

# A Mobile Agents Platform for Telecare and Teleassistance

L. M. Camarinha-Matos, João Rosas, Ana-Inês Oliveira

New University of Lisbon / Uninova Monte Caparica, 2829-516 Caparica, Portugal

**Abstract.** In order to facilitate the development and integration of new services for telecare there is a need for the design and development of an open and flexible infrastructure into which services and supporting resources can be easily plugged in. A mobile agents based approach for such infrastructure was developed in the TeleCARE project and its architecture and implemented prototype are discussed.

## 1 Introduction

Teleassistance and provision of remote care to elderly living alone at home represents a very demanding case of distributed system. Developments in this area have to cope with some important requirements, namely:

- 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 telecare community.

Furthermore, this area represents a very sensitive domain in the sense that it deals with critical data about individuals, raising issues of security, and the need to guarantee a high continuity of service. These are in fact some of the pre-requisites to guarantee social acceptance of any new approach to the introduction of technology in elderly care.

In spite of recent developments in computer networks and pervasive computing, effective implementation of telecare systems still requires the development of open and flexible infrastructures offering some form of plug-and-play functionality. With different organizations developing different products and services, in a variety of different areas, there is a need for a common platform into which all these development may be plugged so that interoperability is possible.

In this context, one of the main aims of the IST TeleCARE project was the design and development of a generic platform that allows progressive addition, at a minimal cost, of new care services, home appliances, and monitoring / assistance devices. A mobile agents approach is adopted as the base development paradigm. Although

different approaches could have been considered (e.g. transaction-based or web service-based), the mobile agents approach offers some potential benefits in face of the requirements of this domain. In particular:

- Moving the code to the places where actions are required enables real-time response and autonomy and continuity of service provision, with reduced dependency on network availability and delays;
- Since new mobile agents can be built and sent for remote execution whenever needed, higher levels of flexibility and scalability are achieved.

**The TeleCARE Extended MAS Platform.** The TeleCARE platform is developed on top of AGLETS [1]. AGLETS, as a mobile agents framework, provides basic inter-platform mobility and communication mechanisms. Nevertheless, in order to implement security, a critical issue in the elderly care domain, such basic mechanisms need to be extended. This paper is focused on the multi-agent related components and extensions of the TeleCARE platform, the gray areas in Fig. 1.

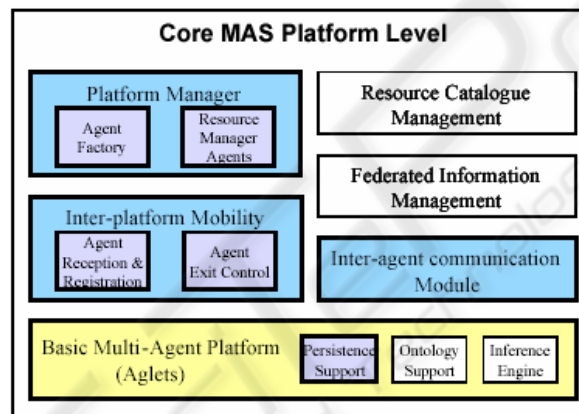


Fig. 1. Modules of the TeleCARE Core MAS Platform

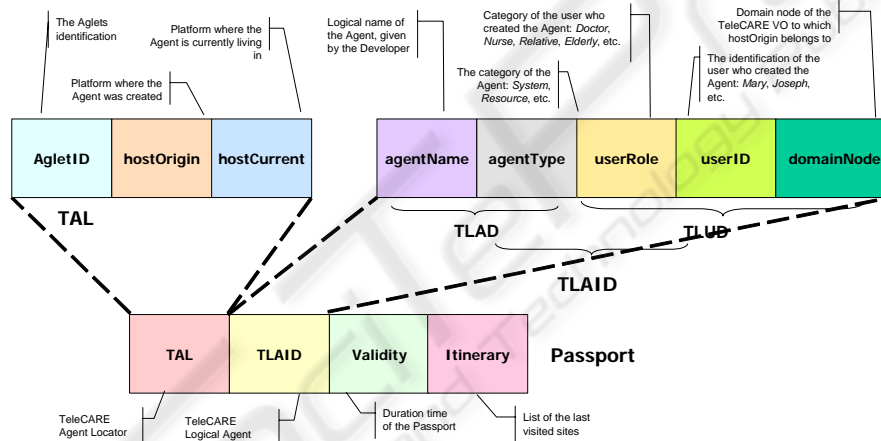
The Inter-Platform Mobility module controls the processes involved in agent migration, such as permissions to leave from/enter into a node and keeping a registry of the agents at each node. The Inter-agent Communication Module provides extended message communication mechanisms, such as the inclusion of sender's identification and the support for FIPA ACL [5]. Finally, the Platform Manager is responsible for the configuration and specification of the operation conditions of each TeleCARE node.

