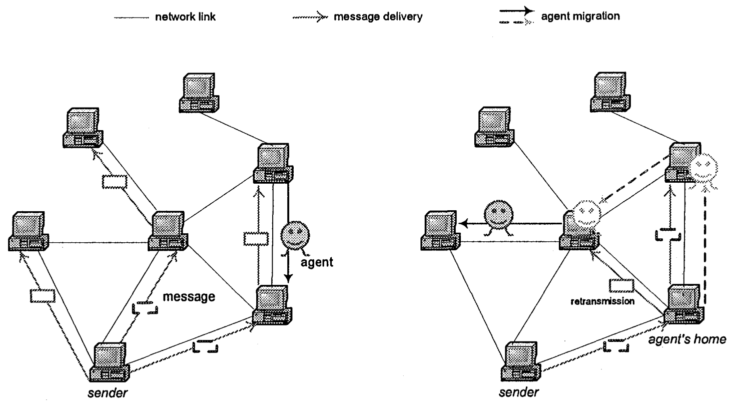
In this paper a practical solution for reliable inter-agent communication, which was developed in TeleCARE project, is presented. The paper is structured as follow: Section 2 presents the motivation for this work, where some approaches for (reliable) communication are described; in Section 3 a general overview of the TeleCARE system is given; Section 4 describes the architecture and strategy to provide reliable communication for mobile agents; and, finally, Section 5 presents the conclusions and suggests further work.

2. MOTIVATION AND RELATED WORK

A classical approach for communication in mobile agents is to offer mechanisms for local messaging, where agents talk with others only if they are living at the same host. This approach is enough for many typical applications of the mobile agent paradigm, since the agents roam among several remote hosts in order to make use of local resources in each one (Fuggeta et al., 1998). This approach can include event notification for group communication (Lange and Oshima, 1998), tuple spaces (Picco et al., 1999), among other features. There are however scenarios that require communications among remote agents. Some of these scenarios are related to mobile agent management and monitoring, in the "master-slave" case, or to accessing resources offered by other agents, in the "client-server" case. Other examples can arise within the context of a distributed application, a mixture of mobile agents and message exchange can be used to achieve different functionalities (Murphy and Picco, 2002).

Two typical approaches to message delivery are broadcasting and forwarding, both using server/hosts capabilities in order to deliver messages. A simple broadcast scheme (see Figure 1(a), based on (Murphy and Picco, 2002)) assumes a spanning tree of the network hosts that any host can use to send a message. The source host (sender) broadcasts a copy of the message to each of its neighbors, which on their turn broadcast the message to their neighbors, and so on until the leaf hosts are eventually reached. However, this does not guarantee the delivery of the message; in the process of broadcasting it might occur that when a message is being broadcasted to a host, at the same time, the destination agent is migrating in the opposite direction and destination delivery will not occur. Some approaches use a simple forwarding scheme (see Figure 1(b)) that keeps a pointer to the mobile agent at a well-known location. Upon migration, the mobile agent notifies the last place of its new location in order to make possible a future communication, leaving references (forwarding pointers) to the location where the agent currently is; or, in other variations of the scheme, the mobile agent must inform the home place in order to enable further communication. However some messages sent during the "migration and update process" might get lost.

In general, practical approaches to communications in mobile agents are based on the aforesaid. Some agent systems, such as Aglets (Lange and Oshima, 1998) and Voyager (Glass, 1999), employ a forwarding schema by associating to each mobile component a proxy object that "points" to the agent's home. Others, e.g. Mole (Baumann et al., 1997), assume that an agent never moves while engaged in communication; if migration of any of the parties involved in a communication takes place, the communication is implicitly terminated. Mole also exploits a different forwarding scheme that does not keep a single agent's home; rather it maintains a trail of pointers (forwarding pointers) from source to destination for faster contact (Baumann and Rothermel, 1998). Finally, some systems, e.g. D'Agents (Gray et al., 2002), provide mechanisms that are based on remote procedure calls, and transfer to the application developer the task of handling a missed delivery.



