

A multi-agent based platform for virtual communities in elderly care

Luis M. Camarinha-Matos, Octavio Castolo, João Rosas
New University of Lisbon
Faculty of Sciences and Technology,
Quinta da Torre, 2829-516 Monte Caparica,
Portugal
{cam,lgc,jrosas}@uninova.pt

Abstract – Due to the growing numbers of elderly population there is an urgent need to develop new approaches to care provision. The convergence of a number of technologies such as multi-agent systems, federated information management, safe communications, hypermedia interfaces, rich sensorial environments, increased 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 paper, a platform based on mobile agents combined with federated information management mechanisms is introduced as a flexible infrastructure on top of which specialized care services can be built.

1. INTRODUCTION

An integrated elderly care system consists of a number of organizations such as care centers / day centers, health care institutions, social security institutions, which involve the cooperation of a number of different human actors e.g. social care assistants, health care professionals, the elderly people, and their relatives. If supported by computer networks and adequate supporting tools, the collaboration among the care institutions may evolve towards operating as a long-term virtual organization and the various involved actors become part of a virtual community (VC).

Various research projects have been trying to develop technological solutions to improve the care services and reduce their costs. For instance, various experiments have installed “social alarm” systems, namely in the case of people living in remote places. Such systems comprise a portable alarm trigger and an alarm telephone that dials a social alarm control center when there is an emergency situation. More recent works are focused on mobile social alarm systems and monitoring systems based on a diversity of sensors and other devices. Advances in computer networks and ubiquitous computing suggest the opportunity for more advanced care approaches including comprehensive status monitoring, other forms of assistance such as agenda reminders, but also the creation of the opportunity for the elderly becoming involved in a community, and thus reducing their feeling of loneliness.

In this context, the IST TeleCARE project [2] aims at designing and developing a configurable framework focused on virtual communities for elderly support. The proposed solution has to be seen as complementary to other initiatives for the integration of elderly in the society and to reduce their isolation.

The idea of using new technologies to support elderly care provision is not new and there are many initiatives already underway, in a variety of areas, to address the challenges ahead. However, these initiatives are being carried out independently of one another. 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 in such way that interoperability is possible. Project TeleCARE is developing this platform (a common infrastructure) so that the elderly community and those responsible for providing care and assistance can get the best out of the technologies developed (Fig. 1).

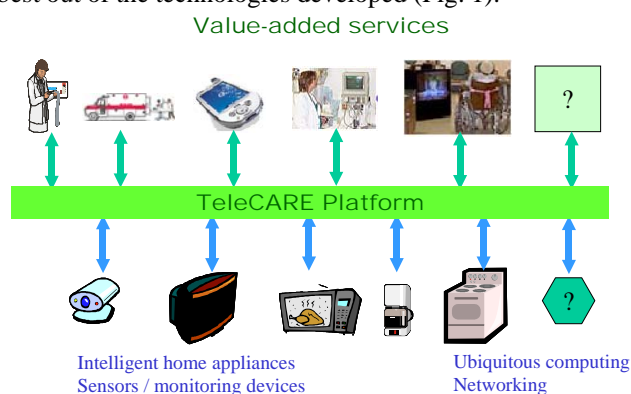


Fig. 1 – Plug-and-play approach

2. TeleCARE ARCHITECTURE

Fig. 2 shows a block diagram of the proposed architecture for a layer to be installed in each node of the TeleCARE organization [1, 2].

The three-level platform comprises:

- **External Enabler Level:** Supports the communication and interfacing to the external devices.

- a. *Safe communications infrastructure*, that provides safe communications – supporting both agent mobility and inter-agent message passing. A virtual private network (VPN) approach is being adopted; in critical cases where communications reliability is mandatory redundant channels, in addition to Internet, may be supported.
- b. *Device abstraction layer*, interfaces to the sensors, monitoring devices and other hardware (home appliances, environment controllers, etc.). These interfaces represent the bridge to any “intelligent home” or “local domotics network” hiding aspects such as low-level protocols, wire-based or wireless communications, etc. UPnP is one of the approaches being evaluated.

