



**HAL**  
open science

## A two-layered virtualization overlay system using software Avatars

Mehdi Loukil, Badii Jouaber, Djamel Zeghlache

► **To cite this version:**

Mehdi Loukil, Badii Jouaber, Djamel Zeghlache. A two-layered virtualization overlay system using software Avatars. Computers and Communications (ISCC), 2010 IEEE Symposium on, Jun 2010, Riccione, Italy. pp.1086 -1090, 10.1109/ISCC.2010.5546606 . hal-00735990

**HAL Id: hal-00735990**

**<https://hal.science/hal-00735990>**

Submitted on 27 Sep 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Two-Layered Virtualization Overlay System using Software Avatars

Mehdi LOUKIL\*, Badii JOUABER\*† and Djamal ZEGHLACHE

(\*) TELECOM SudParis

(†) CNRS, UMR SAMOVAR

[{Mehdi.Loukil, Badii.Jouaber, Djamal.Zeghlache}@it-sudparis.eu](mailto:{Mehdi.Loukil, Badii.Jouaber, Djamal.Zeghlache}@it-sudparis.eu)

**Abstract**—Virtualization is a promising approach to meet the objectives of next generation networks. It will enable building pervasive systems, with the possibilities of automated adaptation and provision of personalized services for users with different needs and contexts. In this paper, we propose the use of *Avatars*: software delegates to hide heterogeneity between entities such as networks, users, services and any resource or device. *Avatars* act at an abstraction overlay level, on behalf of their represented entities. They carry information and intelligence (workflows) and cooperate to make decisions like the reconfiguration of the entity they stand for. A software agents-based implementation is then proposed and evaluated.

**Keywords** : Pervasive Systems, Overlay, Virtualization, Context-Awareness, Avatars, Software Agent Technology.

## I. INTRODUCTION

Traditional networks and services' management techniques are not adapted for future telecommunication systems, characterized by high levels of heterogeneity, dynamicity, and complexity. Service adaptation and personalization as required for the Internet of the Future are difficult to achieve since existing tools are mostly built for homogeneous environments.

It is predicted that future telecommunication systems will be composed of autonomous, communicating and intelligent objects, deployed everywhere in the life space. These will form the future *Internet of Things*, a pervasive system [1] that will ease dynamic production of value added services. Hence, automated service adaptation at both networking and applicative levels is needed. In this context, system awareness, automated reasoning and adaptation are considered as key

functionalities and enablers. These remain difficult to achieve over heterogeneous environments.

Pervasive computing is considered as a candidate response to this problem. It should allow automated and dynamic service creation and adaptation in complex environments. This is challenging due to the coexisting of multiple heterogeneous solutions in the networking and service domains (e.g. access and transport, protocols, execution environments, terminals ...etc.).

In this paper we propose the use of *Software Avatars*, **within** a distributed and mobile overlay solution, to ease the management of heterogeneous telecommunication environments.

The remainder of this paper is organized as follows. Problem statement and related work are presented in section II. In section III, we present the proposed virtualization solution, its details as well as the test bed for its evaluation. Finally, conclusions and future work are given in Section IV.

## II. PROBLEM STATEMENT AND RELATED WORK

Heterogeneity as considered in this paper is relative to three main domains: (a) the hardware domain (i.e. physical) that reflects the coexistence of multiple standards and technologies within physical devices like computers, routers, laptops, PDAs, and wireless access points; (b) the software domain (i.e. logical) that reflects the presence of different protocols, operating systems, service execution environments (i.e. middlewares); finally (c) the architectural domain (i.e. design) that reflects the multiple ways of interconnecting the above entities like distributed, centralized, mesh and ad hoc topologies.

Building autonomous solutions to manage the above heterogeneous environment remains challenging. For instance, existing management solutions like COPS (Common Open Policy Service) [2][3] and SNMP (Simple Network Management Protocol) [4][5][6][7] are more adapted to homogeneous, centralized and relatively small scale systems. They require common tools and protocols shared between all the entities within the system.

In this context, the use of a high level harmonized representation seems promising. For instance, the use of semantic annotations and ontologies is more and more tackled in the literature. In [8] Semantic Descriptions are proposed to express services and devices as functional components. This allows to dynamically attaching them to workflow models and enable reasoning. Ontologies are also proposed in [9][10] to allow automation and to enable reasoning possibilities. Authors in [10] present a service overlay architecture for next-generation networks. This overlay allows to offer personalized services that can be adapted to users' context such as his (her) current device, access network, preferences, or quality of service (QoS) requirements. Adaptation is achieved by dynamically establishing service level path to fulfill the user's requirements.