(a) Spanning tree broadcasting (b) Forwarding

Figure 1 – Missing message delivery in simple broadcast and forwarding schemes

Advanced approaches to reliable message delivery in mobile agents have been proposed in (Murphy and Picco, 2002), (Assis-Silva and Macêdo, 2001; Liu and Chen, 2003), (Ranganathan et al., 2000), and (Roth and Peters, 2001). In (Murphy and Picco, 2002) the approach supports uni- and multicast, but failures are not tolerated. In (Ranganathan et al., 2000) failures are considered, but the approach only supports unicast (peer-to-peer communication). In (Assis-Silva and Macêdo, 2001; Liu and Chen, 2003) an approach based on mobile groups and group communication is presented; this approach requires synchronism among mobile agents for well-functioning, considering only environments with synchronized events. In (Roth and Peters, 2001) the establishment of a public global tracking service for mobile agents, using dedicated tracking servers to storage the global name of each agent in the system, is proposed.

3. THE TELECARE SYSTEM

Due to the growing numbers of elderly population there is an urgent need to develop new approaches to care provision. Tele-assistance and provision of remote care to elderly living alone at home represents a very demanding case of a distributed system. Developments in this area have to cope with some important requirements, namely (Camarinha-Matos et al., 2004):

- Openness, in order to accommodate a growing number of new services and supporting devices.
- Support for heterogeneity, as different users have different needs and might possess a diversity of legacy systems (e.g. computers, home appliances, domotic infrastructures).

- Scalability, in order to allow the integration of a variable number of users in a tele-care community.
- Reliability of the system in terms of continuity of the service.

The mobile agents paradigm offers interesting characteristics that address some of these requirements. In fact, moving the code to the place where actions are required enables timely response, autonomy and continuity of service provision with reduced dependency on network availability and delays. Since new mobile agents can be built and deployed for remote execution whenever needed, higher levels of flexibility and scalability are achieved. By investing on the level of autonomy / decision-making capability of mobile agents, it is possible to conceive solutions that smoothly adapt to different user environments.

The convergence of a number of technologies such as multi-agent systems, federated information management, safe communications and security over Internet (Pozo et al., 2004), hypermedia interfaces, rich sensorial environments, increase of intelligence of home appliances, and collaborative virtual environments, represents an important enabling factor for the design and development of virtual elderly support community environments.

In this context, the IST TeleCARE project (Camarinha-Matos and Afsarmanesh, 2002) aimed at designing and developing a configurable mobile agents' framework focused on virtual communities for elderly support. Figure 2 shows a blocks diagram of the proposed architecture for a layer to be installed in each node of the TeleCARE organization (Camarinha-Matos et al., 2003).

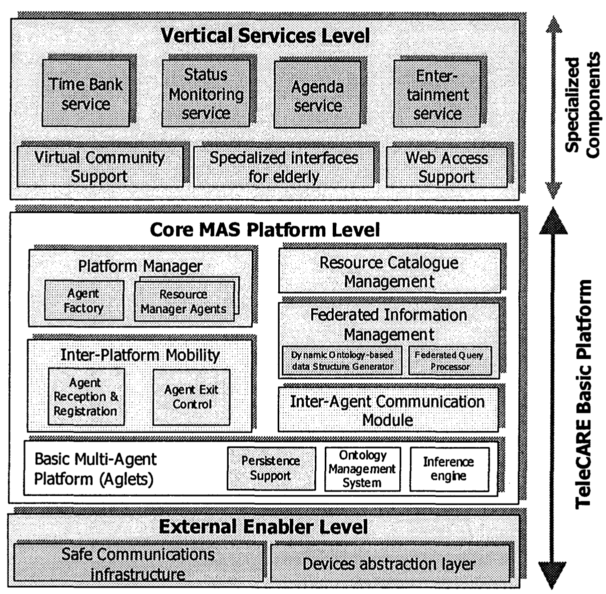


Figure 2 – The TeleCARE system architecture

The three-tier TeleCARE platform comprises:

- ❒ The **External Enabler Level**, which supports the communication and interfacing to the external devices, and other nodes.
- ❒ The **Core MAS Platform Level**, which is the core layer of the platform architecture. It supports the creation, launching, reception, and execution of stationary and mobile agents as well as their interactions / coordination.
- ❒ The **Services Level**, that consists in a variety of application services that can be added to the basic platform in order to assist the elderly, care providers, elderly relatives, etc.

3.1 Multi-Agent Infrastructure Design

The design and construction of the Multi-Agent Infrastructure of the TeleCARE project (the TeleCARE extended MAS Platform) comprises several modules of the **Core MAS Platform Level** of the TeleCARE system architecture, which is shown in Figure 3. From the multi-agents' perspective, the main modules of the TeleCARE Core MAS Platform are the following:

- Basic Multi-Agent Platform,
- Inter-Platform Mobility,
- Inter-Agent Communication Module, and
- Platform Manager.

