Modeling of User Interaction in Context-Aware Interactive Television Application on Distributed Environments [1]

João B. dos Santos Jr, Rudinei Goularte, Gustavo B. Faria and Edson dos S. Moreira
{joao, rudinei, gusfaria, edson}@icmc.sc.usp.br
Institute of Mathematics and Computing - ICMC
University of Sao Paulo - USP
13566-570, São Carlos-SP, Brazil, PO Box 668

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 paper presents 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 paper 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.

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 [16][1].

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) [8]. 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

[1] The ideas developed in this paper result from projects funded by FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico)

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.

This paper speculates on how to model the interactions (both user, network and application interactions) and is structured in four sections. In the next section, the needed infrastructure for context-aware ITV is presented. In the third section, considering the infrastructure presented in the second section, a proposal for modeling a context-aware ITV application is discussed. In the last section, final remarks are discussed.

2 Interactive TV Infrastructure Background

According to the categories described in the previous section, we propose an infrastructure for Interactive TV that enhances interactivity between the user and the whole application. This infrastructure is composed of three components: the *server*, the *network* and the *terminal*. The *server* concerns about how to store and to deliver the multimedia information. Also the server is 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 and the structure of the program that relates the objects inside the program in time, space and relationships.

The content can be provided by MPEG-4 objects [3]. The structure can be represented by MPEG-7 descriptions [15][9], which are instances of an ITV model. In this way, the MPEG-7 descriptions give to the content (MPEG-4 objects) the necessary semantic to increase the interactivity and facilitate how to get contextual information about the program to adapt the server services [10].

The second component is the *network*. It needs to be flexible enough to adapt to the traffic conditions while keeping the presentation quality manageable. Traditionally, processing within the *network* was limited basically to routing, congestion control and quality of service (QoS) schemes. Active networks represent a new approach to network architecture. These networks are "active" 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 [17]. 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[2]. The MPEG-4 standard provides an API called MPEG-J that can be used like an interface between MPEG-4 player and Java programs [11]. 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 Proposal for Modeling Context-Aware ITV Applications

Context-awareness aspects are very important when associated with the environment in which the application is inserted. These aspects refer it to operations associated to contextual sensing, contextual adaptation and contextual resources discovered that, in general, control the presentation of information and services to the user and the automatic execution of environment actions. The Figure 1 shows such a model. It is composed of three different components. The first one is called *The Terminal Model*. There are two objects that are more important; the *Terminal* that will run all the programs and receive all the interactions. It will also communicate with the network (choosing the needed configuration) and with the sensors (that will provide information for the ubiquitous computing). The other relevant object of this

---

[2] The ISO/IEC releases a free version of a player at http://isotc.iso.ch/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm/, see 14496-5

component is the *Person*. The *Person* will trigger the interactions and the time of its occurrence.

The second component is called the *Profile Model*; this part will have the information about the programs and the user preferences for that program. Each *Program* can have many different *Profiles*. *Profiles* can differ in types of interaction, available media, aspects of layout, customization and privileges. All possible interactions of those profiles will be specified on this model.

