Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

SCOTIMM
A System for Structured Context-Aware Interactive MultiMedia Applications[1]

João Benedito dos Santos Junior
Rudinei Goularte
Gustavo Blengini Faria
Edson dos Santos Moreira

Nº 146

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Julho/2001

[1] The ideas presented in this work are supported by FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo - and CNPq - Conselho Nacional para o Desenvolvimento Científico e Tecnológico.

SCOTIMM
A System for Structured Context-Aware Interactive MultiMedia Applications

Abstract. The possibility of using multimedia objects in modern applications has proven to be a bonus. We want to provide ways in which the user can interact with multimedia objects while it is been presented with features that go beyond the regular controls of a VCR. Furthermore, the presentation of the video can be modified by events happening in the surrounding environment. This technical report presents a system SCOTIMM - Structured ConText-Aware Interactive MultiMedia Applications - as a proposal to model the user actions from contextual information, in a way to ease its interaction with context-aware interactive applications like Interactive TV (ITV). The work also discusses how and which kind of contextual information could be used in one ITV environments that have an active network as the underlying infrastructure. The *profile* concept is presented and methods for using profiles in ITV programs are discussed.

1 Introduction

With the arising of the interactive video technologies, a number of new applications and interaction forms became possible. The different ways that the user can interact with the video (or multimedia objects) make a need how applications can adapt (even automatically) to the situations, improving interaction. The computation presence in ubiquitous way gives to the users the expectation that it is possible to access information and services anywhere. Besides that, the mobility provided by the ubiquity makes the user context, like the location, people and objects around, become more dynamic. The great variety of situations in which the user can be involved makes necessary a way to the applications adapt (eventually in an automatic way) themselves according to the situations, providing a better support to the human-computer interaction (McCanne et al., 1996; Dey and Abowd, 2000).

A way to improve the support to the human-computer interaction is to improve the communication during the interaction, making the computer able to process the contextual information of the user, the machine and the system communication, allowing the implementation of more useful computational systems (context-awareness).

Context-aware applications use environmental context inputs to provide information to the user or to enable services for him/her. Also information to the application and the network can be provided, to adapt (maybe automatically) the application infrastructure to the actual conditions.

In interactive video applications, context-awareness support can be split into these categories:

- Dynamic control of individual objects. New video technologies allow multimedia scenes to be composed by different video objects. These objects have the potential to be manipulated independently of the other objects in the scene. To do this, descriptions (shape, content, action and relationships) providing semantic to the objects must be available, easing that the application be aware of which actions can be and cannot be applied to the objects;
- Application dynamic adaptation to the network and server conditions. The interactive video interfaces must provide a mechanism to the user specifies the presentation quality. Once the user specifies the quality, he/she hopes that this quality stays the same during the whole presentation;
- Application dynamic adaptation to the user context. There are situations where the interfaces can adapt itself according to the user context. Depending on parameters like identity, location, action and time, services can be or cannot be offered or executed.

The second category helps content and network providers. The two others bring direct benefits to the end-users. Traditionally, events and relationships in computational environments are analyzed and modeled to build interfaces for applications. As computing is used ubiquitously, the design of interfaces must change to demand further capabilities both from the designer and the infrastructure available. Furthermore, the context-awareness offers new possibilities of interactions related to users, applications and infrastructure. These interactions and relationships need to be modeled to ease the building of interactive context-aware distributed environments, like ITV.

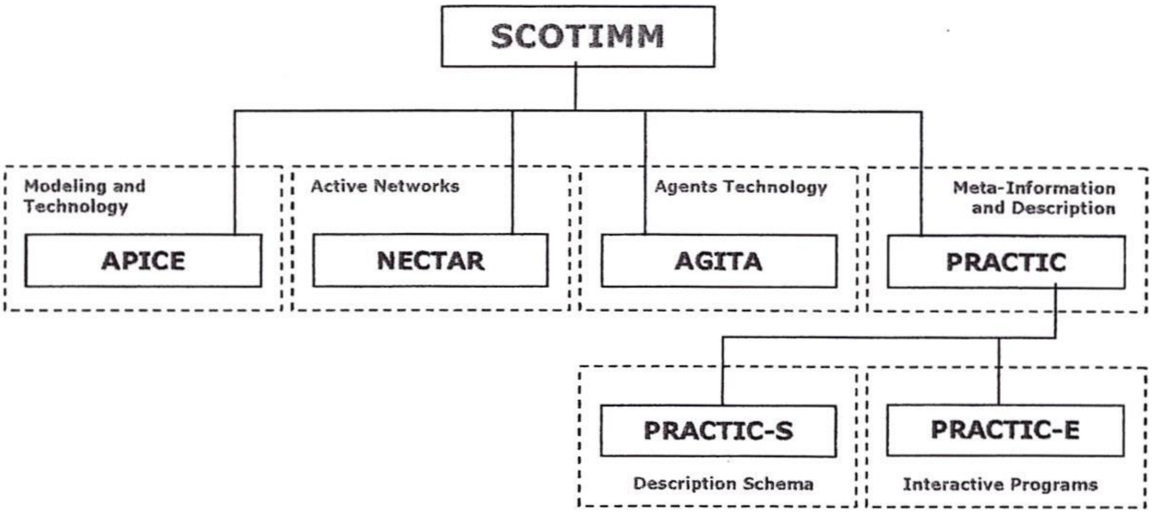

Figure 1: The SCOTIMM components - Graphical Structure.

In this context, was born the SCOTIMM - A System for **S**tructured **CO**n**T**ext-Aware **I**nteractive **M**ulti**M**edia Applications. The **Figure 1** presents the conceptual structure of the SCOTIMM.
The SCOTIMM system is composed by the following components:

- APICE (Structured Context-Aware Interactive Applications): covers the modeling of structured context-aware interactive applications, specifying methods for integration of technologies like MPEG-4, MPEG-7, XML, among others, and the approaches for development of interactive applications like Interactive Television, Videoconference, among others.
- NECTAR (Network Environment with ConText-Awareness Rules: covers the infrastructure necessary to context-aware interactive applications, especially in terms of the networks aspects;
- AGITA (Agents for Context-Aware Interactive Television Applications): covers the use of the agents technology in context-aware interactive applications;
- PRACTIC (Profile for Context-Aware Interactive Television Applications): covers the specific scope of the generation of profiles for interactive applications; it is composed by PRACTIC-E (PRACTIC Program Examples) - that covers the specification of profiles for interactive program examples - and by PRACTIC-S (PRACTIC Schema) - that covers the specification of description schemes for interactive program examples.

The use of active networks makes possible new approaches on user-network and application-network interactions, and represents an innovation in context-aware ITV applications.

The potential, represented by the active networks, is used in active nodes to process contextual information about network, server and applications actual needs, allowing the network to make dynamic decisions (based on these contextual information) about which is the better way to deliver ITV programs, in terms of the traffic, the bandwidth, the quality of presentation, the level of interactivity, the programs needs and the user interactions. These decisions can be taken at the moment that relevant events occur, without the user intervention or server response. The network is aware of the ITV application's context.

The last component is about the *terminal*, which is the interface between the user and the ITV application. It needs to be able to receive and process context-awareness information, and negotiate dynamically, with the server and with the network, the requirements and status of the application. The terminal also will exhibit the presentation, composed by MPEG-4 objects. Therefore, the terminal must have MPEG-4 player. The MPEG-4 standard provides an API called MPEG-J that can be used like an interface between MPEG-4 player and Java programs (Nack and Lindsay, 1999). This API can also be used to implement the server dynamic negotiation needs.

The proposed infrastructure and the ITV Application (our target application) generate a new and complex set of possible interactions. This scenario needs modeling, which will be discussed in the next section.

3 A Modeling for Context-Aware Interactive TV Applications

This section presents aspects covered by **APICE**. The ITV application model was divided in three major components and two other components that are related to all parts. The modeling, presented in **Figure 2**, follows an UML-like (*Unified Modeling Language*) syntax. UML (Booch et al., 1996) is used in the modeling of objects classes on the context-aware ITV application to allow the formal specification of these classes and their relationships, in terms of the activities (actions) that are generated during the use of the application by the user.

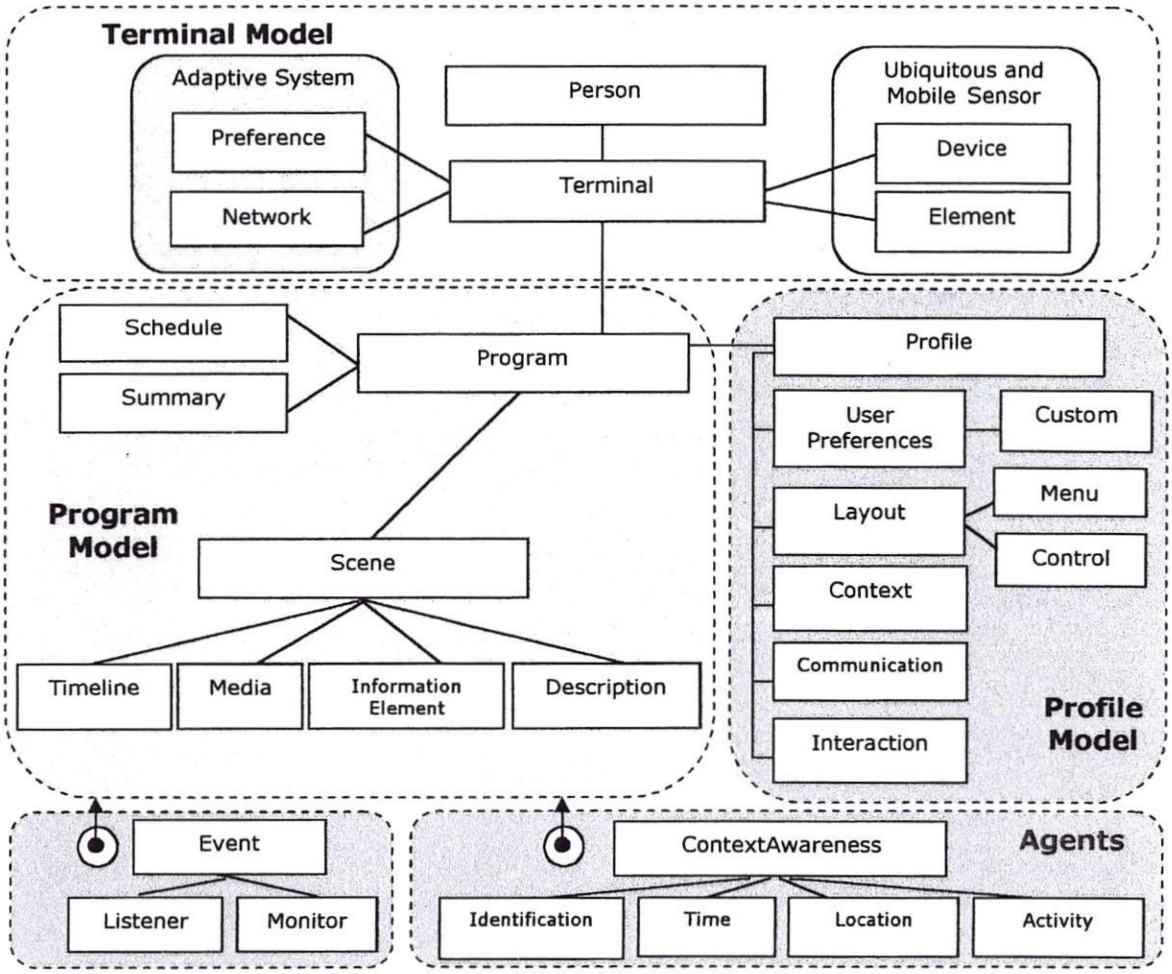

Figure 2: A high-level class modeling for an Interactive TV Application.