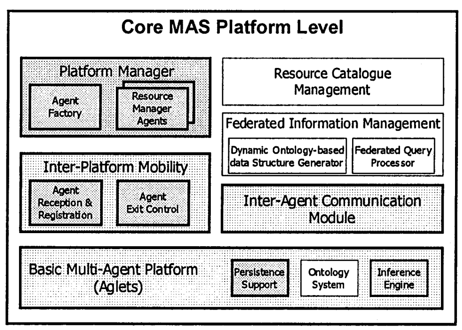


Figure 3 – Components of the TeleCARE Basic Platform

3.1.1 Basic Multi-Agent Platform

The *Basic Multi-Agent Platform* is the basic engine of the **Core MAS Platform Level**. The Aglets (Lange and Oshima, 1998) open-source system has been chosen as the multi-agent development tool mainly because it provides a strong support for agent mobility. The Aglets system is complemented with three additional modules that extend its functionality: (*i*) an *Ontology System* for knowledge modeling, using Protégé (Protégé-200); (*ii*) an *Inference Engine*, which uses Jinni (Jinni2004), a Prolog-like inference machine; and (*iii*) a *Persistence Support Service* (developed in TeleCARE) that allows system recovery in the case of a break-down.

3.1.2 Inter-Platform Mobility

The Aglets provides basic inter-platform mobility mechanisms. Nevertheless, in order to implement a security level for accessing TeleCARE resources, which is a critical issue in the TeleCARE domain, such basic mechanisms need to be extended. For this purpose two additional components are included:

i) The **Agent Reception & Registration** is responsible for accepting or refusing incoming agents. It implements the following main functions:
   - Accept / refuse incoming mobile agents,
   - Register agent, and
   - Notify sender (and origin) of accepted incoming agent.

ii) The **Agent Exit Control** is responsible for the "logistics" of sending out an agent. The main function implemented is:
   - The control of outgoing mobile agent.

Complementing these two main components, the *Inter-Platform Mobility* module also comprises a function to:
- Log information registration regarding agent's migration.

3.1.3 Inter-Agent Communication

The Aglets system provides a simple mechanism for inter-agent communication. However this mechanism is not sufficient for reliable communication between mobile agents. Therefore, this module implements additional communication services, namely:
- Extended message exchange mechanism, and
- The use of FIPA ACL.

3.1.4 Platform Manager

The *Platform Manager* is responsible for the configuration and specification of the operating conditions of the TeleCARE Platform in each site, in order to ensure the platform is working adequately. This module comprises functionalities for (*i*) system configuration, (*ii*) system supervision, (*iii*) definition of users and categories, and (*iv*) GUI for both programmers' interaction and users' interaction.

Additionally, this module has two main sub-modules:
- The *Agent Factory* is the module that can help service developers in the implementation of Vertical Services.
- The *Resource Manager Agents* module provides a common and abstract way of dealing with devices and home appliances in TeleCARE.

3.2 The TeleCARE Platform

The TeleCARE Platform is the environment where the TeleCARE agents (named as Agents in the following) will live. These Agents can be stationary or mobile.

The TeleCARE Platform supports two types of Agents: (*i*) the System Agents that are responsible for the good-functioning and management of the TeleCARE Platform; and (*ii*) the Application Agents, which are all the other Agents defined by the user in order to perform any task or service, or to build the TeleCARE applications. The former are stationary Agents, unique for each host considered as TeleCARE Platform; the latter can be stationary or mobile Agents, depending on the application. Several modules of the TeleCARE Basic Platform (at the Core MAS Platform Level) are defined as System Agents.

For purposes of simplification, in this paper only the following System Agents will be considered: (*i*) the *Agent Registry*, (*ii*) the *Agent Reception Control*, and (*iii*) the *Agent Exit Control*. These Agents form the *Inter-Platform Mobility* module, and their characteristics will be described below. Other System Agents of the platform are part of the modules *Federated Information Management*, *Resources Catalogue Management*, and *Platform Manager*, but they are not fundamental in the process of reliable communication support.

On the left side of Figure 4 the two types of Agents in a TeleCARE Platform are shown. On the right side a stylized representation of a TeleCARE Platform, where the System Agents are represented as blocks into the platform, is depicted (please, notice that the Agent Systems can be considered as a part of the agents' platform and not like agents themselves).