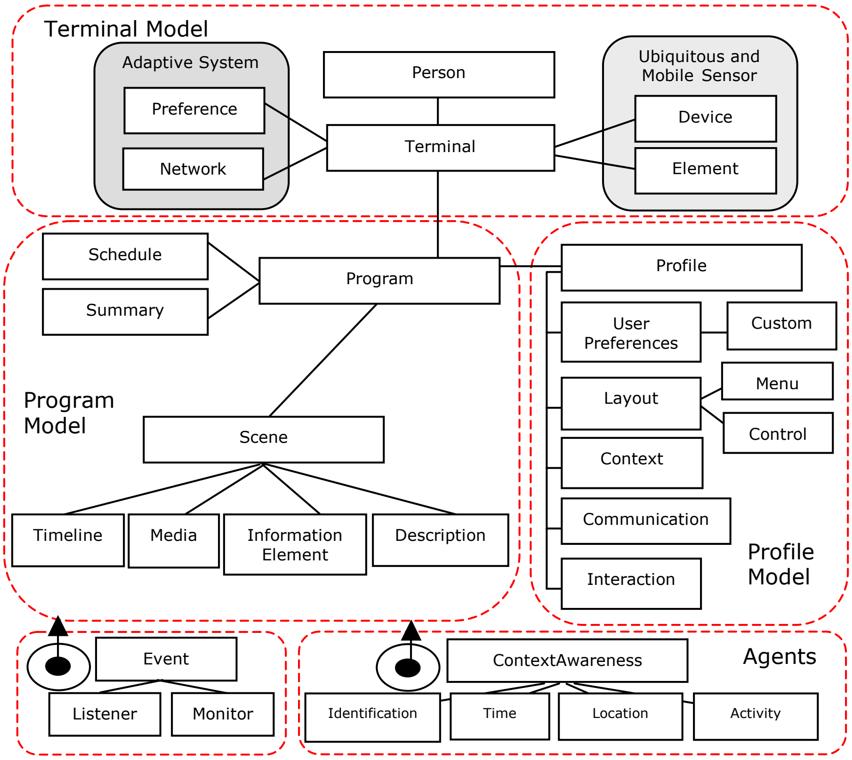

Figure 1: A high-level class modeling for an Interactive TV Application.

The last one is the *Program Model*, where all the objects that compose the *Scene* are present. Their metadata information will be in this part. Technologies like MPEG-7 and XML can be used to implement it. The *Program* entity joins all these parts according to the information provided by the *Terminal* (on the *Terminal Model*), by the *Profile* (on the *Profile Model*) and the *Scene*.

The modeling presented in figure 1 follows an UML-like (*Unified Modeling Language*) syntax [7]. UML 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. Some aspects such as **attributes**, **behavior** and **relationships** (aggregation, specialization and inheritance) can be directly represented into the objects, showing that UML is a good choice for this kind of modeling. The classes of objects can then be described (instantiated) using XML or MPEG-7

Description Schemes. The environment, with its contextual information, can be represented by a set of UML flowcharts. These flowcharts could be class diagram, collaboration diagram, interaction diagram, concurrent objects diagram, events diagram and activities diagram. Furthermore, the UML elements, formally specified, can be transformed in **actions, attributes, conditions, events, methods** (operations) and **code**, traditional features of the object-oriented paradigm.

Agents will be used to allow the awareness of context in different situations. There are four context-aware entities defined by Dey [8]. Therefore four agents will be implemented, one for each entity. In the literature, a computational agent is able to collaborate, via information interchange and services with others programs, to solve complexes problems [14]. 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 [13]. In the figure 2, 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*.

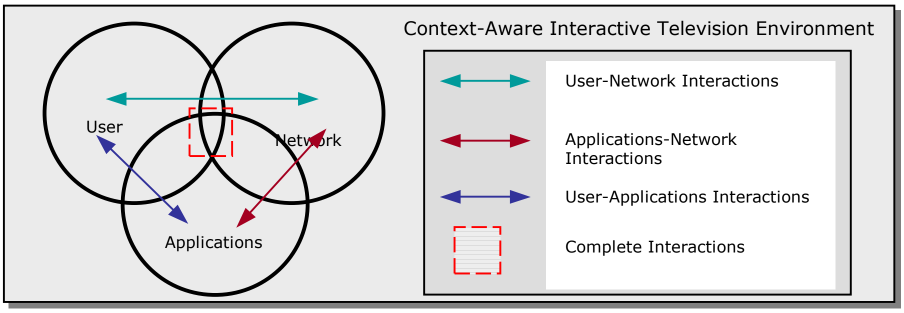

Figure 2: Interactions Scheme in a Context-Aware Interactive TV Application.

Furthermore, agents have its own ideas on to realize tasks and its own schedule of actions [12]. Properties like reactivity, adaptation, pro-activity, autonomy, mobility and social capacity (ability to communicate with others agents via an agents 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 2. 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 2.