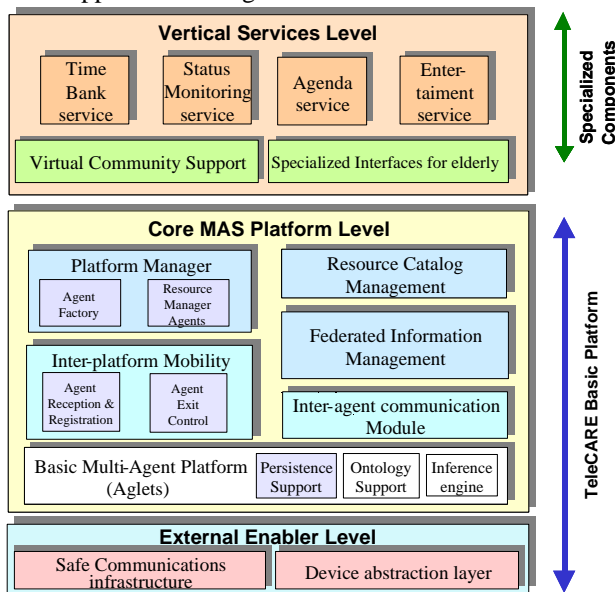


Fig. 2 – The TeleCARE platform architecture

- **Core MAS Platform Level:** It is the main component of the basic platform. It supports the creation, launching, reception (authentication and rights verification), and execution of stationary and mobile agents as well as their interactions. As intelligent agents are envisaged, an inference engine (e.g. JESS or Prolog interpreter) is included. Main modules in this layer:
 - a) *Basic multi-agent platform (AGLETS)*.
 - b) *Inference engine* (e.g. JESS or Prolog interpreter).
 - c) *Ontology support* (e.g. Protégé).
 - d) *Persistence support* – Extension to the MAS platform, providing some basic recovery mechanisms in case one node goes down.
 - e) *Inter-platform mobility* – Extension to the basic MAS platform to support generalized mobility of agents, including security mechanisms. This module includes the Agent Reception & Registration component (for incoming mobile agents) and the Agent Exit Control component (for outgoing mobile agents).
 - f) *Inter-agent communication* – Extension to the basic MAS platform to support communication between / coordination of agents independently of their current location, via FIPA ACL messages.

- g) *Agent factory* – module that supports the creation / specification and launching of new agents.
- h) *Resource manager agents* – To provide a common and abstract way of dealing with devices and appliances in TeleCARE.
- i) *Platform manager* – To configure and specify the operating conditions of the platform in each site, recover from errors, monitor operation, etc.
- j) *Federated information manager* – To support the needed local information management while preserving the information privacy and careful control of access rights to local data for external users. This module, installed in each site, is the local component of the Federated Information Management Architecture (FIMA).
- k) *Resource catalog management* – To manage the catalogue of resources including the specification (and access proxies) of all services available at the current site.

- **Services Level:** Consists of two sets of specialized services:

- a) **Base services** – A set of specialized base (horizontal) services that provide specific functionality for other (vertical) services.
 - *Specialized interfaces for elderly* – In the case of home sites, specialized interfaces are required for elderly that are not familiar with the use of computers. The ultimate goal is to build a user-friendly and “invisible” infrastructure to be used by elderly people.
 - *Virtual Community Support* – To support the creation and operation of future services of virtual communities designed for the elderly, specific virtual community management functionalities are required for the implementation environment in TeleCARE.
- b) **Vertical Services** – A set of specialized vertical services can be implemented on top of the horizontal TeleCARE infrastructure defined in the previous levels to support different interactions with the system. Examples of services being developed include:
 - *Living status monitoring*. This service represents an advance regarding the more traditional “social alarm” systems, as it allows not only bi-lateral interactions and some semi-automatic supervision functionalities, but also the collection of additional information when help is needed or requested. The tranquility and assistance given 24 hours a day improve in a significant manner the elderly’s quality of life and relatives’ peace of mind.
 - *Agenda reminder*. The daily activities related to the welfare of the elderly can be easily scheduled in order to improve their quality of life and well being. Agents belonging to the TeleCARE system are able to remind the elderly a number of activities, ranging from medication to exercise guidance or appointments made with the care center.

