Agent-Based Virtual Communities for Interactive Digital Television

Federico Bergenti, Lorenzo Lazzari, Agostino Poggi
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Parma
Parco Area delle Scienze 181/A, 43100 Parma, Italy
{bergenti, lazzari, poggi}@ce.unipr.it

Abstract

This paper describes a multi-agent framework designed to support the creation and effective management of virtual communities in an Interactive Digital Television (IDTV) scenario. The possibilities that this framework offers are demonstrated by means of two sample applications: a real-time community game and an asynchronous auction. For the sake of completeness, the paper also presents an overview of IDTV technologies.

1. Introduction

Nowadays, a migration from analogue to digital TV is taking place in TV. This change has two main implications: the capability to broadcast more channels in the same bandwidth, and the possibility to send software applications mixed with audiovisual contents. These two great advantages have permitted the great diffusion of this new technology, which is becoming a new power means to develop new types of services.

In this paper we present our multi-agent framework, developed starting from the idea to integrate the technology of Interactive Digital Television (IDTV) with the concept of virtual community, which we can define as a technology-supported cyberspace, centered upon communication and interaction of participants, resulting in a relationship being built up. So, with this type of integration, our aim is to offer to IDTV users, a range of services (such as multiplayer games, on-line auctions, etc.) which are very common if we think to the idea of virtual community related to the Web.

In this way, the potentialities of interactive DVT can enormously grow allowing its users to take advantage of a new number of useful applications and moving the concept of interactivity from the simple interaction user-application to a new type based on the cooperation among a wide number of users.

2. IDTV

Interactive TV is a technology which combines broadcast video, broadcast radio, computing power and the Internet. This combination of different mediums and services provides the viewer with a new experience. This is possible because of an ongoing transition from analogue TV to digital TV.

We can clearly say that the digital technology is driving television towards a new world of amazing possibilities, where spectator is no longer limited to observe contents selected by the operator. More and more, new dynamic and interactive services are being introduced in everyday digital TV: complementary information to audio-visual contents, electronic program guides, selection of properties in configurable contents (language, camera angle or particularized advertisement), pay-per-view, etc. So we can consider the term "interactivity" as the possibility for the consumer to actively influence the behavior of broadcasted television, services and applications. This can be accomplished, for example, by means of a remote control for channel hopping, by fetching information via teletext or by sending data via an interaction channel. This all creates a context, which allows to have a mutual influence between the viewer, broadcaster and application provider.

The interactive TV technology, as we will see in next section, is based on the broadcasting of a digital transport stream which permits operators to mix traditional audio-visual contents with binary data, so making possible to deliver multimedia applications to be executed in a digital TV or in a set-top box. These applications, synchronized with audio-visual contents, adapt themselves to spectator characteristics, implement interaction with users and provide return channels for communication with content providers.

The Multimedia Home Platform [1] is a standard published by the DVB (Digital Video Broadcasting) consortium in 2001, which consists of a combination of broadcast and Internet, offering a common Application Programming Interface (API) accessible for everyone who wants to develop applications, set-top boxes, television devices or the combination of all.

Fundamentally the MHP standard defines a generic interface between interactive digital applications and the terminals on which those applications execute. This interface decouples different provider's applications from the specific hardware and software details of different MHP terminal implementations.

The MHP extends the existing, successful DVB open standards for broadcast and interactive services in all transmission networks including satellite, cable, terrestrial, and microwave systems.

The applications downloaded to the MHP terminals, typically set-top boxes, are Java applications called Xlet, built on a suite of APIs tailored specifically for the interactive TV environment: Java TV APIs [2], HAVi (user interface) [5], DAVIC APIs [4] and DVB APIs [3].

The 1.1 version of the standard defines three profiles:

1. Enhanced Broadcast: it is the basic profile which only allows the enrichment of the audio-video contents with information and images which can be viewed and navigated by users on the TV screen:
2. Interactive Broadcast: it is the intermediate profile that uses the set-top box return channel to supply services with a higher level of interactivity. In fact this profile supports the loading of MHP applications not only through the broadcast channel but also through the return channel;
3. Internet Access: this profile, using the return channel, allows the user to access to the Internet contents.

As we can understand from the previous description of the MHP levels, the interactive TV paradigm is based on two different channels: a broadcast channel from the application/contents provider to the set-top box and a return channel (dial-up, GPRS, ADSL, Ethernet, etc.) from the set-top box to the provider.