Before discussing in details these components, it is necessary to introduce two fundamental entities: the *TeleCARE abstract agent* and its *Passport* class.

The TeleCARE agent abstract class (*TCAgent*) is the base class for all other agents created in the *TeleCARE* system. This class defines methods for controlling the agent's life cycle, such as methods for dispatching, deactivating and disposing an agent; methods for communication, and methods for persistency support mechanisms. Although some of these features are already provided by the basic AGLETS SDK

platform, the *TCAgent* wraps them and extends the basic functionalities by inclusion of: (i) agent registering and localization, (ii) communication through highly structured content messages, and (iii) permanent and fail-safe agent execution.

Each TeleCARE agent has a passport that constitutes its “proof of identity”. It is the official “travel document” recognized by any TeleCARE site of the community. Therefore, any mobile agent that intends to migrate to another platform must have a valid passport. The passport structure (see **Fig. 2.**) is composed of several substructures: *TAL* (*TeleCARE Agent Locator*), *TLAID* (*TeleCARE Logical Agent Identification*), *Validity* and *Itinerary*. The *TAL* is a system level identifier that is used by the system to locate an agent. With the information provided by *TAL*, the system can find the proxy of any agent no matter where it is. The *TLAID* is a logical identifier that is used to locate an agent using human understandable data and it comprises two sub-structures, which are the *TLAD* (*TeleCARE Agent Data*) that contains specific human readable data on the agent and the *TLUD* (*TeleCARE User Data*) that contains human readable data on the user who created the agent. The field *Validity* is used to assign the time duration of the agent’s passport. Finally, the *Itinerary* represents a list of the last visited hosts (in case of mobile agents).

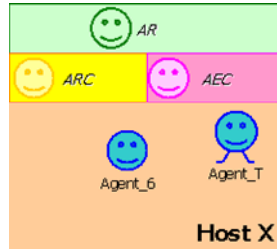


**Fig. 2.** The TeleCARE agent's Passport structure

## 2 Inter-Platform Mobility

Since the TeleCARE system is intended to operate in a geographically distributed system and the application domain is particularly demanding in terms of reliability it is important to implement security means to prevent harm from malicious agents. For this purpose it was necessary to extend the basic mobility mechanisms provided by AGLETS. These extensions are implemented through the inter-platform mobility support agents, a set of stationary agents that provide the main functionalities of the Inter-Platform Mobility (IPM) module: AgentRegistry (AR), AgentReceptionControl

(ARC) and AgentExitControl (AEC). From the IPM perspective, each TeleCARE node is seen as illustrated in Fig. 3. The ARC and AEC participate in the processes of controlling the migration process in terms of the accessibility of the nodes based on the agents' passports.



**Fig. 3.** The IPM view of a TeleCARE node

**Agent Registry.** The class *AgentRegistry* stores a record for each agent that lives at the local host and/or was created at local host and which is currently running in a remote platform. The registration consists of a copy of the passport of each agent.

*AgentRegistry* performs the registration of agents by sending and receiving messages to/from other *System Agents* (*AgentExitControl* and *AgentReceptionControl*). For instance, when an agent is created its passport must be registered in order that it can live in the TeleCARE system, by sending a message of type “REGISTRATION” to *AgentReceptionControl*. To verify and accept as valid the passport of the new agent, the *AgentReceptionControl* will send a message to *AgentRegistry* with type “REGISTER” in order to register the agent in the TeleCARE platform.

When an agent travels to another location it must require its registration at the destination platform. After that, its register at the departing platform must be updated or removed, depending whether the local platform is its origin host or not. These actions are taken care of by the *AgentReceptionControl* and *AgentExitControl* and the result is communicated to *AgentRegistry* with the purpose of performing the needed changes in the agents registry.