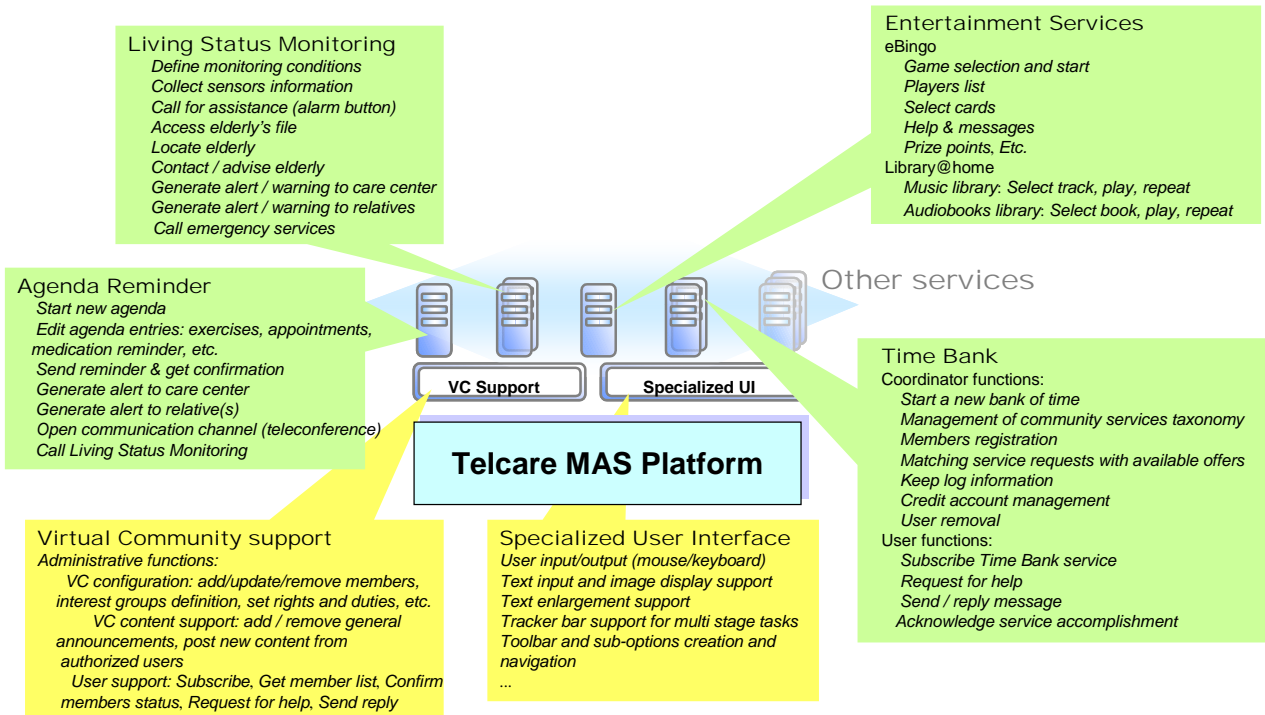


Fig. 3 – Initial TeleCARE services

- ❑ *Time bank.* This service provides a mechanism for collaborative community building / reinforcement, i.e. a way for people to come together and help each other. At the same time it represents one of the mechanisms to support the “active aging” concept.
- ❑ *Entertainment.* The Entertainment Services are designed to ease the sense of isolation elderly feel and provide light entertaining applications to improve their sense of well being, contributing to the maintenance of a social life and also an active aging. A combination of games, music and education programs are offered.

responsibility for the good functioning of the TeleCARE platform, i.e., they act as system agents. Some exceptions are defined in order to deal with the possibility of unusual or unexpected system behavior, such as, the non-existence of the remote TeleCARE platform to where the TeleCARE agent is going to travel.

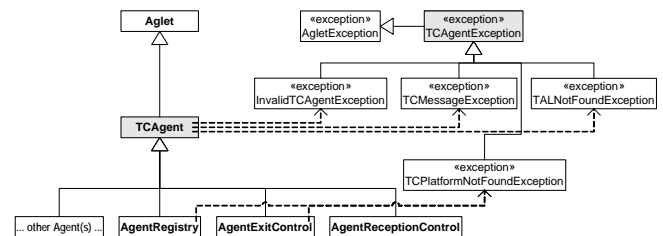


Fig. 4 – TeleCARE agent and related classes

3. TeleCARE MULTI-AGENT PLATFORM

The TeleCARE platform is being developed on top of AGLETS [11]. 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 TeleCARE domain, such basic mechanisms need to be extended. For this purpose a specialization of the AGLETS Aglet class is created – TCAgent – which provides methods for dispatching, deactivating and disposing the agent, communication and persistency support (Fig. 4). The class TCAgent represents the base class for the user defined TeleCARE agents. In the same figure can be seen that the system inter-platform mobility agents, i.e., AgentRegistry, AgentExitControl and AgentReceptionControl, inherit properties from TCAgent. These agents are in charge of managing all the stationary and/or mobile agents inside the platform, and also have the

A. Multi-agent issues

Persistency. Reliability is a major requirement for the TeleCARE platform and the persistency of agents is an important element of it:

- service continuity must be assured for any important task the agents are performing;
- if there is a breakdown of any part of the distributed system, it shall not affect the overall system behavior;
- it is necessary to preserve critical status information in case of a temporary node failure; and
- the recovery process, after a breakdown, shall be as smooth as possible.

Persistency is a mechanism that allows storing information about the running activities of the agents and,

whenever a system crash occurs, to allow them to be resumed when the system is restarted.

AGLETS provides a method called `snapshot`, which saves a snapshot of an agent into a secondary / non-volatile storage. For persistency purposes, every TeleCARE agent can invoke the `TCAgent`'s method `tcSave`, which does a call to `snapshot`, that stores information about the execution status when necessary. 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. In the current version, `TCAgent` provides automatic support for the persistency on three events: (i) at the creation of the agent, (ii) just after the agent arrives to a new location, and (iii) when the agent is activated. It is up to the developer to decide where he/she wants to make additional snapshots of his/her agent, calling the `tcSave` method.