Figure 1 shows the use of a carousel to continue play-out a Java application. The application and the corresponding audio-visual material are then multiplexed to form a single MPEG-2 transport stream.

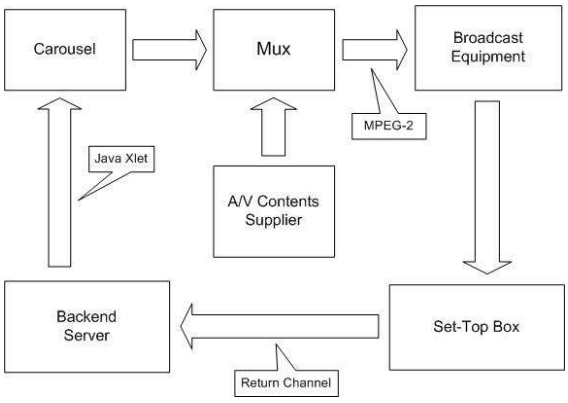

Figure 1. The interactive broadcasting chain

The resulting broadcast is received and decoded by the set-top box, the audio-visual content played and the Java application run.

Subsequent user interactions with the application lead to information being sent via the return channel to a back-end server. Depending on the application, this information may result in modifications to the current application content (i.e. voting information) or stored for later processing in a database present on the server (i.e. for an online shopping application).

About the transport, in digital TV MPEG-2 is not only a standard for encoding audio and video, but it is also used as the means by which raw data and applications are transported in the broadcast stream. In particular, DVB has extended the traditional scheme and way to use MPEG-2 for MHP by specifying how to embed a Java application within the stream, this includes information on how to specify the main class, class search path and the application argument list etc.

Although MPEG-2 provides a means of transporting the Java applications along the audio-visual content, to support the possibility that the user may change channel and select the Java program at any point of the transmission, the same application has to be broadcasted in loop. This is exactly what a broadcast carousel does: it keeps playing the same application around and around. The application is continuously multiplexed with the audio-visual content for the transmission, to allow the viewer to access to the interactive TV application whenever he wants.

About the applications, as we said previously, we have Java applications, but they are not complete Java applications in the normal sense. These applications are much more like applets in that they are loaded and run by a life cycle manager residing on the set-top box.

3. MHP-based Virtual Communities

In spite of the great research interest collected in the last years and the high number of functionalities already supported, in these days the research groups that work on the IDTV MHP standard are focusing their interest especially on the personalization of the IDTV contents on the base of the analysis of the user profile and preferences.

In accordance with our point of view, at the moment what is totally absent it is the collaborative aspect, that is the integration, in the digital television technology, of particular types of services to support groups of users joined by particular types of interests or necessities. These types of services are very common on the Web, we can think about the enormous number of forums, of blogs or of general services which allow a direct interaction among their users (on-line auctions, multiplayer games, etc.).

So the starting point from which our project has risen has been the aim to enrich the IDTV paradigm based on MHP and described in the previous sections with the introduction of the concept of "virtual community" very common on the Internet network.

A generally agreed upon definition of a virtual community would be a good starting point. What we need is a working definition of the virtual community, a consensus found in the major stream of literature, a definition that understood by most of people.

In his definition of a virtual community, Howard [6], the primary early advocator of virtual communities and often quoted in the literature, includes factors that describe a virtual community as a social aggregations that emerge from the Net when enough people carry on those public discussions long enough, with sufficient human feeling, to form webs of personal relationships in cyber-space. Hagel and Armstrong [7] focus on the content and communication aspects with special emphasis on member generated content: for them virtual communities are computer-mediated spaces where there is a potential for an integration of content and communication with an emphasis on member-generated content. The definition from Jones and Rafaeli [8] uses the term "virtual public" instead of virtual community. In particular, they say that virtual publics are symbolically delineated computer mediated spaces, whose existence is relatively transparent and open, that allow groups of individuals to attend and contribute to a similar set of computer-mediated interpersonal interactions. Another interesting point of view is the Romm and Clarke's [9] definition, which points out only the aspect of communication, that is via electronic media: virtual communities are groups of people who communicate with each other via electronic media, rather than face to face.