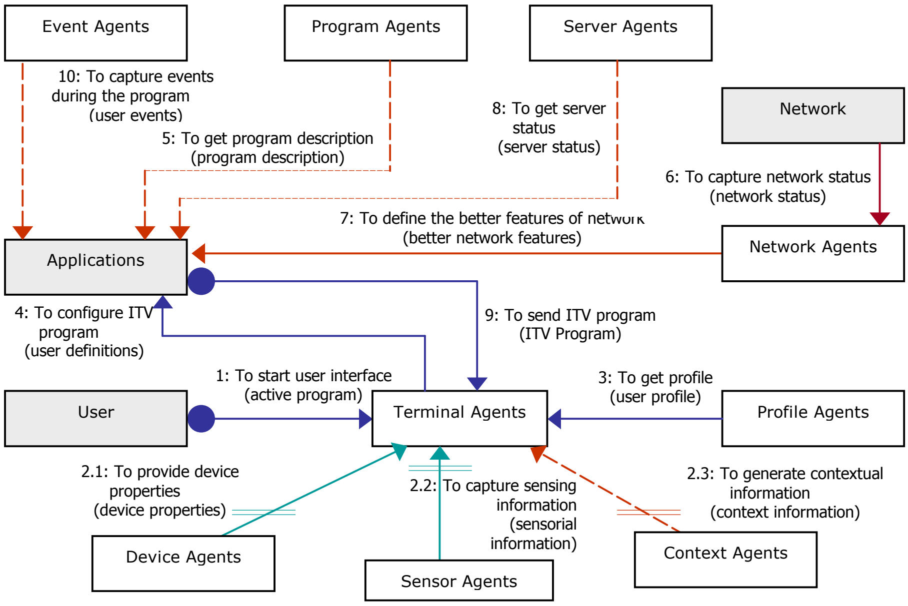

Figure 3: UML Collaboration Diagram – Simplified (for space reasons) Agents-based Approach for Context-Aware Interactive Television Application.

In the approach presented in the figure 3, is important to observe that is showed a simplified version of the complete model proposed for context-aware ITV application. The entities *user*, *applications* and *network* are presented together with a subset of agents proposed for a scenario in a context-aware ITV application. This scenario has its initial state in the action one (1) "*to start user interface*" realized by "*Terminal Agents*", and the final state in the action ten (10) "*to capture events during the program*" realized by "*Event Agents*"; is also relevant the penultimate action (9): "*to send ITV program*" realized by "*Terminal Agents*". Another scenarios can be described from the simplified version of the model presented in the figure 3.

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 an ontology for description of the lifecycle of the interactions supported by agents. This ontology, called OnIT (Ontology for Description of the Agents Communication in Interactive Television Application), is based on FIPA-ACL (Agents Communication Language) [4] 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 4 presents a basic architecture for communication between agents to the *User-Network-Applications* approach.

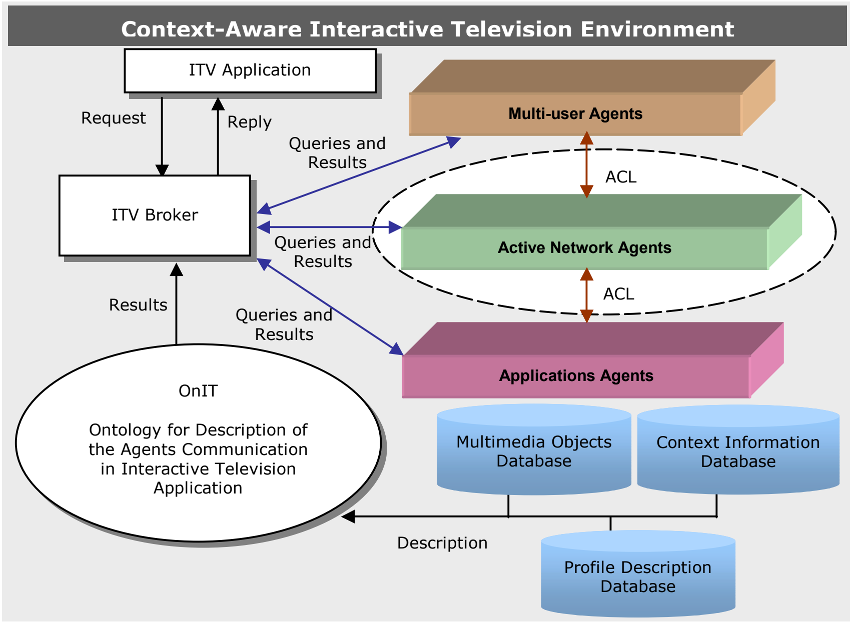

Figure 4: Ontology-based architecture for use of agents in ITV environment.