3.1 Program Model

The central object of the ITV application is the *program*. The program has connections to the mains parts of the other models (profile and terminal). These connections exist because the terminal receives the information from a program profile but all this information is on the program model, so it communicates with the program object. The *schedule* object archives the information about which times the program will be on the air, this is important to the Digital Video Recorders (DVR) and to the Electronic Program Guides (EPG). Using the schedule is possible to record all the episodes of a program without care, just selecting the program. Another component of this model is the *summary*, which provides summaries of the programs. A scene composes a program also. The scene includes *timeline, media, information element* and *description*. The timeline determines when, how and which each media object will appear on the scene at a specific moment. The media object has information about how to process the media. The description has the metadata about the scene and its element. In this point, information element can be used to provide scene information for the agents involved in a determined negotiation started by the agent communication process, especially in terms of the interactions generated by the user.

3.2 Terminal Model

The central object of this component is the *terminal* that is connected to the *Adaptive System* and to the *Ubiquitous/Mobile Sensor* that includes all the sensors and devices that can provide useful contextual information. The terminal can have embedded devices like CODEC (video and audio, for example) and interaction accessories (e.g. keyboards and joysticks). Other object that is present on the Terminal Model is the *person*. The person is the object where all the relevant information about all the people that use the system is recorded. Examples of relevant information are age, gender, salary/total income, favorite channels/shows, access restrictions, sound volume and image brightness/contrast. The *Adaptive System* refers to network related information. There are two objects composing it, the *network* that has information about bandwidth, type of network, proxies, firewalls and also the dynamic information like actual network status and transfer rate. The *preference* refers to terminal features that can be modified/adequate according to the both network conditions and user's application requirements. The *Ubiquitous/Mobile Sensor* is composed by objects (*devices*) representing the devices that make ubiquity possible like sensors and its characteristics and the *element* that is an object that can directly interact with the terminal. Agents can interact with the terminal using the element object.

3.3 Profile Model

Each program is composed (and presented) by one or more *profiles*. Each profile can be viewed as a different "skin" to the same program. One example is the news program that can have the morning news, lunch news and late news profiles. Each profile has different characteristics and different needs. These characteristics can be mapped into *layout, communication, context* and *interaction*. The layout will have typical interface variables like size, color, elements position and fonts. The *layout* is composed by the *menu* that will indicate the fields /elements in the menu and the type of the menu and by the *control* that informs which controls, like zoom, resize and crop, are available. The *communication* has all the requirements of the profile like bandwidth, CODEC and QoS level. The *context* informs which context information is important to that profile, how this information needs to be processed and what to do with it. The *interaction* object concerns about which types of interaction are available to that profile. Possible interactions can be search related topics on the Web, search related topics on TV, change the angle of a scene, change the audio and remove objects of the scene. The last object of the Profile Model is the *user preferences* where the user can specify his/her preferences related to content. The layout aspects will be customized in the *custom* object.

4 Infrastructure

This section presents aspects covered by the both **NECTAR** and **APICE**. A proposal of an infrastructure for Interactive TV, that enhances interactivity between the user and the whole application, is composed of three components (the *server*, the *client* and the *network*) and is illustrated on the **Figure 3**.

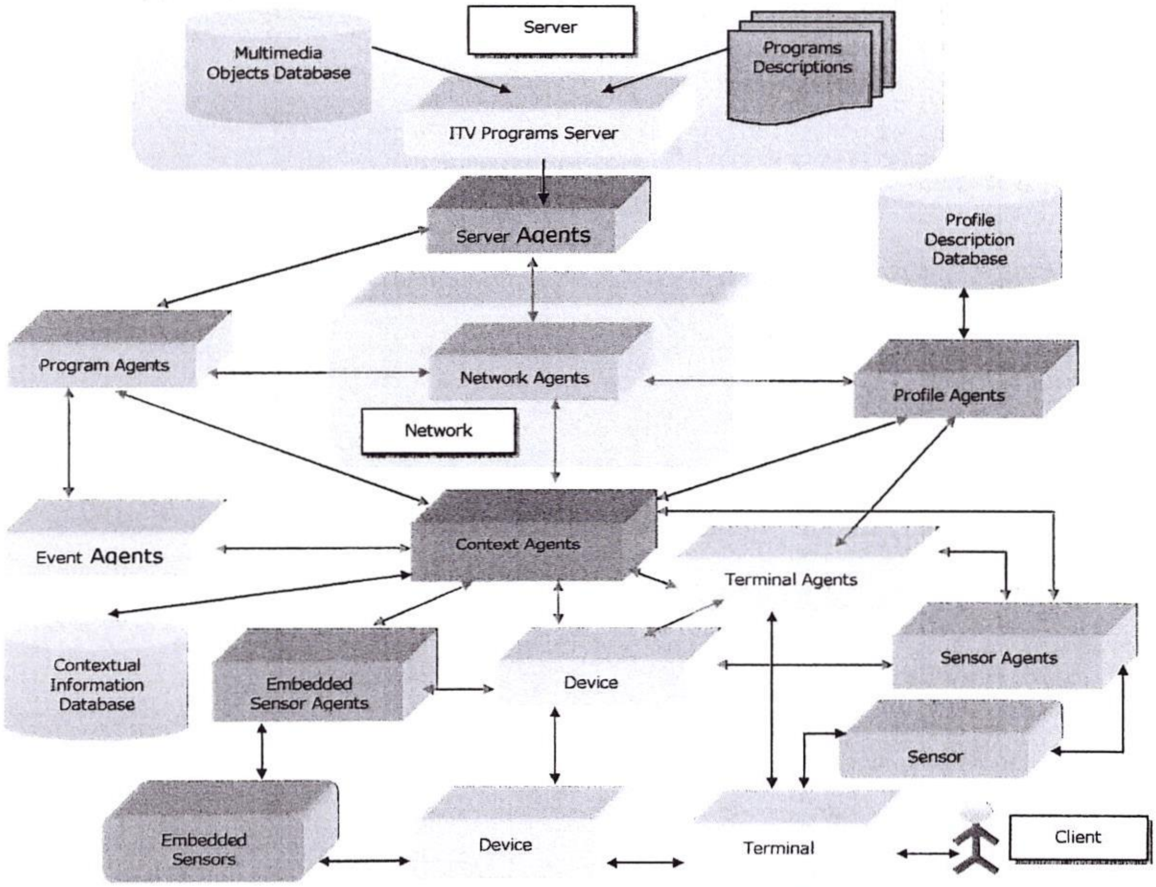

Figure 3: Agent-based architecture for an Interactive TV Application.

The *red arrows* in the **Figure 3** indicate the communication among agents in the proposed infrastructure for ITV applications, while the *black arrows* indicate relationships among hardware and software elements that compose an ITV environment. Details about the agents' communication are described in the **Section 5**.

4.1 The Server

The *server* concerns about how to store and to deliver ITV programs. These programs are composed of multimedia objects like video, audio, images, animations, graphics and texts, among others. Besides that, a program also has a kind of script that specifies the structure of the program and the spatial and temporal relationships of the objects with the program and with other objects. In other words, a program is composed of content and structure. Also the server must be able to deliver different objects to different users according to the users characteristics and needs. The server must store the multimedia objects compounding an ITV program (content) and also store the structure of the program that relates the objects inside the program in time, space and relationships, like showed in **Figure 3**.

MPEG-4 objects can provide the content. The MPEG-4 standard (ISO, 2000; Battista et al., 1999) has been developed to provide solutions to the new multimedia applications. The functionality of this standard is based on objects manipulation.

A MPEG-4 presentation is based on scenes. Each scene is composed by some objects (video, audio, graphics 2D and 3D, animation 2D and 3D, text, and so on) and these objects are coded in elementary streams. The MPEG-4 streams are decoded in a way to allow the separation and the reconstitution of the objects, being possible to interact with the objects in the scene (**Figure 4**). A possible list of interaction can be: changing objects position, scale, rotation, speed (of the objects movement on the screen), add or remove objects that compose the form the scene (ISO, 2000).

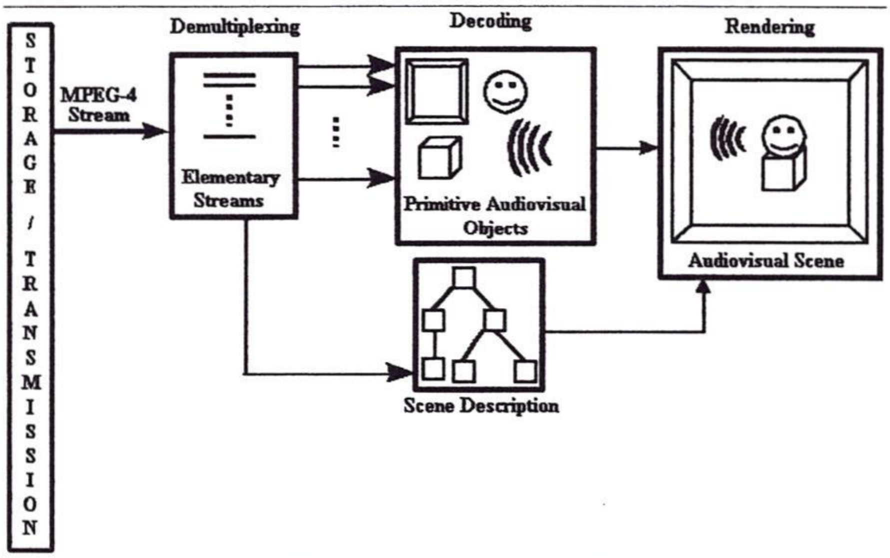

Figure 4: MPEG-4 stream decoding (ISO, 2000).