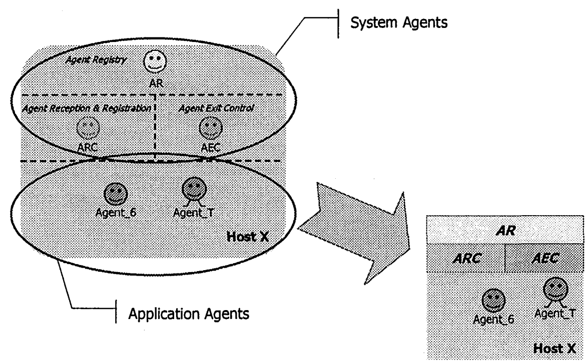


Figure 4 – Types of agents in the TeleCARE Platform

3.3 The TCAgent Class

The *TCAgent* class, illustrated in Figure 5, represents the base class for all System and Application Agents in TeleCARE.

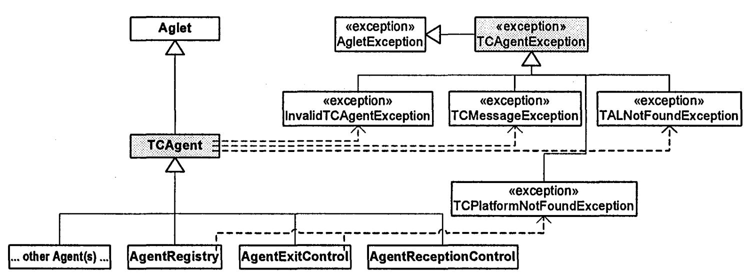


Figure 5 – TeleCARE classes' hierarchy

The *TCAgent* is the key class in the TeleCARE API. It is the abstract class that the developers must use as the base class to create customized Agents. Every class that inherits from it can be instantiated as an Agent. Some classes of exceptions are defined in order to deal with the possibility of unusual or unexpected system behavior, such as, e.g., the non-existence of the remote TeleCARE Platform to where the Agent is going to travel to.

The *TCAgent* defines methods for controlling its own life cycle, which are: (*i*) methods for dispatching, deactivating and disposing the Agent; (*ii*) methods for communication; and (*iii*) methods for implementing persistency support mechanisms.

Some of these features are already provided by the basic Aglets framework. The *TCAgent* uses them within the context of the TeleCARE extended functionality. The extended functionalities of the *TCAgent* are: (*i*) Agent registration and localization, (*ii*) communication through structured content messages, and (*iii*) fail-safe Agent execution (mainly using persistency mechanisms).

Furthermore, some of these functionalities, mainly the first one, are executed using the support offered by System Agents. As mentioned above, System Agents are in charge of managing all stationary and/or mobile Agents inside the platform. Thus, the *TCAgent* internally communicates with System Agents in order to achieve the mentioned features and functionalities. The communication processes are totally transparent to the developers.

3.4 The Passport

The Agent's passport is a mechanism that allows for some levels of security to protect TeleCARE communities, i.e., it is a "gate" for accessing and using the TeleCARE resources. The passport is also used for migration control (in a similar way as described in (Guan et al., 2003), but without using visas) and locating Agents, and it is encapsulated in messages sent by Agents as well. The principal characteristics of the passport are: (*i*) the passport is unique for every Agent, (*ii*) the passport is part of every Agent, and (*iii*) the passport can be partially assigned by the developer, but cannot be modified by him/her.

After the creation of an Agent, its passport constitutes a proof of its identity. It is the official "travel document" recognized by any TeleCARE site of the network. Any mobile agent that intends to migrate to another platform must have a valid passport. The passport structure is shown in Figure 6, and it is composed of the following fields:

- ***TAL*** – The *TeleCARE Agent Locator*, which is an identifier used by the system for locating an Agent. With information provided by TAL, the system can find the proxy of any agent, no matter where it is (for instance, to send it a message), in almost all cases. It contains data of the Aglets' identification of the agent (a string of 16 hexadecimal characters), the host where the agent has born, and the host where the agent is currently living.
- ***TLAID*** – The *TeleCARE Logical Agent Identification*, which is used to validate an agent at any platform, and to locate an agent (using human understandable data) as well. The developers can identify any TeleCARE agent with the information provided by the TLAID, given any parameter of the two substructures the compose it:
  - ***TLAD*** – The *TeleCARE Agent Data* that contains specific human readable identification of the Agent, namely its name and type; and

- ***TLUD*** – The *TeleCARE User Data* that encloses human readable identification of the user who created the agent, namely the role and ID of the user, and the domain node of the TeleCARE Virtual Organization that the origin host (or platform) of the agent belongs to.