Communication Description Schemes can be defined, based on XML, for example, to represent the agents' features and its actions in a cooperation session for an ITV application. These schemes are part of the OnIT, that is represented by UML class diagram, as proposed in [6]. The Java technology is being both evaluated and tested for the implementation of the OnIT through JADE (*Java Agents Development Framework*) resources [5].

4 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) [2]. 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 [18]. 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

1. G. D. Abowd; Software Engineering Issues for Ubiquitous Computing; International Conference on Software Engineering (ICSE'99); pp. 75-84; 1999.
2. Advanced Television Enhancement Forum – Enhanced Content Specification (2000). http://www.atvef.com/library/spec1_1a.html.
3. Battista, S; Casalino, F.; Lande, C., MPEG-4: A Multimedia Standard for the Third Millennium, Part 1, IEEE Multimedia, pp 74 – 83, October – December 1999.
4. Bellifemine, F., Poggi, A., Rimassa, G.: Developing multi agent systems with a FIPA-compliant agent framework. In: Software - Practice And Experience, 2001 no. 31, pag 103-128.
5. Bellifemine, F., Poggi, A., Rimassa, G.: JADE A FIPA-compliant agent framework. Proceedings of PAAM'99. London, April 1999, pag.97-108.
6. Blásquez, M.: Building Ontologies at the Knowledge Level using Ontology Design Environment, KAW'98, Canada. (1998)
7. Booch, G.; Rumbaugh, J.; Jacobson, I., UML: Unified Modeling Language for Object-Oriented Development Documentation set. Addendum, Santa Clara, California, Rational Software Corp, July 1996.
8. A. K. Dey, G. D. Abowd; Towards a Better Understanding of Context and Context-Awareness; CHI 2000 - Workshop on The What, Who, Where, When, and How of Context-Awareness; 2000.
9. ISO/MPEG N3201, DDL Working Draft 1.0, MPEG Systems sub-group, ISO, Maui, 1999-December. http://www.cselt.it/mpeg/public/mpeg-7_ddl.zip.
10. ISO/MPEG N3158, Overview of the MPEG-7 Standard, MPEG Requirements Group, ISO, Maui, December 1999. http://www.cselt.it/mpeg/standards/mpeg-7/mpeg-7.htm.
11. ISO/MPEG N3536, MPEG-4 Overview (v.15 – beijing version): http://www.cselt.it/mpeg/standards/mpeg-4/mpeg-4.htm, [03/13/00].
12. Jennings, N. R. ; Wooldrige, M.; Agents Theories, Architectures and Languages: a Survey, Intelligent Agents, p.55-67, 1995.
13. Ketchpel, S.; Genesereth, M; Software Agents; Communications of the ACM, p.48-53, 1994.
14. Moreira, D. A.; Walczowski, L. T.; Using Software Agents to Generate VLSI Layouts; IEEE Expert Intelligent Systems, p.26-32, 1997.
15. Nack, F; Lindsay, A. T., Everything You Wanted to Know About MPEG-7: Part 1, IEEE Multimedia, pp 65 – 77.
16. Albrecht Schmidt. Implicit Human Computer Interaction Through Context. Personal Technologies Volume 4(2&3),. pp191-199, June-2000.
17. David L. Tennenhouse and David J. Wetherall, Towards an Active Network Architecture, in Multimedia Computing and Networking (MMCN '96), 1996, San Jose, CA.
18. TV-Anytime Forum – Specification Series: S2: System Description (Informative) – 2000-October. ftp://tva:tva@ftp.bbc.co.uk/pub.