In addition to coding (there are many types of rates and algorithms to coding in order to cover a large range of applications) the objects are compressed and can be multiplexed in a single stream or in a group of streams to be transmitted or stored.

The MPEG-4 composition process is flexible, allowing the creation of presentations with multimedia content (natural or synthetic) through a graph representing the scene. The media objects are organized hierarchically in a way to allow that complex objects be formed from primitive objects (which are on leafs of the graph).

Each one of these objects is an independent entity, synchronized with the other objects that compose the scene. The synchronization is, in MPEG-4 standard, at the level of individual frames (of the objects in general). It is an advantage when compared with other multimedia standards, like SMIL (W3C, 1998) and HyTime (DeRose and Durand, 1994), which have synchronization at the level of streams or group of frames. This is very important in ITV programs once that the users may insert or remove objects during the scene exhibition.

Other interesting characteristic of the MPEG-4 standard is that MPEG-4 video is composed by layers (Ebrahimi and Horne, 2000). There is a basic layer and some enhancement layers that allow spatial and temporal scalability. The enhancement layers enhance the basic video layer quality in terms of resolution and frame rate. Due characteristics like composition of presentation as structured and related objects, streaming, error resilience, powerful compression, synchronization, among others, the MPEG-4 standard is a strong candidate to implement ITV programs.

The structure for the ITV programs stored in the server can be provided by MPEG-7 (Nack and Lindsay, 1999; ISO, 2000) descriptions, that represents instances of an ITV model.

The MPEG-7 standard aims standardize the descriptions of data with multimedia content. This is done through the specification of a standard set of *Descriptors* (D), which can be used to describe a variety of types of multimedia information. The MPEG-7 standard specifies pre-defined structures of descriptors and its relationships, also specifies the ways to user define his own structures. These structures are called *Descriptors Schemes* (DS).

The definition of a new DS is made using a specific language for this, called *Description Definition Language* (DDL) (ISO, 1999b). Through the DLL, DSs and Ds, the MPEG-7 can do for the multimedia content the same that XML (W3C, 2000) do for the textual content.

The MPEG-4 objects can be described by MPEG-7, providing the ITV content with the necessary semantic to facilitate tasks like content management, searches and increase of the possible interactivity actions.
MPEG-7 is not limited to describe multimedia objects. Its can describe abstractions, like an ITV program:

```
<DSType name="Program">
  <!--core-->
  <attribute name="identificator" type="integer">
  <attribute name="member_of" type="GroupRefType" minOccurs="0"
   maxOccurs="unbounded"/>
  <attribute name="generic_program" type="GroupRefType" minOccurs="0"
   maxOccurs="unbounded"/>
  <attribute name="episode" type="SeriesRefType" minOccurs="0"
   maxOccurs="unbounded"/>

  <!—program basic information -->
  <attribute name="title" type="ProgramTitleType" minOccurs="1"
   maxOccurs="unbounded"/>
  <attribute name="abstract" type="ProgramAbstractType" minOccurs="0"
   maxOccurs="unbounded"/>
  <attribute name="keyword" type="ProgramKeywordType" minOccurs="0"
   maxOccurs="unbounded"/>
  <attribute name="genre" type="ProgramGenreType" minOccurs="0"
   maxOccurs="unbounded"/>
</DSType>
```

This DS shows a way to structuring a program and how it is composed. The composition is determined by identification, association (group, episode) and index information (title, keyword, among others). This DS also establishes a link to other DS through the *member_of* attribute.

The server must adapt its services on demand. These services, for one user or a group of users, can be: stream delivery, data multiplexing, delivery adjusts, program storage and recuperation, searches, among others. The information (parameters), needed to do this adaptation, is described on the client application (actual desired quality level, for example), on the network (traffic, available bandwidth, among others) and on the server (actual load, availability of data, actual delivery rate, among others). The situations where this adaptation must occur are specified on the ITV model, presented by **Figure 1**, especially in terms of the information element present in *Terminal Mode*. In this point, is relevant to observe that the model can be instantiated by MPEG-7 descriptions. Besides that, this information represents contextual information, like identity, location, activity and index information, and can be mapped by MPEG-7 descriptors or descriptions schemes.

Different users can request different levels of quality for the same presentation. Then, the server must be able to deliver different streams of the same program. In the video case approaches based on layered coding are common (Wu et al., 1997; McCanne et al., 1996; Li et al., 1998). The video signal is coded in one or more layers with different priorities. The layer with the higher priority, the base layer, provides the basic level of quality. To sharp incrementally this quality, enhancement layers with priorities progressively lower are delivered. This is exactly what the MPEG-4 standard specifies to multimedia objects (Ebrahimi and Horne, 2000). Then, the server can use MPEG-4 encoders to achieve this functionality.

Software agents are trigged to monitoring the delivery process. These agents access the program descriptions, the user requirements for the presentation and the server parameters. In an example of agents use, consider that the server load is high and the delivery stream average is decreasing, indicating a possible service interruption (or a server overload). In this situation, the agents can consult the program description to find what enhancement layer stream delivery must be stopped, in order to reduce the load in the server and preserve the presentation.

4.2 The Client

A terminal and terminal agents compose the client. The terminal is composed by a personal computer, connected to a network, and must exhibit ITV programs. This terminal must run a player implemented according to the MPEG-4 standard.

The **Figure 5** shows the components involved in the exhibition process of the client MPEG-4 presentation (an ITV program, eventually). The MPEG-4 coded streams are sent to the terminal. Through a demultiplexer, the MPEG-4 elementary streams are separated. Each elementary stream is decoded in objects and sent to the player to realize the rendering. The player uses the scene description and the objects descriptions to put each object in its position, and to verify what kind of control can be applied to each object.

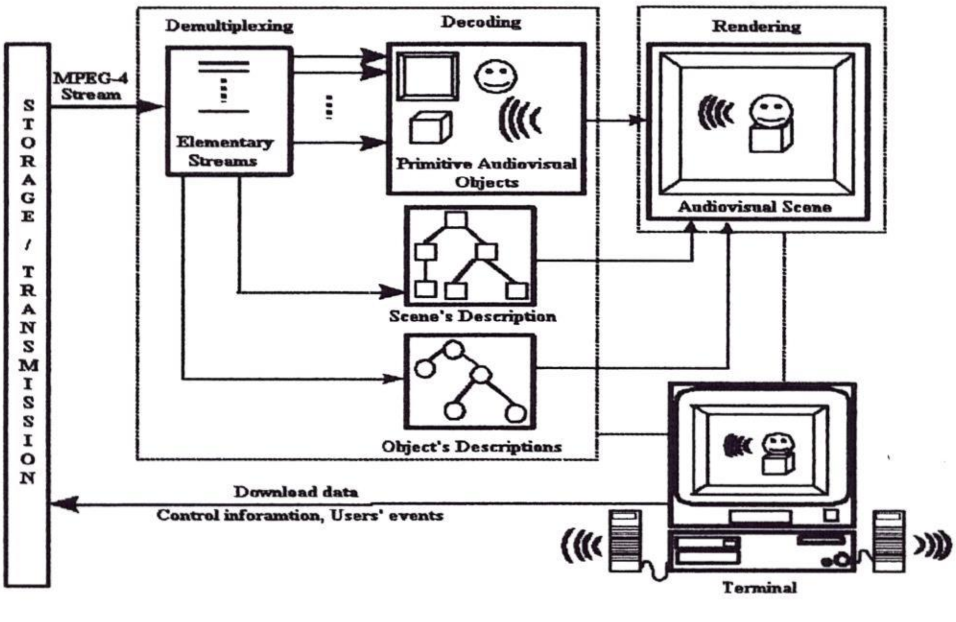

Figure 5: MPEG-4 presentation components at the terminal (ISO, 2000).

The author of the presentation determines these controls (which can be applied to the objects). Each object has a list of standard controls (like shrink or enlarge an image), that can be accessed by the user during the presentation (if the author allows). Complex kinds of controls can be achieved using the MPEG-J API (ISO, 2000; ISO, 1999a). This API acts like an interface between the player and Java programs. In this way, authors can code complex situations of interactivity and sent this code like a MPEG-4 elementary stream, in the same MPEG-4 stream of the presentation.

The users can interact with the presentation in a large range of ways: controlling the objects in the scene, doing searches (with textual, images, sounds, or other object inputs) on the server or on the net, and interact with the network and the server. The user interactions generate events. Some events have a local effect, like enlarging an image, and are easy to process. Others can affect the network or the server services, like a request for a higher rate of transmitted frames. In this case, these events, representing the user wishes or needs, can be sent back to the server or to the network (**Figure 3**).

The terminals can also be connected to devices like VCRs, play stations, cable TV, cameras, and so on. These devices can have sensors, which process some kind of information is order to provide ubiquity and/or context-awareness. The sensor communicates with the devices that are controlled by the terminals.

4.3 The Network

The formulation of this architecture is based in the fact that the network must be aware of the ITV application context (server, client and network actions and interactions), in order to provide a better support to the user-ITV application interaction and to provide new possibilities of infrastructure and application management. These possibilities are related to interactions among the three components of the infrastructure. To do these tasks, the network needs be able to change, dynamically, its protocols and algorithms.

Traditionally, processing within the *network* was limited basically to routing, congestion control and quality of service (QoS) schemes. New network architectures, like active networks, allow dynamic changing of behavior in two ways: routers and switches within the network can perform computation on user data flowing through them, and, users can "program" the network, by supplying their own programs to perform these computations (Tennenhouse and Wetherall, 1996; Calvert et al., 1998).

The proposed architecture aims use active nodes to process contextual information (through the network agents) about network, server and applications actual needs. The contextual information is the descriptions of the ITV program, multimedia objects and streams, and the status of the three infrastructure components. This allows the network to make dynamic decisions (based on these contextual information) about which is the better way to deliver ITV programs, in terms of the traffic, bandwidth, quality of presentation, level of interactivity, programs needs and the user interactions. These decisions can be taken at the moment that relevant events occur, without the user intervention or server response. Finding which contextual information are relevant to the network decisions and unused information, and which decisions really helps the application is an undergoing work.

The client will play MPEG-4 presentations that can be manipulated by Java programs through the MPEG-J API. Some events, triggered by the Java programs at the client, can affect the server (a search, for example), so the server must provide Java support. Therefore, Java is a natural choice to implement active networking, for this project purposes.