**Agent Reception Control.** The agent *AgentReceptionControl* functions as a gatekeeper of incoming agents that wish to enter in a TeleCARE node. The application agents will interact with this agent through messages of type “REQUEST\_VALIDATION” and “REGISTRATION”. In both cases, if the verification process is successful, the *AgentReceptionControl* will send a message of type “REGISTER” to *AgentRegistry*, so that the agent can be registered at the local platform.

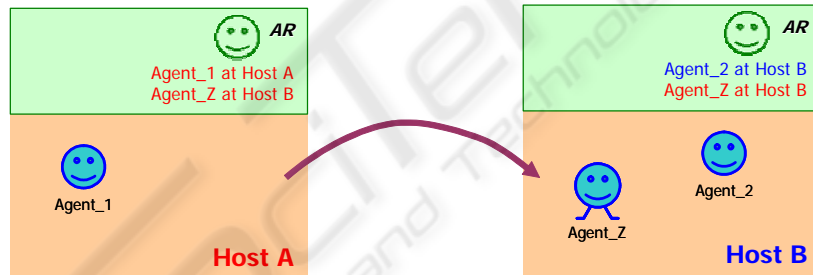
Whether an arriving agent is accepted or not in the local platform, the *AgentExitControl* of the remote (departing) platform is notified.

**Agent Exit Control.** The *AgentExitControl* is responsible to implement the functionalities of the exit control module. Thus, it includes methods to control if an agent has permission to travel, and to check if the destination is an available and valid TeleCARE platform. Like the other two classes above, this class does not have public methods so the only way to access its functionalities is through messages.

When an agent is about to travel, it has to require permission for traveling from the *AgentExitControl* of its current node by sending a message of type “REQUEST\_TRAVELLING\_PERMISSION”, and also by sending its passport and specifying the destination platform. The *AgentExitControl* verifies the passport of the agent in order to know if it has traveling permission, and also verifies if the destination is an available and valid TeleCARE platform. If the response is positive for both conditions, the agent can travel to the intended destination.

Immediately after an agent arrives to a destination, it requires from the local *AgentReceptionControl* to be registered in the platform, by sending a message of the type “REQUEST\_VALIDATION” together with its passport. The passport is verified by the *AgentReceptionControl*, which sends a response to the incoming agent. If the passport is valid, the *AgentReceptionControl* sends a message of type “REGISTER\_UPDATE” to the *AgentExitControl* instances running at the local host, last host, and origin host of the mobile agent. This will lead to a registration of the agent at the local platform, a removal of its register from the last platform registry, and an update of its register at the origin platform, as illustrated in **Fig. 4**.

If the verification process of the agent’s passport results fails, the agent is disposed and its respective registers, both at last host and origin host, are erased.



**Fig. 4.** After Agent\_Z has traveled to Host B, the AR of the original platform is updated. This way any agent can be located, provided that at least its original platform is known.

Through the operation of these three system agents it is possible to achieve some degree of privacy and security since it is possible to configure which kind of agents are allowed to perform their tasks in each platform.

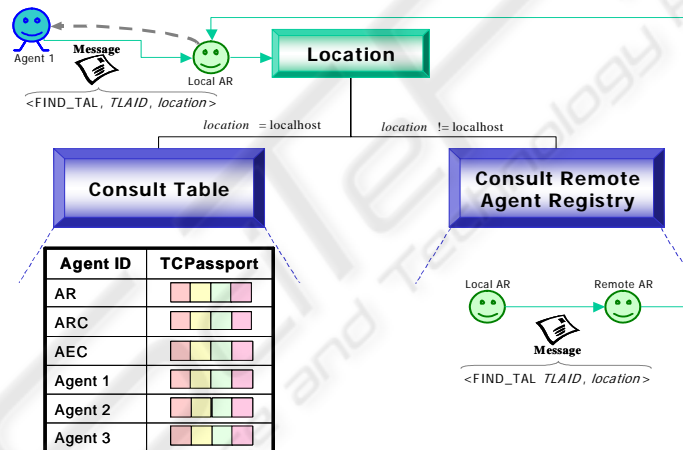
### 3 Inter-Agent Communication

The objective of extending the AGLETS basic message exchange mechanism is to have additional functionality needed for the adequate operation of the TeleCARE Platform in terms of reliable communication. Whenever an agent wants to send a

message to another agent, it first has to locate the receiver. The localization process is driven by the values of the agent's *TLAID*. Whenever an agent receives a message, the message must include the sender's identification and other details so that a validation of the message can be made by the receiver. The implemented approach is to encapsulate the sender's passport in the message structure. This result in a new structure called *TCMessage*.