- ***agentVal*** – It is used for assigning the duration time of the Agent's passport.
- ***itineraryDone*** – It indicates the itinerary traveled by the Agent, and stores a list of the last visited hosts.

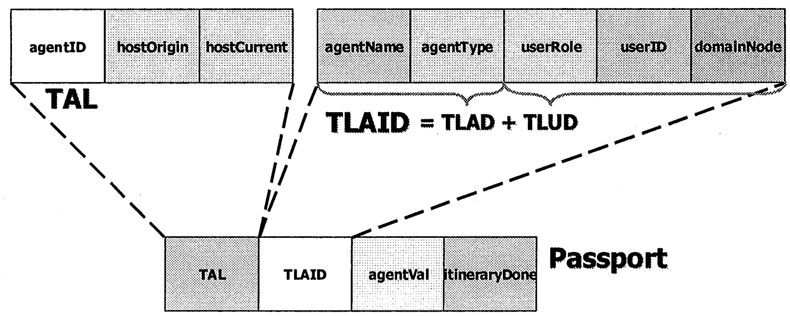


Figure 6 – The TeleCARE passport structure

3.5 The Inter-Platform Mobility Module

The module is composed of a set of stationary Agents at each TeleCARE Platform that provide its main functionalities. These Agents, which are the System Agents described in Section 3.2, are the following:

❒ The **Agent Registry**, that keeps a record / register of every Agent currently living and/or that was created in the platform. This register consists of a copy of the passport of each Agent. For instance, whenever an Agent needs to send a message to another Agent, it first gets the Receiver's TAL from the local (or remote) *Agent Registry*. A timer to refresh the register is included as well.

❒ The **Agent Reception Control**, that is responsible for the reception of the incoming mobile Agents. Depending on their passports, these Agents can be accepted or refused. Whether an arriving Agent is accepted in the local platform or not, the *Agent Exit Control* of the remote platform is notified.

❒ The **Agent Exit Control**, that controls the outgoing of mobile Agents. Every time an Agent is going to leave the platform, its passport and destination (as an available and valid TeleCARE Platform) are first checked.

4. RELIABLE COMMUNICATION IN TELECARE

4.1 Communication Mechanism

As abovementioned, TeleCARE MAS is built on the top of Aglets. Some extensions of Aglets messaging were developed in order to allow that:

- An Agent can communicate with other Agents if it knows some information about them. With the knowledge of, for instance, an Agent's location and some other parameters of its TLAID, an Agent can be easily reached (by another Agent) without further effort, in order to establish a contact (see Figure 7).

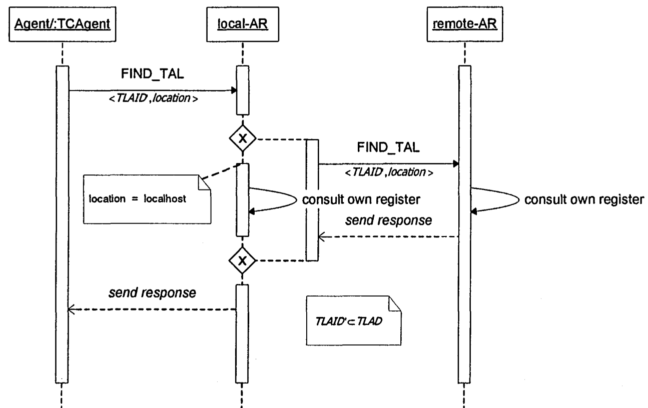


Figure 7 – Finding an Agent's TAL from some parameter of its TLAID

- An Agent that receives a message knows whom the Sender is. A typical situation is when an Agent sends a message to a resource manager Agent (asking it to perform some action on a resource); before granting access to the resource, the Sender's permission to use the resource must be first verified.
- Messages from non-Agents are adequately managed. There might be applications that require message exchanging with non-Agents; the Receiver must have a way of identifying such type of messages.
- Messages are certified / verified. Before being processed by the Receiver, the outgoing message has to be certified by encapsulating the passport of the Sender into the message to be sent. On the Receiver's side, the incoming message may also be verified, obtaining the passport of the Sender from the message. The verification process would succeed if the message carries a known/valid passport, otherwise the message should not be handled (see in Figure 8).