Unfortunately, most of the proposed solutions and approaches are presented for particular environments and use cases. At the best of our knowledge, these solutions propose new formalisms and architectures for future services and do not consider coexistence of today and future systems.

In the following, we propose a new approach based on delegation and overlay virtualization. It allows the construction of pervasive systems with both existing and up-coming technologies, devices and services whatever their heterogeneity is.

### III. PROPOSED ARCHITECTURE

We aim here to ease the management of a significant number of heterogeneous entities including users, terminals, services, networks, service providers...etc. The idea is to build a conceptual control level (see Figure 1), composed of two virtualization layers. This control level can make reasoning on behalf of physical entities within the system.

Virtualization is based on the use of *Software Avatars*. These are software entities, with communication and reasoning capabilities, able to act on behalf of the physical entity they represent.

The two virtualization layers are defined according to the dynamicity of contextual information (knowledge) they manage. The first virtualization level is defined to make reasoning on frequently changing contextual information. The second level is defined to make reasoning mostly on static contextual information. It has a global view on the entire system; it is also responsible of orchestrating the first one.

Like in [12], the proposed solution is a context-aware system. In addition, it is distributed, scalable, extensible and more resistant to failures.

Distribution is introduced through the definition of *Active Zones* within the first abstraction level. *Active Zones* can be defined to achieve performance and scalability objectives. For instance, an *Active Zone* can be defined to reflect geographic proximities of a set of interworking entities, or to bring together entities cooperating to achieve a given service over a fixed period of time. In the following, details about the proposed distributed architecture are provided. We first present a functional description. Then we present a possible implementation through the use of mobile agents.

#### 1) *First abstraction level*

The first virtualization level is composed of *Avatars*: autonomous software components that can represent any entity within the system. Entities may be persons (users), services, resources or any physical or logical entity (user terminal, network, network entity, network device, service provider ...etc). Each entity can delegate to its corresponding *Avatar* reasoning activities that require shared knowledge with other system entities.

*Avatars* are able to communicate through unified interfaces to exchange data and to participate to common decision making. However, communications between the physical entities and their *Avatars* can be proprietary allowing heterogeneous entities to co-exist within the system.

*Avatars* can carry work-profiles and adaptation rules relative to their correspondent entities. Together, they constitute a virtual space representing each physical entity within the

system. Additional avatars can represent logical entities and functions like mobility and security managers. They are responsible for decision making and adaptation. These decisions can be based on the analysis of entities' work-profiles and contexts (static information are gathered from the second abstraction level and dynamic information are gathered from the entity it self).

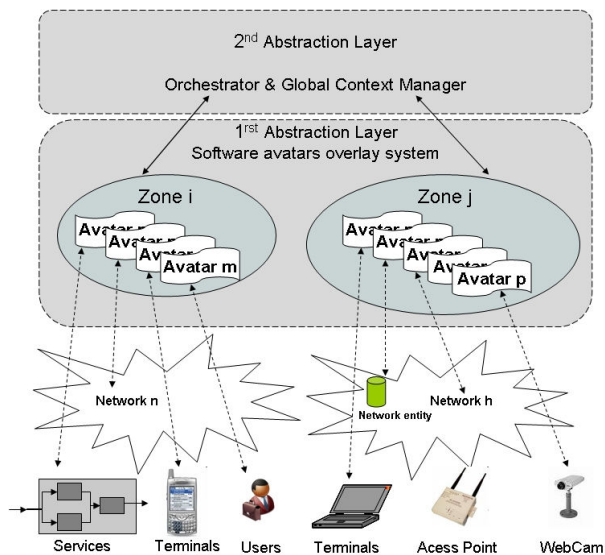


Figure 1: Top layers Virtualization Overlay Architecture

*Avatars'* mobility contributes to enhance system performance since context gathering and decisions can be made at the closest of the represented entity. Therefore, the 1<sup>st</sup> abstraction level is enriched through the use of *Active Zones* among with Avatars can move. These zones can be defined according to the target applications or scenarios. For instance, different active zones can reflect different applications or different geographic locations.

*Avatars'* mobility is useful if we consider that the represented entities can move from one *Active Zone* to another, because of changes within the context or because of geographic mobility of the entity for instance. To enhance system performances, *Avatars* can be created and moved in/to active zones according the contextual information of the represented entities.

Using *Active Zones* can ease the creation of value added services like:

- Printing a document on the closest printer according to user's location (based on location zones).
- Broadcasting messages to a given category of users (based on category zones)

- Adapting light in the house according to the journey time and/or user context and preferences (based on time zones)
- Helping service providers to supply adapted service components in different *Active Zones* (based on profiling zones).