There are active networking researches using Java, like the ANTS toolkit (Wetherall et al., 1998), showing that Java is not a bad choice. The ANTS toolkit consists in an API that allows network packets contain byte-coded Java programs. The ANTS classes implement methods to decode and interpret capsules (active network packets), which are responsible to install state and invoke classes installed by other capsules. Because the standard JVM does not support access to transmission resources at low level, ANTS is limited to the basic network capabilities provided by Java.

5 Software Agents for Support to Interactive TV Environment

This section presents aspects covered by **AGITA**. Agents will be used, in the infrastructure of the **Figure 3**, to allow the awareness of context in different situations. There are four context-aware entities defined by Dey (Dey and Abowd, 2000). Therefore four agents will be implemented, one for each entity. In the literature, a computational agent is able to interoperate, via information interchange and services with others programs, to solve complex problems (Moreira and Walczowski, 1997). However, there are several features that increase this basic definition. For example, software agents are components able to communicate and to cooperate with others agents using a communication language (Ketchpel and Genesereth, 1994).

There are several interfaces that allow the establishment of events and actions for an ITV application. These several interfaces can be mapped onto agents-based approach (considering agents features), due the aspects of context-awareness desired for the environment. Specifically, the agent behavior in a context-aware environment can be mapped and presented through a collaboration diagram (UML formalism and graphic syntax), as showed by (Santos Jr. et al., 2001). The relationships between agents characterize the interactions for User-Network-Application approach identified in the **Figure 3**.

In the **Figure 6**, is presented one scheme to represent the interactions between entities in a context-aware ITV application. In the scheme, is possible to observe the existence of well-defined interactions, called *User-Network Interactions*, *Applications-Network Interactions*, *User-Applications Interactions* and *Complete Interactions*.

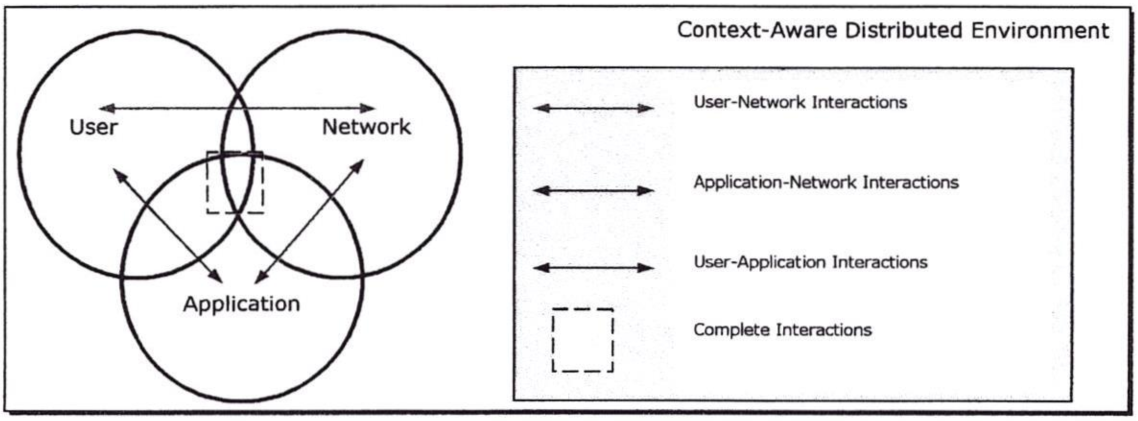

Figure 6: Interactions Scheme in a Context-Aware Interactive TV Application.

Furthermore, agents have its own ideas on how to realize tasks and its own actions schedule (ISO, 2000). Properties like re-activity, adaptation, pro-activity, autonomy, mobility and social capacity (ability to communicate with others agents via an agent communication language) are relative at the environment in that agents are inserted. Thus, this environment is the medium where the agent gets information and can act.

The first one - *User Network Interactions* - represents the interactions between one user (or multi-user) and the native network applications. The second interaction - *Applications-Network Interactions* - covers the relationship between all applications (agents, programs, drivers, daemons, among others) and the network (inclusive the native network applications). The third interaction - *User-Applications Interactions* - represents the interactions between user, via several kinds of interfaces, and all applications of the environment. Completing, we named *Complete Interactions* the well-defined scope of full interactions that happen involving user, network and applications.

There are several interfaces that allow the establishment of events and actions for an ITV application. These several interfaces can be mapped onto agents-based approach (considering agents features related previously), due the aspects of context-awareness desired for the environment, as shows the **Figure 3**. In terms of modeling, several flowcharts can be specified, like class diagram, collaboration diagram, interaction diagram, concurrent objects diagram, events diagram and activities diagram, using UML graphical syntax, for example. Specifically, the agent behavior in a context-aware environment can be mapped and presented through a collaboration diagram (UML formalism and graphic syntax), as showed by **Figure 3**. In this point, the relationships between agents characterize the interactions for User-Network-Application approach identified in the **Figure 6**.

On the other side, the social capacity of computational agents is used for the establishment of cooperation in the resolution of complex problems, like is the case of interactions in context-aware environments.

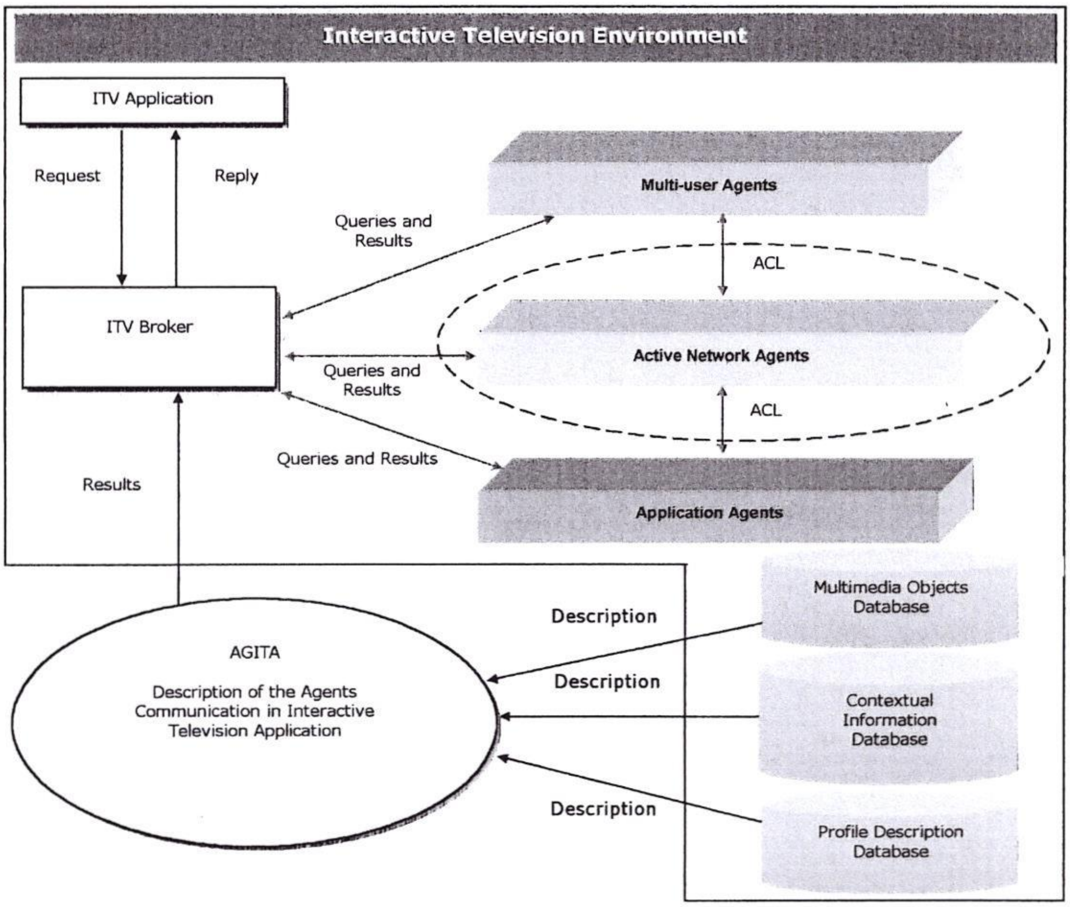

Figure 7: Ontology-based architecture for use of agents in ITV environment.

Considering the treatment offered to an ITV application in this work, especially in terms of the use of technologies like MPEG-7 for scene description, is necessary the definition of ontology for description of the lifecycle of the interactions supported by agents. This ontology, called AGITA (Description of the Agents Communication in Interactive Television Application), is based on FIPA-ACL (Agents Communication Language) (Bellifemine et al., 2001) and description schemes are represented in a class diagram using UML, being relevant for building protocols and applications, mainly to ease both data and document interchange. The **Figure 7** presents a basic architecture for communication between agents to the *User-Network-Applications* approach.

Communication Description Schemes can be defined based on MPEG-7, for example, to represent the agents' features and its actions in a cooperation session for an ITV application. These schemes are part of the *AGITA*, which is represented by UML class diagram, as proposed in (Blásquez, 1998). The Java technology is being both evaluated and tested for the implementation of the *AGITA* through JADE (*Java Agents Development Framework*) resources (Bellifemine et al., 1999).

The **Figure 8** shows a subset of agents that compose the full infrastructure presented in the **Figure 3**. Furthermore, the relationships among *Network Agents*, *Profile Agents* and *Context Agents* are used for to present one subset of the approach built for AGITA. The complete model is being developed in the research scope in which this work is inserted. This complete scope is described in Santos (Santos Jr. et al., 2001).

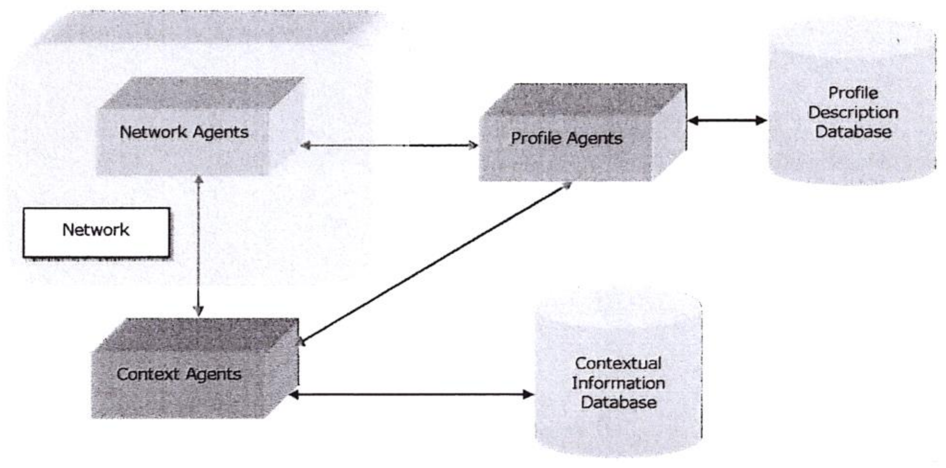

Figure 8: A subset of agents for modeling AGITA.