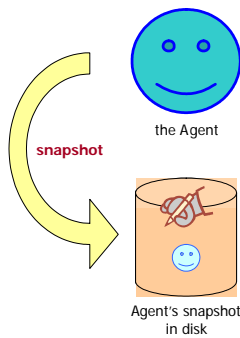


Fig. 5 – Persistency in TeleCARE

Security and passport. Both in case of mobile agents, when an agent arrives at one node, or remote (inter-node) agent communication, it is important to know who the agent is and who it represents. For this purpose, the concept of passport is introduced. After the agent has been created, its passport constitutes a proof of its identity. It is the official “travel document” recognized by any TeleCARE site of the community. Any mobile agent that intends to migrate to another platform must have a valid passport.

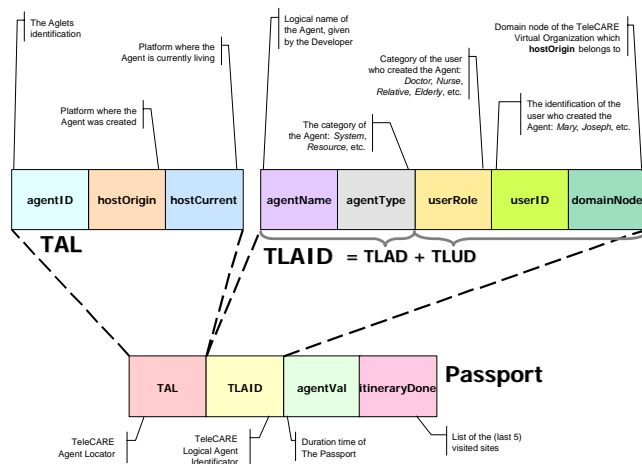


Fig. 6 – TeleCARE agent passport

As shown in Fig. 6 the passport is composed of the following fields:

- **TAL** – The *TeleCARE Agent Locator*, which is a

system identifier used to locate an agent; with the 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);

- **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; with the information provided by the **TLAID**, the developers can identify any TeleCARE agent, given its name, type, role or user ID, and/or domain node of the TeleCARE Virtual Organization that the origin host (or platform) of the agent belongs to. **TLAID** is composed of two substructures:
 - **TLAD** – The *TeleCARE Agent Data* that contains specific human readable identification of the agent, and
 - **TLUD** – The *TeleCARE User Data* that contains human readable identification of the user who created the agent;
- **Validity** – Used to assign the duration time of the agent passport;
- **Itinerary** – Used to store a list of the last 5 visited hosts.

Please note that the native AGLETS *AgletID*, which is a string of 16 hexadecimal characters, is not a user-friendly identifier. Furthermore, it might change if a clone replaces the original agent. **TLAID** is, therefore, a high-level user-friendly identifier. If only part of **TLAID** information is given, the result of the method `tcGetAgentTAL` can be a set of agents (from which the developer chooses the right one). This happens when a **TLAID** object is used to get the **TAL** or the passport of an agent. In many cases the result of the query is a set of agents.

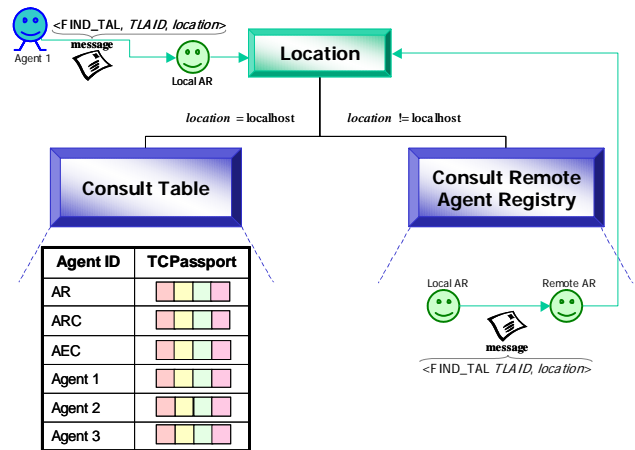


Fig. 7 – Finding and Agent or set of Agents

Mobility. The inter-platform mobility (IPM) module is a set of stationary agents that extend AGLETS mobility functionalities. These agents are:

- **Agent Registry** – Keeps a record of all agents that are living in the platform. The registration 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 **TAL** from the local (or maybe remote) Agent Registry.
- **Agent Reception Control** – 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 *Agent Reception Control* functions as a gatekeeper for incoming agents that want to enter into the platform.

- *Agent Exit Control* – Controls the outgoing of mobile agents. Every time an agent is to leave the platform, its passport is checked first to see if the agent has permission to travel, and if the destination of the agent is available and/or is a valid TeleCARE platform.