As indicated above, an *Avatar* embeds and executes intelligence on behalf of the entity it represents, including processes for decision-making and adaptation. For instance, a mobile terminal *avatar* can run decision processes for network selection and vertical handover decision making. A video service *avatar* can run video adaptation according to the available throughput.

Therefore, *Avatars* should be able to communication with the correspondent physical or logical entity they represent. In addition, different *Avatars* can share common context and can cooperate to explore available resources in a given zone. *Avatars* should also be able to communicate with the second abstraction level, i.e. with the *Orchestrator* and the global *Context Manager*. This will allow them to provide, update and/or request context information, profiles, preferences, statistics...etc.

*Avatars'* functional architecture and their interaction with the 2<sup>nd</sup> abstraction level as well as with the represented entities are given in Figure 2.

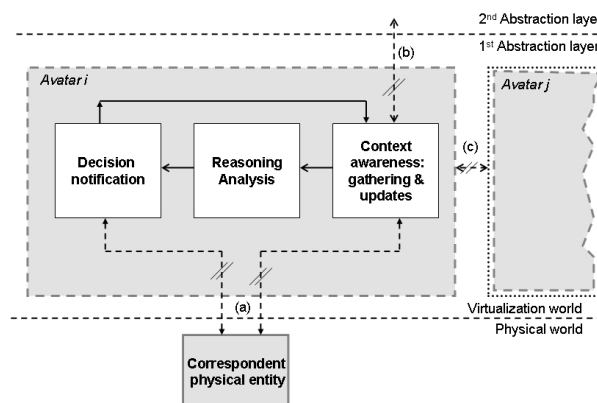


Figure 2: *Avatars'* functions and interfaces

## 2) Second abstraction level

This second abstraction level contains a global *Context Manager* and an *orchestrator*. These have a global view of the system. They offer a unified representation through ontologies allowing reasoning and inferring.

The use of ontologies enables automated reasoning and inferring modules. Another advantage is to

automate communications between system components including the *Avatars*, the *Orchestrator* and the correspondent physical entities.

Figure 3 gives a global overview of the proposed second abstract level, including the use of ontology [13].

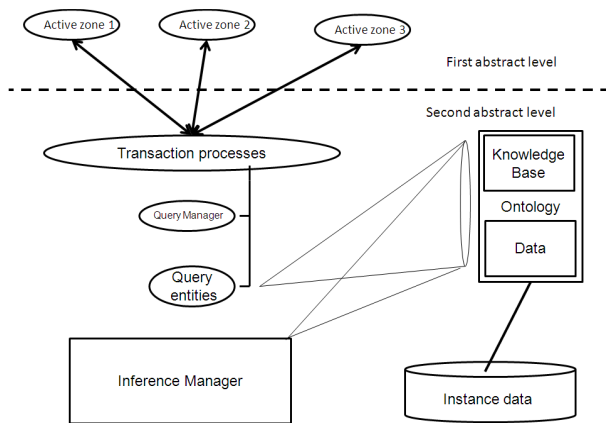


Figure 3: Second abstract level overview

The *Orchestrator* acts as a moderator to avoid concurrent or conflicting decisions. For instance, in a heterogeneous wireless access scenario, if the available bandwidth decreases for a given connection, the *avatar* representing the device may decide to handoff to another available access point (handover) while the *service avatar* may decide to change the service mode and degrade the QoS. In this case, the *orchestrator* can give priorities between the two actions or define an execution order.

The *reasoning and inferring engine* can provide the *Orchestrator* with inferred rules and knowledge based on the elementary ontology. This allows to make global reasoning to enhance system behavior and performance.

### 3) Implementation and performance evaluating

In order to evaluate the proposed solution, a real testbed is set using different technologies. A Multi-Agents sub-System (MAS) with JADE Environment [14] is used. *Avatars* and *Active Zones* are then implemented as mobile agents. This Multi-Agents technology is selected because it offers a multitude of advantages including practical deployment on existing and heterogeneous platforms. In addition, this technology allows to instantiate the *Avatars* within an initial *Active Zone* and to move them when

required to another *Active Zone* as specified in the proposed solution. Another advantage is the use of ACL (Agent Communication Language) for unified communications between software agents. Software agents can also carry a state machine that can reflect the work profile of the *Avatar* and its correspondent entity.