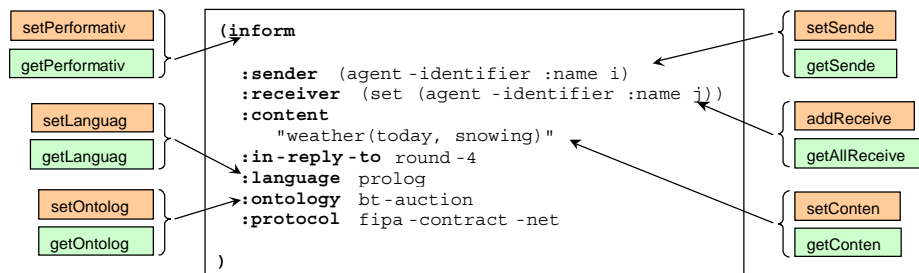
**Communicating with an Agent.** Since AGLETS is the base of the *TeleCARE* platform, the basic communication mechanism between agents follows the directives of agent communication in AGLETS, mainly in terms of the proxies of the agents. Although the proxy mechanism is appropriate to deal with distributed environments, it is not sufficient in some aspects: it is not human readable and it is unsafe as by using the AGLETS's broadcast the agent's proxies are available to any agent, even malicious ones.

Therefore, in *TeleCARE* the communication process resorts to the agent's passport information when a message needs to be sent to an agent. The passport's *TLAID* and *TAL* structures are used to locate an agent or set of agents. By specifying the values (not all required) for the *TLAID* attributes as a search pattern, the *TAL* of the matched agent(s) is (are) obtained and the communication can proceed (see **Fig. 5**).



**Fig. 5.** Process of locating an agent based on the values of the passport's TLAID; after the desired agent is found, the message is sent.

**FIPA ACL Messages.** One of the features integrated in the *TeleCARE* MAS platform as an extension to AGLETS is the possibility to communicate using *FIPA ACL* Messages, which allows transmission of messages with high structured content. An example of an *ACL* message can be seen in Fig. 6.



**Fig. 6.** Example of an ACL message. For each element of the message, the correspondent ACL methods for the definition/extraction of the content are indicated.

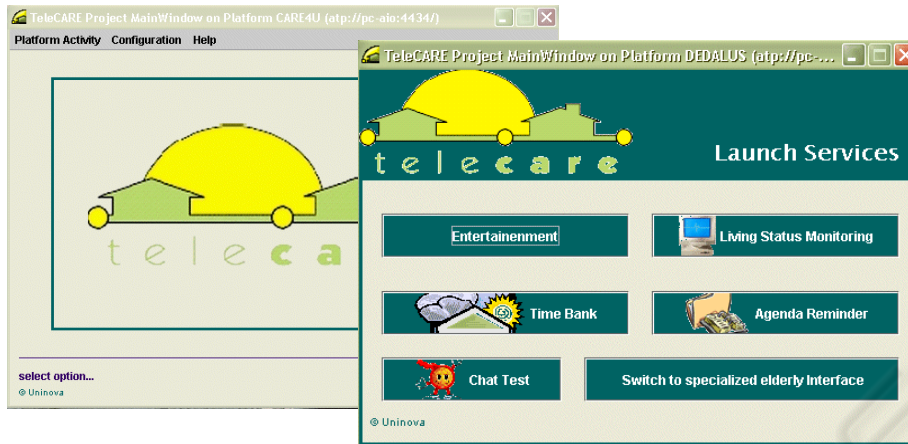
## 4 Platform Management

The platform management functionalities are used to ensure that the platform is working adequately. This sub-system of the TeleCARE platform consists of three basic modules, which are: Platform Manager, Agent Factory, and Resource Manager. The Agent Factory (AF) is the module that helps service developers in the implementation of Vertical Services (VS). It also assists in the instantiation process of the TeleCARE agents. Finally, the module Resource Manager (RM) provides a common and abstract way of dealing with devices and appliances in TeleCARE.