Given such functionality, the developers only have to focus on the agents they are developing. The IPM agents provide the extended mobility services and have their independent living in the platform. Also, they control the creation of the agents in the TeleCARE platform. The IPM agents are born and execute without intervention of the users, in the sense that if any of the services (application agents) are instantiated by the user (or by the platform at start-up), these agents will automatically launch themselves.

Inter-agent communication. The AGLETS system provides a simple mechanism for inter-agent communication. However this mechanism is not sufficient for reliable communication for highly mobile agents [9] or when persistence mechanisms based on cloning are implemented, namely due to the changes in the *AgletID*. Therefore, this module implements additional communication services:

- Extended message exchange mechanisms, which allow reliable inter-agent communication, in almost all cases. These mechanisms are based on the *forwarding scheme* described in [9, 10], with the difference that only the origin host of the Agent (where the Agent was created) is notified when it migrates to another location.
- FIPA ACL. For more advanced communication it is necessary to have a well-structured language; FIPA ACL (Agent Communication Language) is a well-known standard for inter-agent communication, which supports negotiation mechanisms and provides a good representation for structured messages exchange between agents.

A new class *TCMessage* is developed in order to allow the handling of several of these additional functionalities; this class acts as a wrapper of the Aglets *Message* [11], but enriched with the identification of the Sender and adding a method for handling exceptions between Agents in different (remote) platforms.

TeleCARE agents may also define message handlers for *TCMessage* (and ordinary Aglets messages) and FIPA ACL messages, which are shown in Fig. 8. These methods are:

- boolean `handleMessage(TCMessage msg)` – to handle messages, of type *Message*, coming from untrustworthy or non-TeleCARE agents.
- boolean `handleMessage(TCMessage msg, TCPassport passport)` – to handle messages, of type *TCMessage*, coming from trusted Agents with valid passport.

- `ACLMessage handleMessage(ACLMessage msg)` – to handle FIPA ACL messages coming from untrustworthy or non-TeleCARE agents.
- `ACLMessage handleMessage(ACLMessage msg, TCPassport passport)` – to handle ACL messages coming from trusted agents with valid passport.

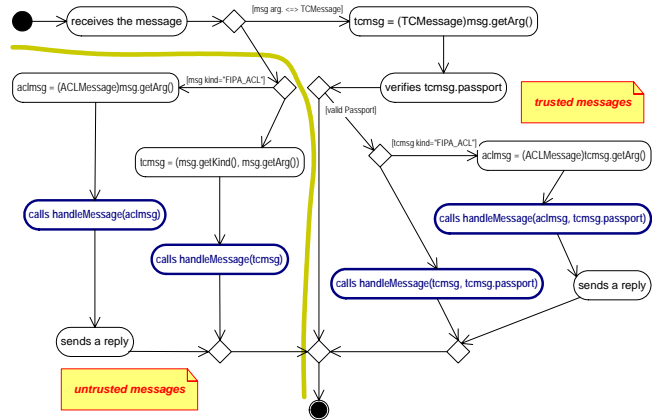


Fig. 8 – Handling messages in TeleCARE agents

Resources Management. For each resource (e.g. sensors and other devices) in each node, a specific agent – *Resource Manager Agent (RM)* – shall be available. This agent is responsible for granting access to the resource functionalities and for checking the access rights whenever another agent requests access. Fig. 9 shows a generic template of a resource manager.

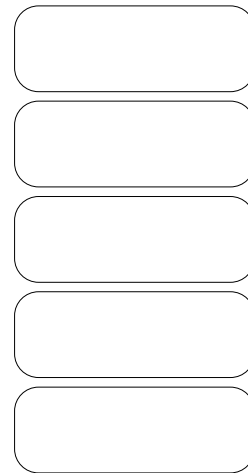


Fig. 9 – Template for a resource manager

This module works with the *Resource Catalog Manager*, which contains the models of the resources functionalities, the proxies and set of parameters to access those functionalities, as defined in the Device Abstraction Layer module (Fig. 10). The specific information items are:

- Abstract specification of the device,
- Functionality definition (UPnP specification) [12]: properties, functions, access rights,
- Device definition,
- Device services definition.

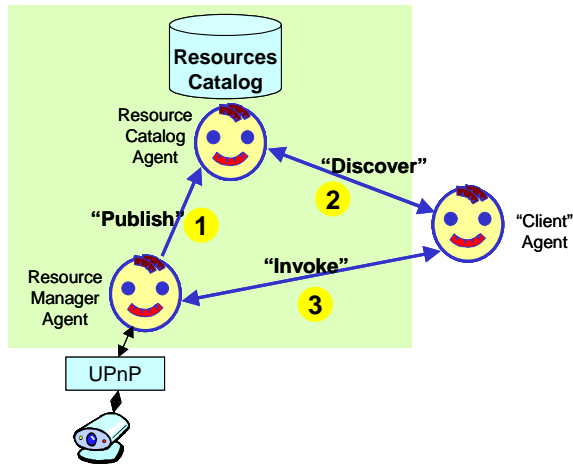


Fig. 10 – Resources registration and access

The RM agents are the ones that have direct interaction with the devices. Thus, the RM agents translate the requests encoded in the ACL messages into corresponding (low-level) actions that the devices must perform. A device, as seen by RM agent, is defined as following the UPnP specification. Therefore, the devices (and home appliances) in TeleCARE are described as follows:

- Each device provides a list of services that the resource manager can invoke.
- Each service, in turn, is characterized by its Service type, its ID and a Service Control Protocol Definition (SCPD).
- The implementation of each service is defined in device and in event control placeholders that, in the context of the UPnP specification, are URL addresses.