In addition, we implemented a basic context manager, an orchestrator and a basic ontology, using Protégé [15] and RacerPro [16] tools, for the second virtualization level. First experiments on the testbed proved the feasibility of our proposed solution. *Avatars* were easily created, moved from one *Active Zone* to another and communication was successful. Communications between the *Avatars* and their correspondent physical entities is achieved through the generation of ACL-like messages. The testbed also proved the possibility to adapt service logics based on the analysis of the context. This was possible by defining different agent behaviors (through state machines) that can be re-configured according to the context.

*Avatars* migration within different zones is achieved through agent mobility, allowing load balancing, higher scalability and optimized response time.

The proposed architecture is tested for video service adaptation based on user context. All devices and heterogeneous system entities were represented by their own avatars (agents) within the overlay system. A video streaming service is first initiated for the user according to his (her) profile. When network conditions changes (degradation is supposed here), the user' avatar detects it, analyses the new situation of the user and then invokes the video avatar by requesting a service adaptation. This behavior is totally automated since avatars have reasoning and communication capabilities. The virtualization overlay system successfully allowed higher flexibility and automated service adaptation.

Early analysis and measurements made on the platform showed that additional enhancements are possible to reduce the system response time. Two directions are understudy: optimizing inter-avatars communications and splitting context databases.

## IV. CONCLUSIONS

In this paper, we proposed a two layer distributed overlay architecture based on the use of *Software*

*Avatars*, a new virtualization concept we defined to deal with heterogeneous complex environments. The proposed architecture was successfully implemented using a MAS platform. Initial results showed that it is possible to virtualize heterogeneous entities including devices, services and users through *Avatars* within a first dynamic abstraction level. In addition, we proposed a second abstraction level to provide higher layer with a global view of the system. This allows avoiding conflicting and concurrent decisions and adaptations. The use of a unified ontology allows the automation of the reasoning and facilitates knowledge sharing between physical entities and between *Avatars*.

### REFERENCES

- [1] S. Loke, Context-Aware Pervasive Systems: Architectures for a New Breed of Applications. AUERBACH, December 7, 2006, ISBN-10: 0849372550.
- [2] R. Yavatkar, D. Pendarakis, and R. Guerin, "A Framework for Policy Based Admission Control," RFC 2753, Jan. 2000.
- [3] D. Durham, Ed., J. Boyle et al., "The COPS (Common Open Policy Service) Protocol," RFC 2748, Jan. 2000.
- [4] P. E. Miller and M. A. Miller, Managing Internetworks with SNMP, 3rd ed: M&T Books, Nov. 1999.
- [5] D. Perkins and E. McGinnis, Understanding MIBs. Englewood Cliffs, NJ: Prentice-Hall, 1997.
- [6] M. T. Rose and K. McCloghrie, How to Manage Your Network Using SNMP. Englewood Cliffs, NJ: Prentice-Hall, Jan. 1995.
- [7] W. Stallings, SNMP, SNMPV2, SNMPV3, and RMON 1 and 2: Addison-Wesley, Jan. 1999.
- [8] A. Bandara, , T. D. de Roure and G. Clemo, An Ontological Framework for Semantic Description of Devices, International Semantic Web Conference (ISWC), Hiroshima, Japan, 2004.
- [9] A.K. Dey, Providing Architectural Support for Building Context-Aware Applications, PhD. Thesis, Georgia Institute of Technology, 2000.
- [10] Project Aura: toward distraction-free pervasive computing, D. Garlan D.P. Siewiorek, A.Simailagic, P. Steenkiste, IEEE pervasive computing. ISSN 1536-1268, CODEN IPCECF, vol. 1, no2, pp. 22-31, 2004
- [11] E. Lavinal, N. Simoni, M. Song and B. Mathieu, A next-generation service overlay architecture. Annals of Telecommunications, Vol. 64, No 3-4, Springer, 2009.
- [12] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell and K. Nahrstedt, A Middleware Infrastructure for Active Spaces. IEEE Pervasive Computing. Volume 1, Issue 4 (October 2002); Pages: 74 – 83; ISSN:1536-1268.
- [13] N. Guarino, Formal Ontology, Conceptual Analysis and Knowledge Representation, International Journal of Human-Computer Studies, 43(5-6):625–640, 1995.
- [14] G. Rimassa, F. Bellifemine and A. Poggi, "JADE - A FIPA Compliant Agent Framework", PMAA'99, p. 97-108, April 1999.
- [15] Protégé, a suite of tools to construct domain models and knowledge-based applications with ontologies. <http://protege.stanford.edu/overview/index.html>, (visited February 2010)
- [16] RacerPro: a Reasoner for the Semantic Web languages OWL/RDF, <http://www.racer-systems.com/products/index.phtml> (visited February 2010)