Philosophically, a meta-language consists of signs that signify something about other signs, but what they signify depends on what relationships those signs have to each other, to the entities they represent, and to the agents who use those signs to communicate with other agents (Sowa, 2000). Furthermore, a resource can be anything that has identity. However, not all resources are network retrievable (Bernes-Lee et al., 1998). These points are relevant for definition of one ontology-based conceptual model for agents' communication, especially for the heterogeneous environments, like those in which ITV applications are inserted.

In the use of software agents, ontology is a specification of one conceptual model, describing objects, concepts and entities of one specific context of the real world. Using this definition is possible to apply it in this working through conceptual model represented by ontology-based architecture showed in the **Figure 7**. According to that figure, the ontology is responsible for to advise the actions of the agents, considering the follows topics:

- The ontology is situated between (middle) *ITV broker* and databases;
- The ontology uses description schemes (based on MPEG-7, for example) for to extract information of the databases;
- The *results* based-extracted on ontology are used as INPUT parameter for agents communication (ACL), through collector mechanism (ITV *broker*);
- The ITV application receives the *reply*. This reply is composed by schemes that describe the actions that the application must to realize, considering, mainly, context-awareness aspects.

The topics above define the "universe" for actuation of the agents, characterizing the communication scope among these agents.

The **Figure 9** shows the UML collaboration diagram of the ontology defined for the agents presented in the **Figure 8** – *network agents*, *context agents* and *profile agents*. The use-case is the follow: a) the *network agents* are stimulated through one request from ITV program server; b) the *network agents* establish communication with *context agents* and *profile agents*; c) with *context agents*, the *network agents* negotiate contextual features that impact the ITV application QoS; d) with *profile agents*, the *network agents* negotiate user profile features that are relevant for adapting of the QoS aspects in the network; e) the *context agents* negotiate with *profile agents* about user profile features that are relevant for context in which the application is running.

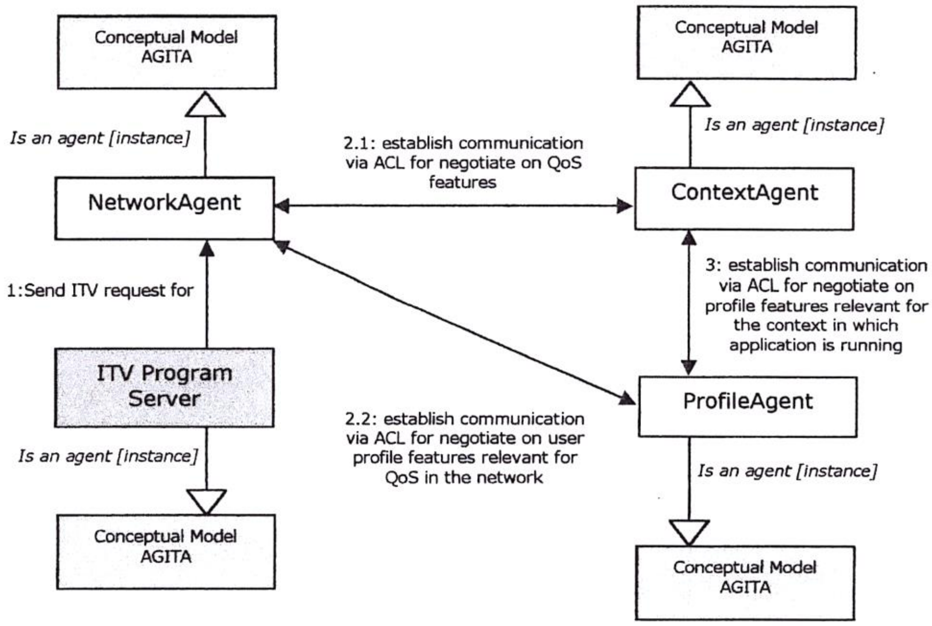

Figure 9: An UML collaboration diagram for representing a portion of the AGITA (Figure 8).

The ontology allows increasing in the relationships among agents, expanding the comprehension about the interactions that are generated from interactions based on *user-network-application* approach, especially in terms of the universe of the complex interactions that exist in the context-awareness ITV environment. The complete model (UML diagram) for AGITA is being developed in the research scope in which this work is inserted (Santos Jr. et al., 2001).

6 Context-Awareness Aspects for Interaction in Interactive Environments

Traditionally, events and relationships are analyzed and modeled for developing of computing environments. However, as the computing is ubiquitous, new methods are need for interface design, involving the both user, network and applications aspects. In further, context-awareness aspects adding new interaction possibilities at the user and also at the application and infrastructure level (Schmidt, 2000).

Context can be viewed as being any information that can be used for characterizing the status of an entity in one specific case. One entity can be one person, one place or one object relevant for any type of interaction between user and application, including user and application itself (Dey, 2000). Therefore, one system is context-aware when it uses the context to provide relevant information and/or services at the user, and this relevance feature depends on user's tasks (Abowd, 1999).

The building of context-aware environments is one complex task and it is necessary to establish parameters for the application. These parameters can be obtained based on five arguments: a) *where* is the user (WHERE); b) *who* is the user (WHO); c) *how* the user works (HOW); d) *when* can it do that (WHEN); e) *what* the user is doing (WHAT).

Considering the automatic task and action execution, it is possible to use the four entities defined by Dey (Dey, 2000): a) location (WHERE); b) identity (WHO); c) activity (WHAT + HOW); d) time (WHEN).

Context-awareness aspects can be relevant when associated to the environment, in which the application is inserted, providing information for adaptability. This is the key-point for defining the *profile* concept in this paper (more details in the **Section 7**). The **Figure 10** shows the schema for mapping context-awareness entities into profile parameters.

The words HOW, WHERE, WHAT, WHO e WHEN denote semantics that can be large complex. In this point, for example, the semantic HOW referring to *"what resources are being used"* expresses, implicitly, *"how the user is using these resources"*. Therefore, implementing one evaluator mechanism (agent + filters, for example) for monitoring and showing a list of allocated resources, can it to get an environment status, indicating *how the user is feeling itself in the environment* (emotional interfaces).

The proposal for joining the semantics HOW and WHAT provides support at creation of the *activity* entity. Later, this entity is used as the key-point for *profile* concepts, once that it is need allocated resources for generating activity at any level of the environment (user, application and infrastructure).

Agents can be used for supporting to automatic tasks and actions execution, observing the entities defined by Dey (Dey, 2000).

In the literature, the basic agent definition denotes that *an agent is able to collaborate, via data interchange, with others agents and programs for solving the complex problems* (Moreira, 1997). Furthermore, an agent is one software component that is able to communicate and cooperate with others agents using one specific communication language.

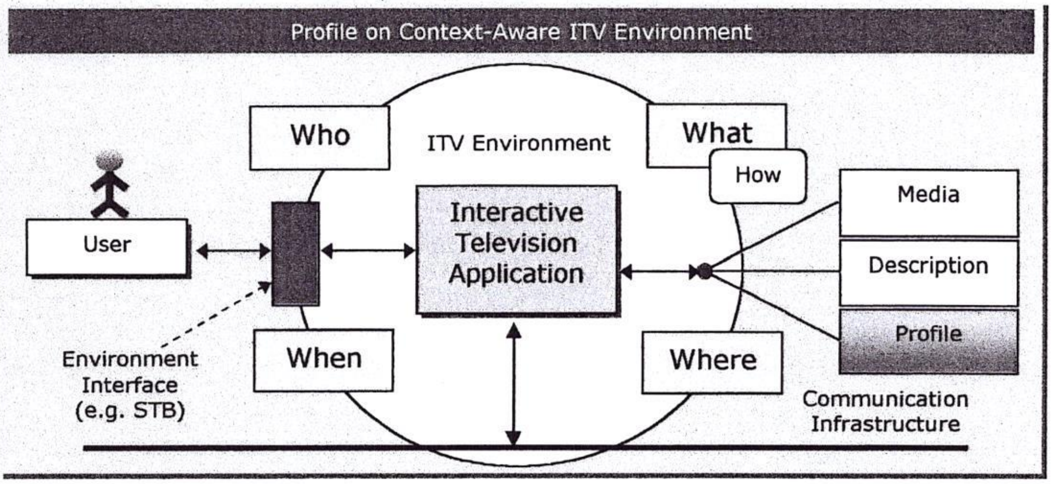

Figure 10: Mapping of context-awareness entities onto profile elements in ITV applications.

In the ITV environments, devices like STB (*Set Top Boxes*) add new interaction possibilities that go beyond the traditional devices like mouse and keyboard. Furthermore, the interaction with the environment in which the application is inserted becomes very important.

The development of *profiles* for ITV applications is a way to define categories that indicate the status of an entity in one application, as well as one way to get information about the behavior of this entity in a timeline(Dey, 2000). Thus, using *profiles* joined to contextual information, an application can make actions and reactions like the follows:

- The presentation of both information and services to the user, like the exhibition of the available media associated to user context, for example;
- The automatic service execution, like the start of events related to current location of an entity, for example;
- The association of the context and information for later recovery, like the storing contextual information during one interactive video session, relating user notes to keywords in one or more scenes.

Context-awareness provides mechanisms for developing adaptive applications and that is important for interactive applications like ITV.

The context-awareness grade of one object quantifies the subjective importance of this object in the environment. This concept also can be applied to a media used in the application. In the ITV application, for example, the context-awareness grade can be mapped onto audio channel volume, image detail level, among others.

An object can control its visibility on relation to other objects. This aspect is relevant in the access to media content of a scene. MPEG-4 technology offers support for this feature, for example.

7 A proposal of structure for Interactive Television Application

This section presents aspects covered by **PRACTIC**. Actually, the ITV models present some aspects for adding interactivity at the traditional TV programs (ATV, 2000; DVB, 2000; TVA, 2001a).