The device definition used by the Resource Manager Agents is obtained from the *Resource Catalogue*.

The interactions between the Vertical Service Agents and the RM, and between the RM and the corresponding device are defined by a set of methods, in correspondence to each block of Fig. 9. The following formal model [15] defines some of them.

Data types :

- SN* - Service Name
- AN* - Action Name
- SVN* - State Variable Name
- SVV* - State Variable Value.
- Value* - System data type (float, string...)
- SST* - The *Service State Table* for a device's service.
- SR* - Structure that specifies a Service Request.
- ESR* - Structure that specifies an *Event Subscription Request*.
- SVQ* - Structure that specifies a *State Variable Query*.
- Context* - Set of (application dependent) properties that define the actual context of the Resource.
- ACL* - Message received by the RM.

...

Authorization operators:

CheckAccess: Passport × *SN* → *Boolean*

...

State Variables operators:

Assign: SVN × *SVV* → *SVV*

Increment: SVN × *SVV* → *SVV*

...

PerformQuery: SVN × *SN* → *SVV*

DeviceEvent: SVN × *Value* → *SVV*

...

Services/actions operators:

Initialize → *SST*

Start: SST × *Context* → *SST*

Stop: SST → *SST*

ReceiveRequest: ACL → *SR*

PerformService: SR × *SST* × *Context* → *SST* × *ACL*

performQuery: SVN × *SN* → *SVV*

...

Intrinsic behavior operators:

ReadDevice: → *SST*

ExecAction: SR → *SST*

...

To access the resources of a TeleCARE node, the vertical service developers consult the agent *Resource Catalog Manager* and build adequate request messages to send to the respective Resource Managers. Suppose that we need to operate the device₁, controlled by the agent ResourceManager₁. After consulting the catalog we find out that we can execute the action₁, with the corresponding parameters a1 and a2, of the service₁ of that device. Then we simply make a request using the necessary information with the ACL's REQUEST performative. The order of the arguments is not important. Fig. 11 illustrates how this method can be used.

```
(request
  :sender (agent-identifier :name Agent_1)
  :receiver (set (agent-identifier :name
                                ResourceManager_1))
  :content
    "(run (service service_1)
          (action action_1)
          (parameter (a1 value1) (a2 value2)))"
  :language rms-s1
)
```

Fig. 11 – ACL message to request the "service₁.action₁"

Agent Factory. The Agent Factory (AF) is a sub-system composed of several modules performing two functionalities: (i) development side, where the AF acts as a CASE-like tool; it manages a library of agent skeletons that are used to help the programmer to develop new TeleCARE services/agents; and (ii) operation side, where the AF is intended to provide support services for the platform functioning. The former is intended for people with software engineering skills, helping developers to build TeleCARE agents. The latter provides a set of agent classes for basic tasks in order to ensure that people without software engineering skills are able to operate the system. These classes only need some parameter data to be instantiated.

Platform Manager. The Platform Manager is the component of the TeleCARE platform responsible for the configuration and specification of the operating conditions of the platform in each site. The main functionalities include:

- System configuration;
- System supervision, including a watch-dog mechanism, history registry, and monitoring;
- Definition of users and user categories.

Performing these operations implies the definition of a set of basic entities (e.g., users, roles...), which are represented as ontologies (using Protégé). A sub-component of the *Agent Factory* gathers the ontology definitions and transforms them into corresponding Java classes.

The agents launched by a given user inherit the user permissions (which were assigned by the administrator, or defined by the user role). Therefore, the agent access rights depend on the agent type, its role and the respective user.

B. Information management issues

Considering the sensitivity of the data exchanged within this type of VC, and in addition to the safe communication mechanisms provided by VPN (Virtual Private Network) technology, careful preservation of the information privacy and authorized access to data is of great importance. Therefore, proper definition of information access rights and authorization rules are mandatory. To properly preserve the definition of adequate levels of information privacy, the TeleCARE platform also supports a federation layer that extends the functionalities of standard database management systems - the *Federated Information Management Architecture* (FIMA) component (developed by the University of Amsterdam). The FIMA design is based on a federated database architecture, in order to support and reinforce the required level of autonomy and heterogeneity of each individual data source within the TeleCARE network.