In literature we can find a lot of other definitions, but we can find some common aspects. The first similar point is cyberspace. All of the definitions state that the virtual community should be on the net, use computer-mediated spaces, or cyberspace. This point differentiates the virtual community from a real community. The second aspect in common is the usage of technology to support the activities in the virtual community. The different definitions directly or indirectly emphasize that access to the virtual community is through the computer or electronic media, i.e., technology. The third similar aspect is that the content or topics of the virtual community are driven by the participants. As mentioned, the participant driven community, not the web site coordinators, clearly distinguishes the virtual community from online information services. The final shared aspect is the successful virtual community relationship culminating after a certain period of communicating together.

To sum up, a working definition of a virtual community could be: a technology-supported cyberspace, centered upon communication and interaction of participants, resulting in a relationship being built up.

With our framework, in which the idea of virtual community is integrated with the interactive digital TV technology, we focus our interest especially on the second of the common aspects that define the virtual community concept: the support technology. In fact, we increase the horizons and the possibilities of the virtual communities by giving new types of services based on a new and more user-friendly technology like the IDTV.

In fact, the possibility to integrate the increasing IDTV technology with the idea of virtual community can give two great profits: on one hand we have a large increase of the digital television potentialities, opening new ways of communication and new types of services for the IDTV users; on the other hand, consequently, we give the possibility to enter in a virtual community taking advantage of his services also to a user range, the IDTV users, that sometimes can have not enough ability to surf the Web.

We can say that the integration of the digital television with the paradigm of virtual communities can extend the basic concept of interactivity, moving it from a simple logic user-TV to a more interesting logic based on the interaction user-user or user-community of users.

In particular, the ideas at the base of the development of our framework have been principally two: the support for community games and a more wide support for virtual communities involved in cooperative activities such as on-line auctions.

The technology used is a multi-agent technology, this because the intrinsic characteristics of multi-agents systems and of the agents themselves, such as proactivity, make them very proper to our scope.

4. The Framework

Agents need resources to act and to communicate. In FIPA [10] specifications, the run-time support providing such resources is the agent platform. Agents can run only in the scope of an agent platform providing the basic services to support interoperability: a means for sending and receiving messages and a means for finding agents, i.e., white pages and yellow pages. We do not request the platform to provide any support for concepts from agent-oriented software engineering such as autonomy or service-level interoperability. Basically, the platform is only meant to support the typed-message agent model.

Agents communicate explicitly sending messages and such messages may reach either agents within the same platform or agents on different platforms. This difference must be transparent to the developer and a fundamental characteristic of agent platforms is enabling this to support open societies where agents running on different platforms can join and leave dynamically.

The distribution and cooperation of agents residing on different platforms implies the conformance to a standard. At the moment, only FIPA is producing specifications for agent platforms.

At the moment, a number of FIPA platforms are available [11, 12, 13, 14], our middleware is developing the enabling technology for allowing the seamless deployment of agents to the Java-enabled IDTV devices such MHP-compliant set-top boxes.

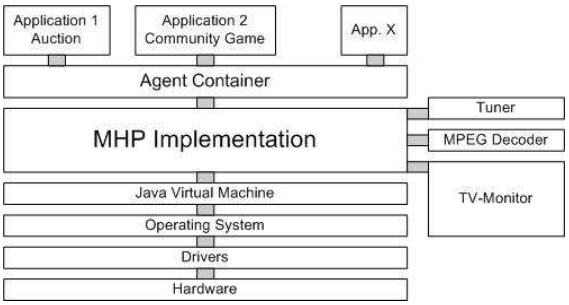

Figure 2. Client side architecture

Our framework is deployed as a multi-agent platform which we can split in two main sides: a server and a client side. The server side is set on a web server and it is deployed using the standard FIPA specifications, instead the client side is the more innovative one, because, since it is set on the set-top box, it requires to enable FIPA Agents on these types of devices.

In the next sections we give a first description of the platform architecture, starting from the client side, and then we will talk about the behaviour of the global platform, giving some example of virtual communities support.

4.1. Client side

The agent container set on the client side must be flexible enough to allow the integration of new services for the virtual community users. For this reason, we think that the best choice is to conceive the client-side of our framework as a MHP interactive application.