**The Platform Manager.** The Platform Manager (PM) is the component of the TeleCARE platform architecture responsible for the configuration and specification of the operating conditions of the platform in each node, and it specifically includes *System configuration*, *Login*, and *System supervision* functionalities.

When starting the TeleCARE system there are some attributes that distinguish each site, which are the site name, type and URL. The initial user interface is shown in **Fig. 7.**, which illustrates the entry point for all platform management functions.

Depending on the site type, the users may have different configuration authorizations. At the user's site, it is only allowed to launch the subscribed services. At the Care Center sites it is possible (for users with the right authorization level) to configure the various operating parameters for a given community, such as the users (**Fig. 8.**), elderly, disabilities, relationship types between persons, user roles, and nodes information.



**Fig. 7.** Entry point of the Care center (left window) and user site (right window).

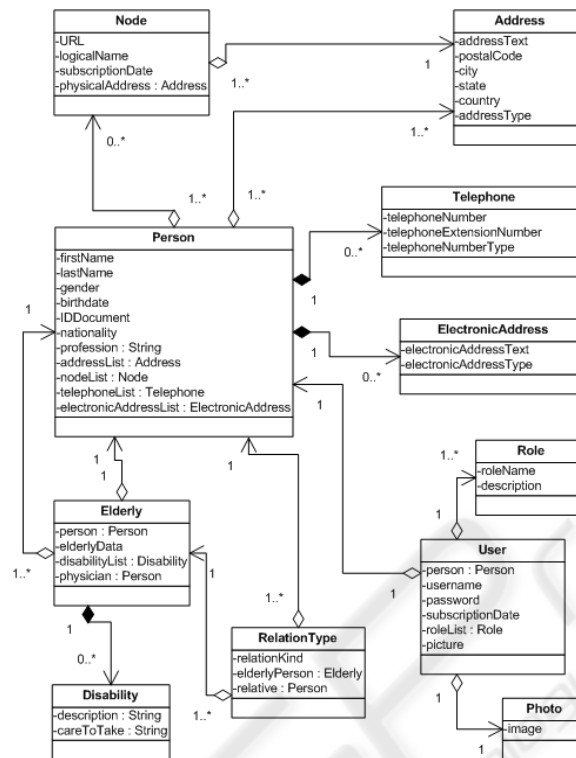
All of this information is structured according to a common ontology (Fig. 9.) followed in all components of the TeleCARE system development.



**Fig. 8.** Users configuration

The access rights to services are granted according to the user's role category, that is, the users inherit the access rights from the roles they perform. The roles specify authorization to allow / deny access to the platform, to types of resources or specific resources, to types of services or specific services, etc.





**Fig. 9.** Common Ontology in the TeleCARE system

As TeleCARE deals with critical information related to the various elderly members, it is necessary to authenticate the users during the login, namely in the case of the care center. For this purpose a biometric identification system (based on finger print recognition) is integrated in the platform (**Fig. 10.**).

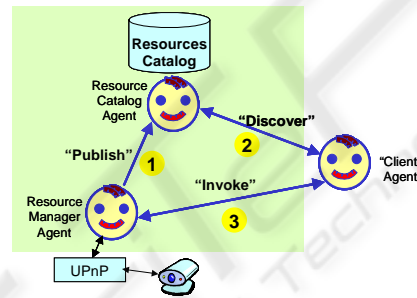


**Fig. 10.** Biometric identification based on finger print recognition

Whenever a user launches an agent, the TeleCARE system fills in the agent's passport with the information of its user (namely the user ID, user role, etc.). Thus, an agent that has been created through the Telecare platform can never forge a passport with false information.

**The Resource Manager.** *Resource Managers* are the agents responsible to provide access to the hardware resources and appliances when requested by upper level Vertical Services running in the *TeleCARE* platform. For each resource (e.g. sensors and home appliances) in each node, a specific agent – *Resource Manager* – has to be developed. This agent is responsible for granting access to the resource functionalities and for checking the access rights whenever another agent requests access to the resource. Although the details of each Resource Manager depend on each particular device, most of the structure of all Resource Managers is common. Therefore, and in order to facilitate the development of new resource managers, a generic template or skeleton – Resource Manager Agents (REMA) – is provided.