The usage of both emerging standards (e.g. ODMG, XML, WSDL, etc.) and the emerging Web tools to support the required interoperability for federated information management (import/export/integrated schemas) and for information exchange in TeleCARE is followed as a design principle.

4. SERVICES IMPLEMENTATION

A. Approach

In the TeleCARE environment each vertical service can, in fact, be implemented as a set of distributed stationary and mobile agents. For instance, a monitoring service might involve a stationary agent in the care center (interacting with the care worker), a number of stationary agents in the elderly home (agents in charge of monitoring local sensors, e.g. temperature sensor, presence sensor), and a mobile agent sent from the care center to the elderly home (Fig. 12). This mobile agent might carry a mission to collect information from different sensors and report back to the care center.

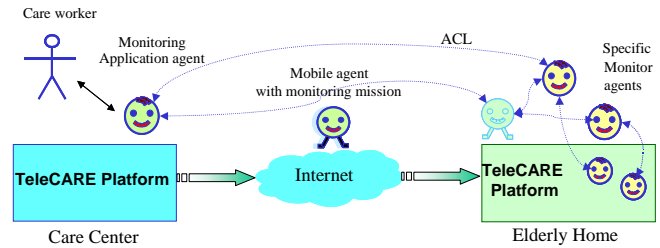


Fig. 12 – Example of a service implementation

The stationary agents in one platform can also communicate, via ACL messages, with other mobile or stationary agents residing at another platform. As the TeleCARE message includes the extended agent identification (see *passport* section), the destination of a message can check the identity and rights of the sender, no matter where it is running at the moment.

B. Time Bank service

One of the vertical services being implemented is the Time Bank. The Time Bank concept provides a mechanism for collaborative community building / reinforcement, i.e., provides a way for people to come together and help each other [13]. The basic principle is that each member “earns” service credits by performing tasks to others, in turn the members that request tasks “spend” service credits. Each credit corresponds to one time unit, i.e., if some service takes one hour to be completed the requester will consume one credit and the provider will receive one credit.

In TeleCARE, the Time Bank concept leads to one virtual community. Participants in this community include the elderly people, their relatives, and care center personnel. To these types of members three types of agents are associated, which perform the tasks of this vertical service (see Fig. 13).

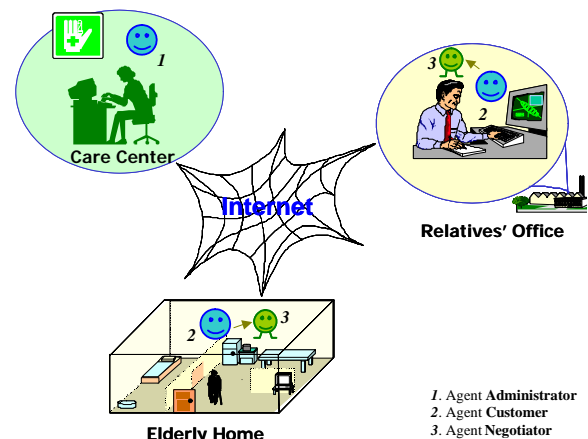


Fig. 13 – TeleCARE actors and associated agents in the Time Bank

The agent Administrator, associated to the care center that manages the Time Bank, is responsible for the customer subscription, account management and the matching of the required services to potential providers. Also, it provides a user interface in order to facilitate the Time Bank management. The agent Customer is

responsible to require the subscription to the Time Bank, request the customer's account statement, require (or accept to provide) a service, and to provide an interface with the human Time Bank members. It represents each user of the Time Bank in the virtual community. Finally, the agent Negotiator, which is created by the agent Customer when a user of the Time Bank requires a service, carries out the tasks of negotiation with the other agents in two phases: (i) with agent Administrator, which indicates who are the potential providers; and (ii) with the agent(s) Customer of the potential providers, in order to define the conditions of the service accomplishment. These interactions are illustrated in the AUML diagram [14] of Fig. 14.