In the DVB MHP standard, applications are executed in the context of concrete services or events in a service, and, usually, they do not survive after finishing that context. In order to support services for virtual communities, we have to take into account that our system needs to store all the viewers' preferences about a particular topic (i.e. the user profile in a community game). So our approach integrates a special agent, named User Agent, which has the basic roles to work as an interface between the user and the rest of the system and to store the user preferences.

The User Agent is responsible of building the user profile, maintaining it when its user is on-line and notify to the system when his related user is active. The communication UA-user is performed by a standard GUI by which the user can manage his profile and the different services. Clearly, on the other side, the communication between the UA and other agents is based on FIPA specifications.

In order to support particular services for virtual communities, such as the possibility for a user to delegate to her/his personal agent the negotiation of a price in an on-line auction, this basic type of agent is always active on the user device.

The framework allows the development of other types of agents to guarantee other particular types of services, but for the moment our idea of the client side is that it must be based on "thin" software, so the reasoning mechanisms for the moment are delegated to the server side agent platform.

4.2. Server side

The server side of our framework consists in an agent container set on a standard Web server connected with the clients through the return channel of the set-top boxes.

In order to support services for virtual communities, the server side of the system has to include at least five different types of agents: a SP Agent (Set-top box Proxy Agent), a MP Agent (Mux Proxy Agent), a User Profile Manager, one or more Service Agent and a Directory Facilitator.

The SP Agent represents the interface between the server side multi-agent architecture and the client-side device: this agent receives the requests which came from the User Agent set on the user set-top box and manages them interacting with other kinds of agents.

On the other side, we have another proxy agent, called MP Agent, which is responsible to update the state of the application and to notify it to the Multiplexer, in order to update the raw data related to the Xlet embedded in the MPEG-2 stream and, consequently, the state of the interactive application displayed on the user's TV screen.

Between the two proxy agents have a specific kind of agent, named Service Agent, which is responsible of a particular type of service offered by the framework to the virtual organization. In example, if we think to a multi-player game, the Service Manager related to this type of service will be responsible to manage the state of the game, to find one or more appropriate partners to play, etc.

The User Profile Manager agent is responsible of maintaining the profile of the users and the information/preferences of the users themselves in relation to the particular types of services offered by the system (i.e. game preferences, skill level, etc.).

In the end, the Directory Facilitator is responsible to inform an agent about the address of the other agents of the system.

Figure 3 gives a graphical representation of the architecture of the system, focusing both on the interactions between agents and between the different devices. In figure 3 groups of three agents means that there can be one or more agents of that type.

5. Sample Services

In this last section we give some example of services supported by our system. In particular, as we said when we introduced our framework, the ideas at the base of the development of our system have been principally two: the support for community games and a more wide support for virtual communities involved in cooperative activities such as on-line auctions.

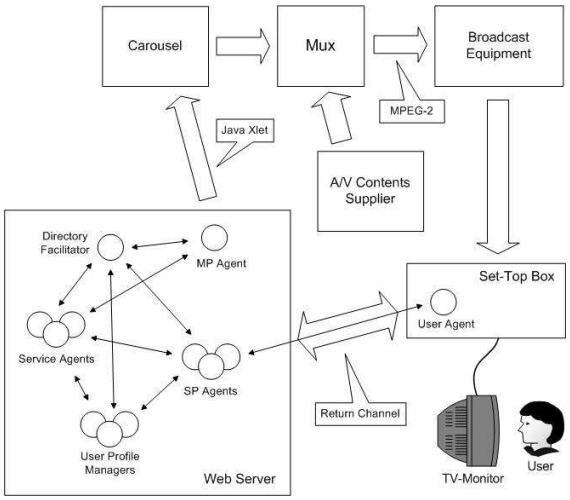

Figure 3. Architecture of the system

5.1. Community games

The idea to play a game in a virtual way with other people connected by a network or, in general, by a technology supporting the real-time interaction between the game participants is very common and diffuse on Internet. With our system we match this idea with the TDV interactive television, allowing IDTV users to play a community game without using any type of computer and of network, but through their IDTV device.

To describe quickly the system behavior relatively to such type of service, we can consider a simple type of game like "Othello", which requires two players. When an IDTV user wants to play an Othello match versus another user he has fundamentally to complete two steps before starting the match: the service configuration and the choice of the opponent. The service configuration is a task that the user has to perform only the first time she/he uses the application: the user has to insert some information like the game preferences, the skill level, etc. Once the game has been configured, the User Agent communicates them to the server side of the system to update the user profile managed by the User Profile Manager agent.