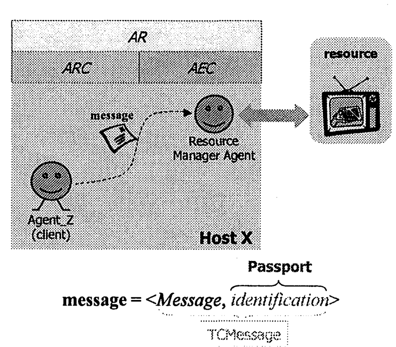


Figure 8 – Accessing to the TeleCARE resources

For dealing with the certification / verification processes a class named *TCMessage* was developed. This class acts like a wrapper of Aglets' *Message*, but enriched with the identification of the Sender (see Figure 8), and adding a method for handling message exceptions between Agents in different (remote) platforms.

4.2 Protocol of the Reliable Communication

In (Vieira, 2001) a mobile agent is defined as a 6-tuple $< id, code, state, ho, hc, t >$, where *id* is the unique identification of the agent, *code* represents the code of the agent, *state* is the vector of its static state, *ho* is the platform/host where the agent was created, *hc* is the host where the agent is currently living, and t is a particular instant in the life of the agent. For TeleCARE we redefined this definition into:

Definition 1. *The representation of an Agent at any instant is a 4-tuple* $< passport, code, state, t >$, *where passport is the unique identity of the Agent, code represents the code of the agent, state is the vector of its static state, and t is a particular instant in the life of the Agent.*

Definition 2. *The passport is a 4-tuple* $< tal, tlaid, agentVal, itineraryDone >$. *The TAL is a 3-tuple* $< id, ho, hc >$. *The TLAID is a 2-tuple* $< tlad, tlud >$. *The TLAD is a 2-tuple* $< agentName, agentType >$. *The TLUD is a 3-tuple* $< userRole, userID, domainNode >$.

All parameters have been defined above (Section 3.4). All passport parameters are constants in the life of an Agent, except the parameters *hc* and *itineraryDone*, which have their value changed in each migration event of the Agent.

An example [hypothetical] scenario of a mobile agent system is depicted in Figure 9. In this case all hosts are TeleCARE platforms. Every Agent that is created at any host can autonomously roam among all platforms, when necessary, in order to carry out its tasks. Communication between Agents is achieved as it is explained afterwards.

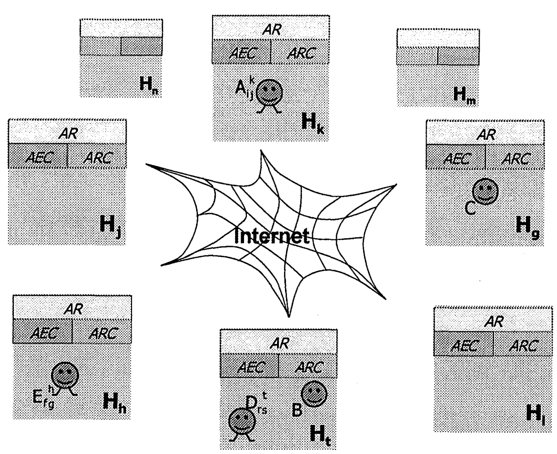


Figure 9 – Schematic scenario of a TeleCARE system

Let A_{ij}^{k} be the i-th Agent that born in the j-th host (H_j) and is currently living in the k-th host (H_k), where j is not necessary different from k. Let B be a stationary Agent that shall send a message to A_{ij}^{k}. For the first contact between both Agents, B must know, at least, either H_j or H_k and, if needed, some parameters of $TLAD'$ of the agent to which it shall communicate, A_{ij}^{k} (where $TLAD' \subset TLAD$). Depending on the searching data, B will receive a null set, a set of several Agents A_{xj}^{k} (and in this case a refinement to obtain A_{ij}^{k} is necessary), or the specific Agent A_{ij}^{k}. Because the Receiver always knows who the Sender is, a reliable communication can be established between both Agents.