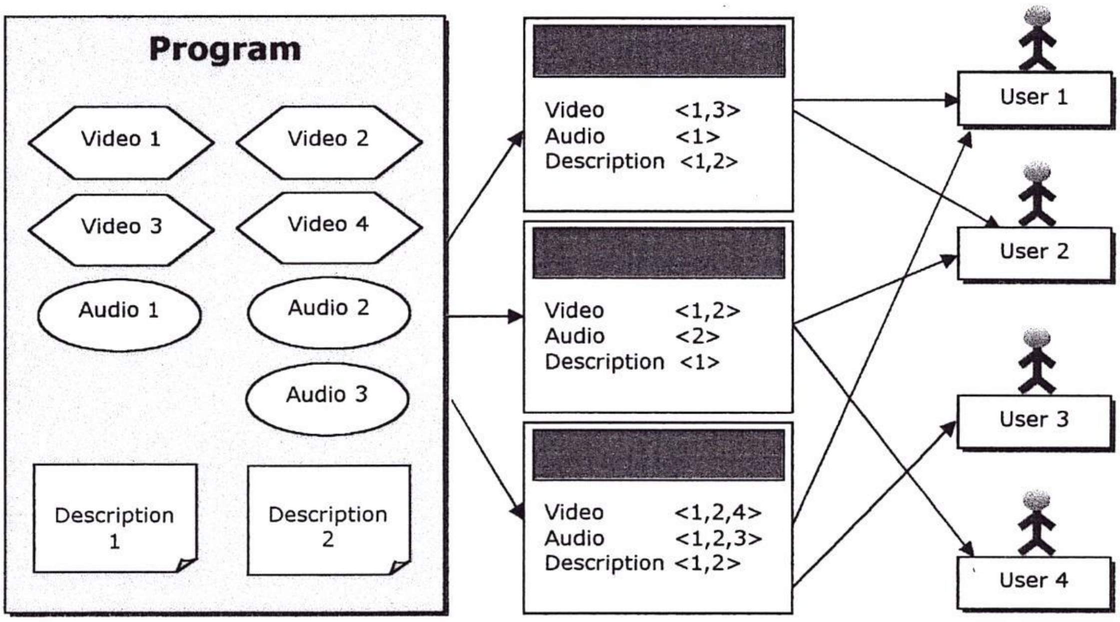

Figure 11: Structure proposed for an ITV program.

However, using the same technology used for these models is possible to reach high interactivity levels, offering bonus to the user. At this point, it is possible to adapt one program, making few modifications, to the users that have different interests, as showed in the **Figure 11**.

The **Figure 11** presents one program and its objects, but for each user of this program shall be presented in a different way, depending on the profiles offered for each user. In the **Section 7** the profile concept is presented, and one example of the use these concepts in ITV programs.

This kind of adaptation, for one particular ITV program, can be reached using context-awareness aspects, as presented in the **Section 6**. The **Sections 7.1** and **7.2** present both program description and profile description.

7.1 The program

The program is the main element of the ITV. One program can be defined as the full set composed by media (and its descriptions), scenes (and its descriptions), a summary (or synopsis) and its programming, as showed by **Figure 12**.

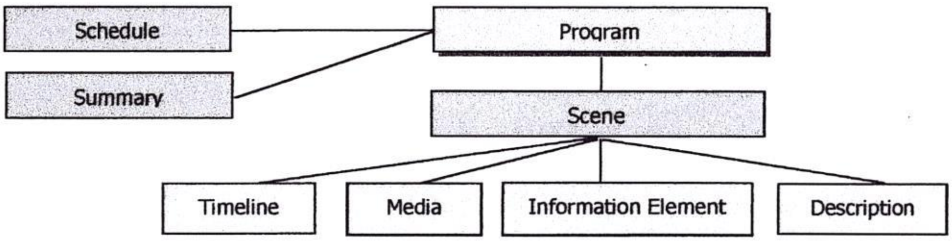

Figure 12: Entities that compose the program in ITV application.

The program is structured using the DDL MPEG-7 (ISO, 2001a) and uses the multimedia descriptions defined by MPEG-7 (ISO, 2000). The entities *Schedule* and *Summary* are defined by TV-AnyTime Forum (TVA, 2001b; TVA, 2001c). Next, a simplified version of the Scene Description Schema is presented. The complete version for scenes (and programs) is found at (PRACTIC, 2001b).

```
<element name="Scene">
  <complexType complexContext="true">
    <element name="Media">
      <complexType complexContext="true">
        <attribute name="type" type="mpeg7:DSType"/>
        <attribute name="id" type="string"/>
        <attribute name="filelocation" type="mpeg7:DSType"/>
      </complexType>
    </element>
  </ComplexType>
</element>
```

Using the media and its descriptions is possible to build different interfaces, or presentation modes, for one program; for this possibility, we have named *profile*.

For coding of the media and scenes, this proposal considers the use of the MPEG-4 technology (one flexible and efficient technology for audio and video compressing).

MPEG-4 has fundamental features, like the storing of the components of a scene as objects, allowing increasing in the dealing of these components. This feature is singular for defining and implementing profiles in ITV programs. However, it is important to observe that the structure proposed by this work is independent of the transport mechanism, being possible the use of the Digital TV standards (ATSC, 2001; DVB, 1999; ARIB, 1998), as well as the Multimedia Home Platform (DVB, 2000) for presenting content.

Both scene and media descriptions allow the creation of the search tools for audio, video and scenes itself. Due to the fact that this work uses the MPEG-7 for describing scenes and media, MPEG-7 search tools can be used.

In other hand, the *summary* and *schedule* have special importance for EPG (*Eletronic Program Guides*) applications, for example. Using summary information, the user can choose exactly the programs for watching/recording; using the schedule information, the user doesn't worry with the office hour for starting of the program, once its STB will make it on transparently. Another relevant user for *schedule* is on ubiquitous computing: the STB can inform to telephone equipment or PDA (*Personal Digital Assistant*) about the starting hour for one determined live program.

7.2 The profile

One program needs presentation mode specification. Due to the fact to have large types of available media, there are very large presentations modes also. The source that generates the programs is responsible for providing profiles to the user, but the user can customize these profiles. Thus, is not allowed the incompatibility between profiles and programs, and different profiles become it possible for different user categories (for example, depending on user channel planning).

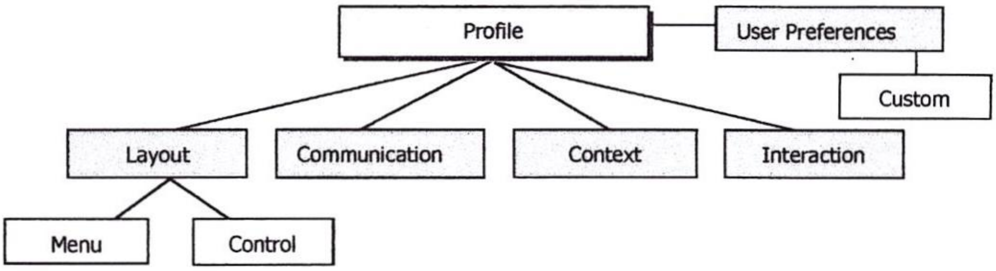

Figure 13: Entities that compose the program in ITV application.

For profile description, this work proposes using MPEG-7 standard and the creation of new elements [available on (PRACTIC, 2001b)], for describing some *layout* aspects, requirements and context, relevant for this profile and its interactions, as shows the **Figure 13**. The information on profile is stored on element <profile>. This element is composed by other elements, which will be described to follow.

The defined presentation aspects, which are part of the element <Layout>, are relatives on background colors, font colors and media positions. Inside still element *layout*, there are the available media controls for one specific profile, the kind of menu and its content. One simplified version of the Description Schema of the element <layout> is showed the follow. The complete version of the schema is found at (PRACTIC, 2001b).

```
<element name="Layout">
  <complexType complexContext="true">
    <attribute name="bgcolor" type="string"/>
    <element name="Object">
      <complexType complexContext="true">
        <attribute name="id" type="string"/>
        <element name="ObjectPosition">
        </element>
        <element name="ObjectSound">
        </element>
      </complexType>
    </element>
    <element name="Menu">
    </element>
    <element name="Control">
    </element>
  </complexType>
</element>
```

The profile requirements are both specified and stored inside to element <Communication>, where are the communication requirements of each media, the needed decoders and STB requirements. One simplified version of the Description Schema of the element <communication> is showed the follow.

```
<element name="Communication">
  <complexType complexContext="true">
    <element name="CommRequirements">
      <complexType complexContext="true">
        <attribute name="medianame" type="string"/>
        <element name="Bandwith">
          <simpleType>
            <restriction base="nonNegativeInteger">
              <minInclusive value="0"/>
            </restriction>
          </simpleType>
        </element>
        <element name="Codec" type="string" />
      </complexType>
    </e      lement>
  </complexType>
</element>
```

Inside element <Context> is inserted all kind of contextual information for one specified profile. This element has been defined from context-awareness parameters and variables, like <who>, <where>, <when> e <what> (Dey, 2000), and presented in the **Section 6**. This element contains the actions that should be realized in determined situations. The follow example shows the use of schema for context-awareness.

```
<Context>
  <who ="mr. 1"> action 1</who>
  <who ="mr. 2">
    <where ="bedroom">
      <when ="24:00">
        openProgram(news1)
      </when>
        openProgram(news2)
    </where>
     openProgram(music1)
  </who>
  <what ="24:00"> turnoff() </what>
</Context>
```

The last big element is the <Interaction>, in which will be all actions that should be realized for each interaction defined for one profile. These interactions are described of the following way:

```
<element name="Interaction">
  <complexType complexContext="true">
    <attribute name="on" type="string"/>
    <attribute name="media" type="pratic:Media"/>
    <attribute name="button" type="pratic:Button"/>
    <attribute name="action" type="string"/>
  </complexType>
</element>
```

In the follow example, the attribute *move* is a MPEG-7 description, the attribute *video3* has been presented in the program and described in MPEG-7, and the action *moveMedia(video3)* represents one command that will be trigged by STB for realizing the task. This command depends on Operating System inserted on STB and of the applications existents, like MHP or JavaTV (Sun, 2000).

```
<Interaction>
  <on ="move" media="video3" action="moveMedia(video3)">
  <on ="click" button="volume" action="openVolumeBox()">
</Interaction>
```

Besides these elements, there is one more element for profiles called <UserPreferences>. This element is stored in the STB and contains some customizations that have been made by the users for one determined profile of the one specific program. The follow example shows the use of this element.

```
<UserPreferences user="Galli">
  <MediaVolume grade="25" tranformation = "ultrabass" />
  <profile="Sports3" program="NewSports">
    <Layout bgcolor="ccffcc" />
  </profile>
</UserPreferences>
```

Using these elements described here and MPEG-7 Descriptions (ISO, 2000) and TV-Anytime Forum Descriptions (TVA, 20001b; TVA, 2001c), it is possible to specify several sets of singular features, besides that personalization for profiles and programs.