The approach used for the Resource Managers and resources integration is based on the UPnP specification [6]. The descriptions of the resources are published in a Resource Catalog Management agent (RCAM), allowing the vertical services (clients) to discover them (**Fig. 11.**).



**Fig. 11.** Resources publishing/discover and invocation

The vertical services interact with the resources via the corresponding Resource Managers, using *ACL* messages.

**The Agent Factory.** The Agent Factory (AF) is the module that assists service developers in the implementation of their services, and also in the instantiation process of the TeleCARE agents. The main functionalities of this module are a minimal set of agent classes with pre-defined mission types, the generation of advanced services by experts, the agent's passport assignment, and a library of agent skeletons and protocols to build agents in the TeleCARE system.

## 5 Agents Persistency Support

Reliability is a major requirement for the TeleCARE platform and persistency of agents is an important element of it. Persistency is a mechanism that allows storing critical information about the running activities of the agents. Whenever a system crashes, they can be resumed when the system is restarted.

The TeleCARE platform provides semi-automatic support for persistency based on the AGLETS built-in snapshot mechanism. Snapshot images of agents can be stored at critical states of the agent's life cycle, such as: at the creation of the Agent, just after the Agent arrives to a new location, and when the Agent is activated. The developer can also use this mechanism at any point of the agent's code assumed to be critical. If there is a system failure, the last snapshot of the Agent is restored and its execution can be resumed with the information stored in that snapshot. The TeleCARE's persistency support also verifies if the agent is compliant with the persistency mechanism.

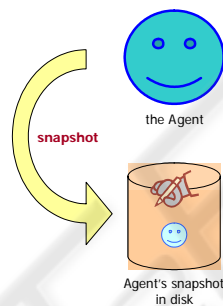


Fig. 12. Basis for persistency in TeleCARE

## 6 Conclusions and Further Work

The mobile agents based platform developed in TeleCARE offers a flexible set of functionalities that proved effective in supporting the development of several vertical services for elderly care. The acquired experience shows however that a proper use of the wide range of functionalities by service developers requires some guidance. For instance, deciding whether to use a mobile agent for a specific mission or a simple exchange of messages might not be a simple decision for programmers less familiar with the underlying machinery of the multi-agent platform. The development of some form of assistance, namely in terms of the Agent Factory, is one direction for further work.

A preliminary validation of the platform functionalities was performed via the implementation of various vertical services that make an extensive use of the offered mechanisms. A minimal field assessment was also made by interaction with some potential software developers but a more extensive field assessment of the functionalities and concepts included in the platform is still necessary.

**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.

## References

1. Aglets API Documentation, v. 2.0.2; Java Aglet Community. From <http://aglets.sourceforge.net/>.
2. Camarinha-Matos, L. M.: TeleCARE Preliminary architecture Design. Deliverable 2.1, IST project 27607, A Multi-Agent Tele-Supervision System for Elderly Care (2003)
3. Camarinha-Matos, L.M.; Afsarmanesh, H.: A Multi-Agent based Infrastructure to Support Virtual Communities in Elderly Care. To appear in International Journal of Networking and Virtual Organisations (2004)
4. Camarinha-Matos, L. M.; Castolo, O.; Rosas, J.: A Multi-Agent based Platform for Virtual Communities in Elderly Care. In Proceedings of ETFA'03 – 9<sup>th</sup> Int. Conf. on Emerging Technologies and Factory Automation, ISBN 0-7803-7937-3, pp 421-428, Lisboa, Portugal, 16-19 Sept (2003)
5. FIPA ACL Message Structure Specification. Foundation for Intelligent Physical Agents. From <http://www.fipa.org/specs/fipa00061/XC00061E.html> (2002)
6. Universal Plug and Play Device Architecture. Universal Plug and Play Forum. From [http://www.upnp.org/download/UPnPDA10\\_20000613.htm](http://www.upnp.org/download/UPnPDA10_20000613.htm) (2000)