Let's consider the following cases:

- If A_{ij}^{k} migrates to another host (H_l), changing now to A_{ij}^{l}, the communication between both Agents will be assured, in the new first contact, if H_j is reachable.
- If for any reason H_j is not reachable (the communication channel fails or the host H_j crashed), the communication between both Agents, B and A_{ij}^{l}, can be reestablished if A_{ij}^{l} sends a message to B.
- If a new Agent C wishes to start a communication with A_{ij}^{l}, H_j is unreachable, and C does not know H_l, the communication cannot be done.
  - A special case occurs when $l = j$ and H_j is unreachable, which has the result that communication between C (or B) and A_{ij}^{l} cannot succeed.
- If A_{ij}^{l} migrates to a new location H_m, changing to A_{ij}^{m}, and H_j is unreachable, H_j will not have any updated reference to where A_{ij}^{m} is currently living.
- If H_j is now reachable and C tries again to establish a communication with A_{ij}^{m}, the communication cannot be done because H_j does not know the current value hc of A_{ij}^{m}, erasing the A_{ij}^{l} record (the last record of the Agent before H_j was unreachable) of it own register; considering that A_{ij}^{x} is probably dead.
  - A special case occurs when A_{ij}^{x} dies. If H_j is reachable, it receives the notification of the death of A_{ij}^{x}. If H_j is unreachable, when it seeks for A_{ij}^{x} at the last known location H_x and does not find it, H_j erases the record of A_{ij}^{x} on its register, and return a null set to all Agents that want to establish a communication with the missing A_{ij}^{x}.
- If H_j is reachable and A_{ij}^{m} moves to location H_n, changing to A_{ij}^{n}, H_j will be notified and it will store the record of A_{ij}^{n} on its own register. Now if C tries to start a communication with A_{ij}^{n}, it will be succeeded.

All these cases can be generalized to a situation where two mobile Agents A_{ij}^{k} and D_{rs}^{t} want to establish a reliable communication, within the restrictions aforesaid, between them. Also, if the Agent E_{fg}^{h} (or B, in the case of considering a stationary agent) needs to send multicast messages to A_{ij}^{k} and D_{rs}^{t}, it can get a reasonable success.

The proposed solution fails when the following three situations appear:

a) The agent's home, H_j, of the Receiver, $A_{x\,j}^{\,x}$, is unreachable, and
b) The Receiver migrates to another host, and
c) The Sender, B, initiates the communication process and does not know what the itinerary of the Receiver is.

5. CONCLUSIONS AND FURTHER WORK

Mobile agents technology can be used in several industrial environments, such as remote operation and monitoring of machines and equipment, tele-robotics, or even in services such as tele-health, tele-assistance and remote elderly care. Some of these domains require maximum reliability of agent-based applications, where communications among agents is one of the key requirements.

In this paper a practical solution for reliable communication in mobile agents systems, as developed in the TeleCARE platform, is described. This solution extends the approach of forwarding pointers in order to achieve reliable communication among mobile agents in almost all cases—some applications developed using the TeleCARE platform, and, in consequence, the proposed solution for reliable communication, can be found in (Camarinha-Matos, 2004).

It is also shown what the involved components on Agent communication of the TeleCARE platform are. This explanation is needed in order to discuss the characteristics and limitations of the solution. Since mobile agents systems have asynchronous characteristics, the solution focuses on guaranteeing, within the discussed limitations, reliable fault-tolerant communication for mobile agents without the restrictions of synchronous schemes. Nevertheless more effort is necessary in order to diminish exceptions as far as possible.

6. ACKNOWLEDGEMENTS

This work was funded in part by the IST program of the European Commission. The authors thank the contribution of the TeleCARE consortium members, and specially thank the participation of João Sarraipa, Ana Inês Oliveira, Filipa Ferrada and João Rosas, in the accomplishment of the reported results.

7. REFERENCES

1. Assis-Silva FM and Macêdo RJA. Reliable communication for mobile agents with mobile groups. IEEE/ACM ICSE 2001 Proc. Workshop on Soft. Engineering and Mobility; Toronto, Canada, 2001.
2. Baumann J, Hohl F, Radouniklis N, Rothermel K, Straßer M. Communication concepts for mobile agent systems. In Rothermel K and Popescu-Zeletin R (eds.) Mobile Agents: 1st Int. Workshop MA'97; LCNS 1219, Springer: Berlin, pp. 123–135, 1997.
3. Baumann J and Rothermel K. The shadow approach: An orphan detection protocol for mobile agents. In Rothermel K and Hohl F (eds.) Mobile Agents: 2nd Int. Workshop MA'98; LCNS 1477, Springer: Berlin, pp. 2–13, 1998.