7.3 Using profiles in interactive television programs

Initially, the utilization of profiles allows more adaptability for a specified program at the user. The **Figure 14** shows one sequence diagram, based-on UML syntax (Booch, 1996), which illustrates the basic action schema of the profiles for presentation of the program in non context-aware environments.

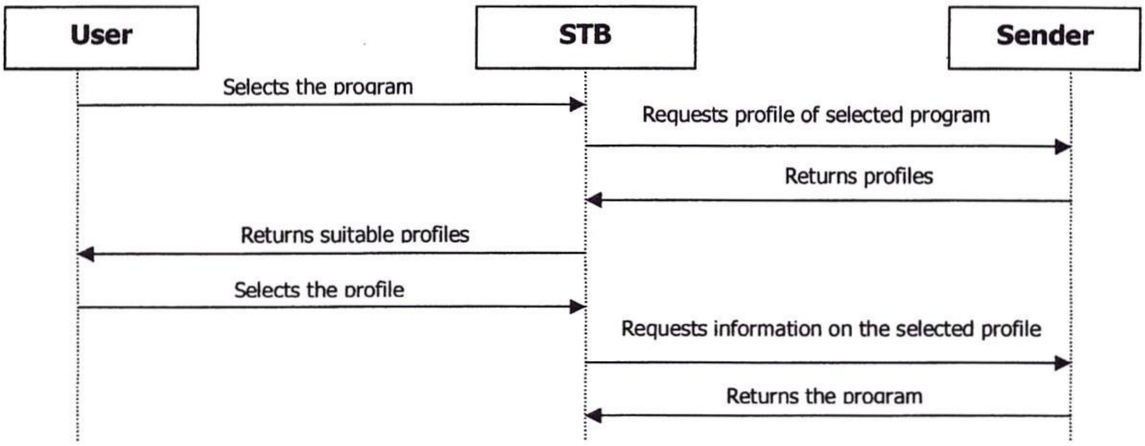

Figure 14: UML Sequence Diagram for ITV program entities.

The sequence of events showed by **Figure 14** starts when the user (**user**) chooses what program he wants to watch; chosen the program, the *STB* (**STB**) requests to the *source* (**sender**) the available and possible profiles for the *user;* the *source*, then, returns these profiles to the *STB* that shows the options to the *user;* then the *user* chooses the profile and its modifications, and the *STB* proceeds the request to the *source*, asking for video *streams*, audio, metadata and scene control *stream;* finally, the *source* will send the program and the profile to the *user*.

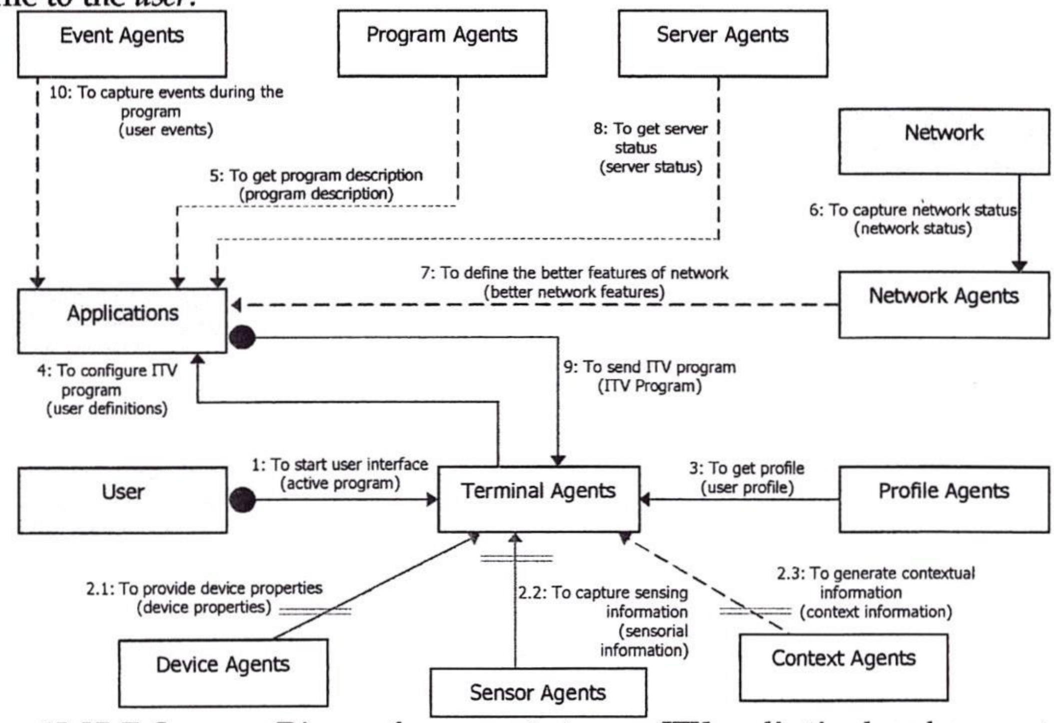

Figure 15: UML Sequence Diagram for one context-aware ITV application based-on agents.

As described, one other use for the profile concept is its use on context-aware environments. In this kind of environment, agents provide information to the STB, as described in the **Section 6**. The STB uses these information (described in the element <Context>) for profile adaptability. The **Figure 15** presents one collaboration diagram for this case.

In the next section, one interactive program that uses profile will be presented.

7.4 Applying profiles to interactive television program

In this section, a proposal for an instance of one interactive television program - *sports event* program - is presented. This program has a soccer game as content, for example, and represents the application of the profiles concepts in one interactive program.

The proposed situation for this program allows the visualization of the possibilities for using profiles and adding new interacting ways, showing that the both user and application are able to interact with the infrastructure described in the **Section 2**.

In the presentation of one sport program in ITV paradigm, some features are expected for user interaction, among that: a) possibilities for different camera view angles; b) possibilities for several audio options, like different speakers and languages; c) access to statistics information on teams, game itself and players; d) possibilities for *replay* situations besides that conventional mode; e) possibilities for providing information about related topics and subjects, like another simultaneous games, for example; f) access to information about the program (summary) and the programming; g) possibilities for access to related programs, like games of the current team and monthly programming, for example.

In terms of the development of the application, these features should be implemented and described from the program elements and profiles presented in the **Section 7**. Each profile can have singular features that must be specified. In this point, some examples of variables of the one profile can be as the following:

1. *Layout*: aspects like background color and images color; buttons style; menus style; menus position; media position in the user interface (if allowed); that media should be presented; that media can have its position changed;

2. *Communication*: that codes are need for each media, as well as that bandwidth bit rate are suitable for each media;
3. *Context*: that decision commands use context variables (like *who* is using; *what* is the use time; *what* the another connected devices; *what* already it seems at determined time;
4. *Interaction*: that changes are allowed in relation to media and *layout*, for example, from user interaction with groups of buttons/controls/menus associated to these media and *layout*.

The definition of one instance for one program, as the case of *sports event*, not is a fact to restrict the application of the profile concepts to ITV programs. Both the infrastructure and profiles can be applied, for example, in other interactive environments, like videoconference in teaching-learning applications. In (PRACTIC, 2001a) is found the description (MPEG-7) for this kind of program and its different profiles.

8 Related Works

The interactivity on television is not new. TV shows that ask for the participation of the spectators using the telephone, like games, are very common. What has changed is the level of the interactivity and how to interact with the programs. Nowadays there are several new functions on the TV using the ITV systems on the market. Is possible to navigate on the WWW, the record programs automatic (using PVRs), make banking transactions, change camera angle, receive Video-on-Demand and play games. Each of these new functions is present in one or more ITV solutions. The ITV solutions that integrate TV content with WWW content are called Enhanced TV. Enhanced TV systems, as the name says, enhance the actual TV and use the analogical TV signals (transmitting the necessary information and the triggers throughout the VBI line 21) and the telephone line to access the Internet. But as the solutions are not compatible a specification was created to standardize them (ATVEF, 2000). The Advanced Television Enhancement Forum (ATVEF) has built this specification.

But ITV isn't just what ATVEF plans to standardize, in the future, when the TV signals will be digital and the TV (or a Set Top Box - STB) has a fast Internet access, new interactions can exist, so there is a need to model it. The goal of the TV-Anytime Forum (TVA, 2000) is a new TV with new functions and new ways to interact with it. It's much more complex and complete than the ATVEF standard. It has specifications on how the hardware must be and work, and has a document specifying how use MPEG-7 to provide metadata (TVA, 2001).

However, none of those standards tries to model interactions of the system with the environment, and also don't use context-awareness, which are important in the development of ubiquitous systems. With this type of interaction, some remote controls functions can disappear, and new types of interaction can arise (or became much easier to do).

This work intents use context-awareness in the building of the ITV infrastructure. The building of context-aware applications is the target of the Context Toolkit (Salber et al., 1999). It provides abstractions that relieve developers from the tasks of sense and access context information, allowing the concentration of efforts on how to use context. These abstractions were built considering the user-application communication, leaving an open question about the application-infrastructure communication.

9 A Scenario Example for SCOTIMM purposes

A situation where the proposed modeling and infrastructure can be applied is as a businessperson entering in his home. This simple fact makes his TV interact with him providing information, choices and remainders. The user and TV interact to each other and the events generated by this interaction can affect the application on the client, server or network, as described below.

Mr. Galli is a high executive in an Internet company. As he works until almost 8:00 PM everyday he looses some of his favorites shows. He arrives at home and finds his son, Galli Jr., 8, watching Japanese cartoons. Mr. Galli, as a good father, asks Jr. if his homework is done and because of the negative answer tells him to study. As soon as Galli Jr. leaves the room, the TV shows to Mr. Galli which TV programs were recorded and which may be interesting to him now, and remembers Mr. Galli that he has new messages (e-mails).

Mr. Galli chooses a boxing fight. During the show a friend of him call in a videoconference to share comments about the fight. Mr. Galli sees and listens to the fight, but only listens to his friend. In the most exciting part of the fight, Galli Jr. returns to the room with his homework done. The TV immediately changes to the images of the Mr. Galli's friend in the videoconference (Mr. Galli doesn't want that Jr. watch violence on TV). As Jr. wants to watch cartoons again and Mr. Galli doesn't like cartoons, he decides to eat in the kitchen. While he eats he reads the e-mails that the TV sent to his PDA when he entered in TV room.