At this point the user is able to play: when she/he run the game by his set-top box, the User Agent notifies the server-side that his associated user wants to play. At this point the Service Agent related to that game creates a new game instance and the User Profile Manager agent find a possible opponent (the other user

has to be "on-line" and has to be a compatible skill level).

Once the opponent has been chosen, the match can start: the system, e.g. the Service Agent, continuously updates the state of the application in relation to the moves made, one after the other, by the participants until the end of the match. Obviously, in relation to the result, the system updates users' profiles.

5.2. Online auctions

Also the paradigm of the on-line auctions is very common for the Web users, we can think about the famous eBay Web site to quickly understand the enormous success that these types of services have collected in the last years. The behavior of the system is very similar to the previous case, in the sense that also for this type of service the user has to make an initial configuration of the application inserting her/his data which are used to update his profile.

Differently from the community game, in this type of service we have not a real-time interactions among the involved users but we have an asynchronous communication. When a user wants to sell something she/he opens a new auction inserting the initial price, the deadline, etc., then the User Agent notify the server side of the system and the Service Agent related to this type of service creates a new auction instance. From this moment all the users using this type of service can participate to the auction making their offers or selecting a maximum budget and delegating to their related User Agent the task. Once the auction has expired, the Service Agent deletes the related auction instance and the User Profile Manager agent updates the user profiles related to the involved users.

6. Conclusions

This paper presents a multi-agent framework that we realized to support the effective and fruitful implementation of virtual communities in an Interactive Digital Television scenario. Our framework gives IDTV application developers and service providers the possibility of running virtual communities to support the realization of interactive end-users applications, e.g., real-time multiplayer games and on-line auctions.

We strongly believe that the tight integration between IDTV and virtual communities that our framework provides can put a new perspective on IDTV. On the one hand, our framework opens new ways of communication and new types of services for the IDTV users and, on the other hand, it expands enormously the range of users that are possibly reached by everyday Internet-based virtual communities.

At the moment, our framework is under development. For the server-side of our multi-agent system we are using JADE (Java Agent DEvelopment Framework) [11, 15], which is a software framework to aid the realization of agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems. Client-side is based on new, yet somehow consolidated, IDTV technologies, e.g., MHP.

Our future work is related to the development of new types of applications and services expanding the functionalities and the multi-agent architecture of the framework.

References

[1] MHP Site *http://www.mhp.org*

[2] JavaTV Site *http://java.sun.com/products/javatv*

[3] DVB Site *http://www.dvb.org*

[4] DAVIC Site *http://www.davic.org*

[5] HAVi Site *http://www.havi.org*

[6] R. Howard. *The Virtual Community: Homesteading on the Electronic Frontier*. Addison Wesley, 1993.

[7] J. Hagel, A. Armstrong. *Net Gain: Expanding Markets through Virtual Communities*. Harvard Business School Press, 1997.

[8] Q. Jones, S. Rafaeli. "Time to Split, Virtually: 'Discourse Architecture' and 'Community Building' as means to Creating Vibrant Virtual Metropolises". *Int'l J. Electronic Commerce & Business Media*, 10(4), 2000.

[9] C. Romm, R. J. Clarke. "Virtual Community Research Themes: A Preliminary Draft for A Comprehensive Model". *Procs. 6th Australasian Conference on Information Systems*, 1995.

[10] FIPA Site *http://www.fipa.org*

[11] F. Bellifemine, A. Poggi, G. Rimassa. "Developing Multi-agent Systems with a FIPA-compliant Agent Framework". *Software Practice and Experience*, 31:103-128, 2001.

[12] J. Heecheol, C. Petrie, M. R. Cutkosky. "JATLite: A Java Agent Infrastructure with Message Routing", *IEEE Internet Computing*, Mar./Apr., 2000.

[13] C. Petrie. "Agent-based Engineering, the Web, and Intelligence", *IEEE Expert*, 11(6), 1996.

[14] F. Bergenti, A. Poggi, B. Burg, G. Caire. "Deploying FIPA-compliant Systems on Handheld Devices". *IEEE Internet Computing*, 5(4):20-25, 2001.

[15] JADE Site *http://jade.tilab.com*