4. Camarinha-Matos, Luis (ed.) Tele-Care and Collaborative Virtual Communities in Elderly Care. Proc. 1st Int. Workshop on Tele-Care and Collaborative Virtual Communities in Elderly Care (TELECARE 2004), in conjunction with ICEIS 2004, INSTICC Press, ISBN: 972-8865-10-4; Porto, Portugal, 13 April 2004.
5. Camarinha-Matos LM and Afsarmanesh H. Design of a virtual community infrastructure for elderly care. PRO-VE 2002, 3rd IFIP Working Conf. on Infrastructures for Virtual Enterprises; Sesimbra, Portugal, May 1–3, pp. 439–450, 2002.
6. Camarinha-Matos LM, Viera W, Castolo O. A mobile agents approach to virtual laboratories and remote supervision. Journal of Intelligent and Robotic Systems, KAP, pp. 1–22, 2002.
7. Camarinha-Matos LM, Castolo O, Rosas J. A multi-agent based platform for virtual communities in elderly care. ETFA-2003, 9th IEEE Int. Conf. on Emerging Technologies and Factory Automation; Lisbon, Portugal, 16–19 September 2003.
8. Camarinha-Matos LM, Rosas J, Oliveira AI. A mobile agents platform for telecare and teleassistance. In (Camarinha-Matos, 2004), pp. 37–48, 13 April 2004.
9. Fuggeta A, Picco GP, Vigna G. Understanding code mobility. IEEE Trans. on Soft. Engineering, 24(5):342–361, 1998.
10. Glass G. ObjectSpace Voyager Core Package Technical Overview. ObjectSpace, Inc., White Paper, 1999.
11. Gray R, Kotz D, Cybenko G, Rus D. Handbook of Agent Technology; chapter Mobile Agents: Motivations and State-of-the-Art Systems. AAAI/MIT Press, 2000.
12. Gray RS, Cybenko G, Kotz D, Peterson RA, Rus D. (2002). D'Agents: applications and performance of a mobile-agent system. Software: Practice and Experience, 32(6):543–573, 2002.
13. Guan S-U, Wang T, Ong S-H. Migration control for mobile agents based on passport and visa. Future Generation Computer Systems, no. 19, Elsevier Science B. V., pp. 173–186, 2003.
14. Hendler, James. Is there an intelligent agent in your future? Nature, Web Matters, 11 March 1999. From http://www.nature.com/nature/webmatters/agents/agents.html
15. Jinni2004: Java and Prolog Software for Internet Programming. From http://www.binnetcorp.com/
16. Kotz D, Gray R, Rus D. Future Directions for Mobile Agent Research. Technical Report TR2002-415, Department of Computer Science, Dartmouth College, 2002.
17. Lange D and Oshima M. Programming and Deploying Mobile Agents with Aglets. Addison-Wesley Longman, Inc., Reading, MA, 1998.
18. Liu C-F and Chen C-N. A sliding-agent-group communication model for constructing a robust roaming environment over Internet. Mobile Networks and Applications, no. 8, KAP, pp. 61–74, 2003.
19. Murphy A and Picco GP. Reliable communication for highly mobile agents. Autonomous Agents and Multi-Agent Systems, KAP, pp. 81–100, 2002.
20. Picco GP, Murphy AL, Roman G-C. LIME: Linda meets mobility. In Garlan D (ed.) Proc. 21st Int. Conf. on Soft. Engineering; pp. 368–377, 1999.
21. Pozo S, Gasca R, Gómez MT. Securing mobile agent based tele-assistance systems. In (Camarinha-Matos, 2004), pp. 63–72, 13 April 2004.
22. Protégé-2000: The Protégé Ontology Editor and Knowledge Acquisition System. From http://protege.stanford.edu/
23. Ranganathan M, Bednarek M, Montgomery D. A reliable message delivery protocol for mobile agents. Proc. Joint Symposium ASA/MA2000; LCNS 1882, pp. 206–220, 2000.
24. Roth V and Peters J. A scalable and secure global tracking service for mobile agents. 5th Int. Conf. MA 2001; LCNS 2240, Springer: Heidelberg, pp. 169–181, 2001
25. Vieira, Walter. Adaptive Mobile Agents for Remote Operation. PhD thesis, New University of Lisbon, 2001 (in Portuguese).
26. Vieira W, Camarinha-Matos LM, Castolo LO (2001). Fitting autonomy and mobile agents. ETFA-2001, 8th IEEE Int. Conf. on Emerging Technologies and Factory Automation; Antibes – Juan les Pins, France, vol. 2, pp. 471–480, 15–18 October 2001.