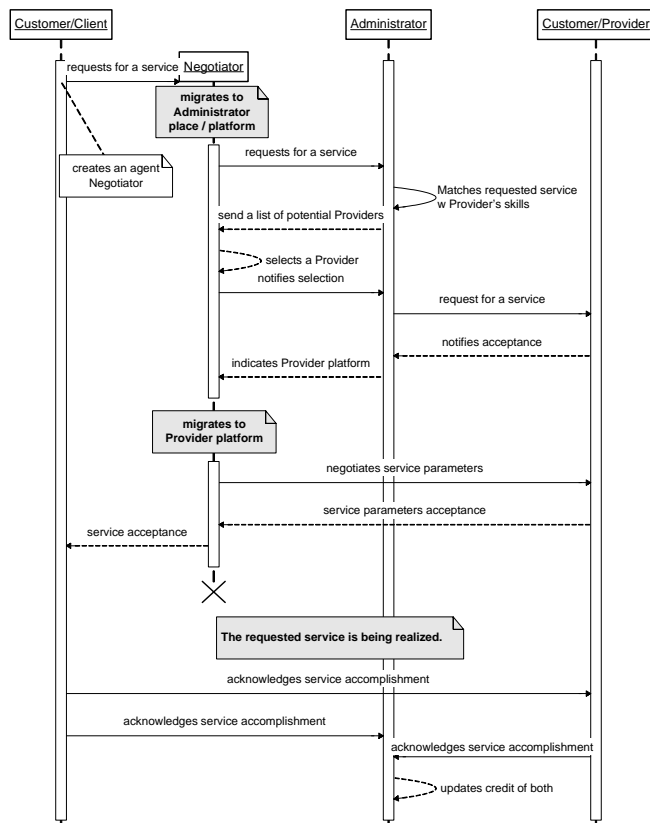


Fig. 14 – Time Bank's agents interaction

5. CONCLUSIONS

The growing numbers of elderly population impose an urgent need to develop new approaches to care provision. The convergence of a number of technologies such as multi-agent systems, federated information management, safe communications, hypermedia interfaces, rich sensorial environments, increased 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 particular, a platform based on mobile agents combined with federated information management mechanisms provides a flexible infrastructure on top of which specialized care services can be built.

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.

7. REFERENCES

1. Camarinha-Matos, L.M.; Afsarmanesh, H. - Virtual communities and elderly support, *MIV'01 in "Advances in Automation, Multimedia and Video Systems, and Modern Computer Science"*, (V.V. Kluev, C.E. D'Attellis, N. E. Mastorakis ed.s), WSES, ISBN 960-8052-44-0, pp. 279-284, Sept 2001.
2. Camarinha-Matos, L.M.; Afsarmanesh, H. - Design of a virtual community infrastructure for elderly care, *Proceedings of PRO-VE 2002 – 3rd IFIP Working Conference on Infrastructures for Virtual Enterprises*, Kluwer Academic Publishers, ISBN 1-4020-7020-9, Sesimbra, Portugal, 1-3 May 2002.
3. Camarinha-Matos, L.M.; Vieira, W. - Intelligent mobile agents in elderly care, *Journal of Robotics and Autonomous Systems* (Elsevier), Vol. 27, N. 1-2, April 1999, ISSN 0921-8890, pp. 59-75.
4. Camarinha-Matos, L.M.; Vieira, W.; Castolo, O. - Mobile agents approach to virtual laboratories, *Proceedings of IFAC Symposium on Artificial Intelligence in Real-Time Control*, Budapest, 2-4 Oct 2000.
5. Dutta-Roy, A. – Networks for homes, *IEEE Spectrum*, Vol. 36, N. 12, Dec 1999.
6. Saranummi, N.; Kivisaari, S.; Sarkikoski, T.; Graafmans, J. – Ageing & Technology – State of the art, Report for the European Commission, Institute for Prospective Studies, Seville, Spain, Nov. 1996.
7. Vlaskamp, F. – Social alarms go mobile: emergency assistance for mobile users, 1999, www.fernuni-hagen.de/FTB/aaate99/paper/99_67.htm.
8. Vieira, W.; Camarinha-Matos, L.M.; Castolo, O. – Fitting autonomy and mobile agents, *Proceedings of ETFA'01 – 8th IEEE International Conference on Emerging Technologies and Factory Automation* Vol. 2, IEEE Catalog Number 01TH8597, pp. 471-480, Antibes - Juan les Pins, France, Oct 15-18, 2001.
9. Murphy, A.; Picco G. P. – Reliable communication for highly mobile agents, *Autonomous Agents and Multi-Agent Systems* (Kluwer Academic Publishers), N. 5, 2002, pp. 81-100.
10. Vieira, W. – *Adaptable Mobile Agents for Remote Operation*, Ph.D. thesis (in Portuguese), New University of Lisbon, 2001.
11. *Aglets API Documentation*, v. 2.0.2; Java Aglet Community, <http://aglets.sourceforge.net/>.
12. *Universal Plug and Play Device Architecture*; Universal Plug and Play Forum, <http://www.upnp.org/>.
13. *What is a Time Bank?*; Time Banks UK, <http://www.timebanks.co.uk/>.
14. Odell, J., Parunak, H.; Bauer, B. – Representing interaction protocols in UML, in Ciancarini, P.; Wooldridge, M. (eds.), *Agent-Oriented Software Engineering* (Springer-Verlag), pp. 121-140, 2001.
15. Slonnegger, K., Kurtz, B.L. - *Formal Syntax and Semantics of Programming Languages: A Laboratory Based Approach* (Addison-Wesley Publishing Company, Inc.), 1995.