As Mr. Galli works at the time that his favorite shows are presented, he programmed the terminal to record them. The information about which shows need to be stored is located on the *Person* Object on the *Terminal Model*.

When Mr. Galli enters in the TV room, a system based on image recognition (could be other recognition technique) identifies him, through the sensor agents, and send this information to the terminal agents. Then Galli Jr. leaves the room. The system (sensor agents and context agents) detects that Galli Jr. leaves the room but his father is still there. The terminal agents consult the context agents to know what action should be taken, once that a new user configuration happens. The context agents indicate that the Jr. profile must be replaced by the Mr. Galli profile, stored on the profile description database.

The terminal loads Mr. Galli profile and scans his preferences. Based on information about time, day of week, programs being exhibited (news, series, according to his preferences) and programs recorded, the terminal shows a list of choices to Mr. Galli. The information to compose this list is obtained from the profile and contextual databases.

As Mr. Galli chooses to watch a show and his PDA is on his pocket, the context agents "understand" that Mr. Galli doesn't want to read messages now, but he may want to do this later, and activates the device responsible to uploads Mr. Galli's messages to his PDA.

When Mr. Galli's friend launches the videoconference, his systems sent through the network his images and sounds to be presented on the Mr. Galli's terminal. When the first data arrive at the Mr. Galli terminal, the network agents interact with the context and terminal agents and find that this terminal isn't powerful enough to exhibit the fight show and the videoconference at same time. The context agents consult the descriptions of these programs, evaluate priorities, and tell to the network agents that the videoconference images are less important that the videoconference sound. Then, the network decides interrupt the videoconference images delivery. In this way, the Mr. Galli's terminal is able to exhibit the show and provide a voice-conference.

Later, when Jr. enters the TV room again, while Mr. Galli is watching his fight show (in the most violent part), the sensors identify Jr. and notify the terminal and the context agents. The context agents find that Jr. is not allowed to watch violent shows and notify the terminal to stop the show exhibition. The terminal starts to record the show for later watching. The "rules" about what kind of shows the people can or cannot watch were programmed previously by Mr. Galli and stored at profile and contextual databases. As the show was interrupted, the context agents process this information, find that a voice conference is happening, that images of this conference can be presented now, and, that Jr. is allowed to see this. The network agents detect this change in the context and the images are presented.

As Jr. finished his homework, wants to watch cartoons again and Mr. Galli doesn't like cartoons, Mr. Galli decides to eat in the kitchen. Again, the system finds that a change of users occurs and starts to exhibit the cartoons that match with Jr.'s pre-defined preferences. Then, Mr. Galli remembers that he doesn't read his e-mails yet and, access his PDA to do this while he looks for something in the refrigerator.

10 Final Remarks

There are some efforts to specify an ITV model. The ATVEF model was specified to improve actual television increasing some functions (and the interactions) (Abowd, 1999). Its goal was standardize actual ITV systems (like WebTV, Liberate and OpenTV). The TV-Anytime Forum's goal is a new TV with new functions and new ways to interact with it (ISO, 1999a). It's much more complex and complete than the ATVEF standard, it describes how MPEG-4 and MPEG-7 can be use. They are not rivals because they have different goals and different timelines. But none of those approaches presents a modeling of interactions of the system with the environment inserted in a context-aware and ubiquitous computing. With this type of interaction, some remote control's functions can disappear, and new types of interaction can arise (or became much more easy to do). Add context-awareness in the network is an area weakly researched, being a novelty in ITV applications. The active networks has resources to support adaptive and aware systems, allowing that the network adapt itself to environment events. This feature is relevant for context-awareness aspects, impacting the interactions of the system as whole, allowing new types of interactions and management of the system.

The proposed modeling helps to determine and to formalize new types of interactions generated by a context-aware ITV application. Towards to future TV, this modeling can be used to ease the development of environments, due to the fact complete interactions (generated by *user, network* and *applications* entities) are represented in the approach. These interactions are very important to context-aware, wearable, mobile and ubiquitous computing.

References

Abowd, G.D., 1999: Software Engineering Issues for Ubiquitous Computing; International Conference on Software Engineering (ICSE'99), pp.75-84.

ARIB, 1998: Association of Radio Industries and Businesses: Terrestrial Integrated Services Digital Broadcast - ISDB-T: Specification of Channel Coding, Framing Structure and Modulation (ISDB-T).

ATSC, 2001: Advanced Television Systems Committee: ATSC DIGITAL TELEVISION STANDARD (Revision A) [2001-April] http://www.atsc.org/Standards/A53/.

ATVEF, 2000: Advanced Television Enhancement Forum: Enhanced Content Specification http://www.atvef.com/library/spec1_1a.html.

Battista, S. et al., 1999: MPEG-4: A Multimedia Standard for the Third Millennium, part 1, IEEE MultiMedia 6(4) 74–83.

Bellifemine, F. et al, 2001: Developing multi agent systems with a FIPA-compliant agent framework. In: Software - Practice And Experience, no. 31, pp.03-128.

Bellifemine, F. et al., 1999: JADE A FIPA-compliant agent framework. Proceedings of PAAM'99. London, pp.97-108.

Berners-Lee, T. et al., 1998: Uniform Resource Identifiers (URI): Generic Syntax. Internet RFC 2396.

Blásquez, M., 1998: Building Ontologies at the Knowledge Level using Ontology Design Environment, KAW'98, Canada.

Booch, G. et al., 1996: UML: Unified Modelling Language for Object-Oriented Development, Documentation set. Addendun, Santa Clara, California, Rational Software Corp.

Calvert, K.L. et al, 1998: Directions in Active Networks, IEEE Communications Magazine 36(10), pp.72-78.

DeRose, S. and Durand, D., 1994: Making Hypermedia Work - A User's Guide to Hy-Time, Kluwer Academic Publishers.

Dey, A.K. and Abowd G.D., 2000: Towards a Better Understanding of Context and Context-Awareness; CHI 2000 - Workshop on The What, Who, Where, When, and How of Context-Awareness.

DVB, 1999: Digital Video Broadcasting (DVB) - Framing structure, channel coding and modulation for digital terrestrial television versão1.2.1 (07/99) [On line] http://www.etsi.org/.

DVB, 2000: Digital Video Broadcasting (DVB) – Multimedia Home Platform (MHP) Specification 1.1 (Draft) [On Line] http://www.mhp.org/.

Ebrahimi, T. and Horne, C., 2000: MPEG-4 Natural Video Coding - an Overview, Signal Processing, Image Communication 15(4-5), pp.365–385.

ISO, 1999a: ISO/MPEG N3201, DDL Working Draft 1.0, MPEG Systems sub-group, ISO, Maui, December. http://www.cselt.it/mpeg/public/mpeg-7_ddl.zip.

ISO, 1999b: ISO/MPEG N3158, Overview of the MPEG-7 Standard, MPEG Requirements Group, ISO, Maui, December. http://www.cselt.it/mpeg/standards/mpeg-7/mpeg-7.htm.

ISO, 2000: ISO/MPEG N3445 MPEG-7 overview (version 3.0)., http://www.cselt.it/mpeg/standards/mpeg-7/mpeg-7.htm.

ISO, 2001a: ISO/MPEG N3536, MPEG-4 Overview (v.15 – beijing version): http://www.cselt.it/mpeg/standards/mpeg-4/mpeg-4.htm.

Jennings, N. R. and Wooldrige, M., 1995: Agents Theories, Architectures and Languages: a Survey, Intelligent Agents, pp.55-67.

Ketchpel, S. and Genesereth, M, 1994: Software Agents; Communications of the ACM, pp.48-53.

Ketchpel, S. and Genesereth, M., 1994: Software Agents, Communications of the ACM, pp.48-53.

Li, X. et al., 1998: Layered Video Multicast with Retransmission (LVMR): Evaluation of hierarchical rate control, IEEE INFOCOM'98, pp.1062–1072.

McCanne, S. et al., 1996, 1996: Receiver-driven layered multicast, SIGCOMM'96.

Moreira, D. A. and Walczowski, L. T., 1997: Using Software Agents to Generate VLSI Layouts. IEEE Expert Intelligent Systems, pp. 26-32.

Nack, F. and Lindsay, A. T., 1999; Everything You Wanted to Know About MPEG-7: Part 1, IEEE Multimedia, pp.65 – 77.

Pascoe, J. et al., 1999: Issues in Developing Context-Aware Computing. In Gellersen, H. (ed.): Handheld and Ubiquitous Computing. Lecture Notes in Computer Science, Vol. 1707. Springer-Verlag, Berlin Heindelberg New York, pp.208-221.

PRACTIC 2001a: PRACTIC Program Example [On line] http://chapinha.intermidia.icmc.sc.usp.br/SCOTIMM/PRACTIC/practic-e.xsd.

PRACTIC 2001b: PRACTIC Schema [On line] http://chapinha.intermidia.icmc.sc.usp.br/SCOTIMM/PRACTIC/practic-s.xsd.

Salber, D. et al, 1999: The Context Toolkit: Aiding the Development of context-Enabled Applications. In proceedings of CHI'99, pp. 434-441.

Santos Jr., J.B. et al., 2001: Modeling of User Interaction in Context-Aware Interactive Television Application on Distributed Environments. Proceedings of the 1st Workshop on Personalization in Future TV. Sonthofen, Germany, July-2001. http://www.icmc.sc.usp.br/~joao/futuretv/JRGEPaperUM2001.html.

Schmidt, A., 2000: Implicit Human Computer Interaction Through Context. Personal Technologies Volume 4(2&3) June, pp.191-199.

Sowa, J. F., 2000: Ontology, metadata and semiotics. Springer-Verlag, Berlin, pp.55-81.

Sun, 2000: Sun Microsystems. Java TV API 1.0 specification (November 2000) [On Line] http://java.sun.com/products/javatv/

Tennenhouse, D. L. and Wetherall, D. J., 1996; Towards an Active Network Architecture, in Multimedia Computing and Networking (MMCN '96), San Jose, CA.

TVA, 2000: TV-Anytime Forum: Specification Series: S2: System Description (Informative) - October. ftp://tva:tva@ftp.bbc.co.uk/pub.

TVA, 2000: TV-Anytime Forum - Specification Series: S2: System Description (Informative) - October. ftp://tva:tva@ftp.bbc.co.uk/pub.

Wetherall, J. et al, 1998: ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols. Proc. IEEE OpenArch'98, San Francisco, CA.

Wu, L. et al., 1997: Thinstreams: An Architecture for Multicasting Layered Video, NOSSDAV'97.