International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 4 (2012) pp. 244 -254

Interactive Digital Television System using Interactive Service Provider based on Service-Oriented Architecture

Gabriel Massote Prado and Sergio Donizetti Zorzo

Computer Science Department, Federal University of São Carlos
Via Washington Luis, km 235
13565-905 São Carlos - SP, Brazil
{gabriel_prado,zorzo}@dc.ufscar.br

Abstract: Interactive Digital TV systems allow a new range of services concerning broadcasted television content. Thus, interactive programs can be broadcasted allowing user interaction. The interaction implicates in a new way to produce and think about television content. This paper aims at presenting Interactive Service Provider architecture for Interactive Digital TV systems, based on service-oriented architecture, ensuring a standardized communication between client applications and its interactive services. The diversity of broadcasted applications and usage of an interactive channel as external communication in Brazilian context, leads to a case-study considering the social inclusion intention specified in Brazilian Presidential Decree No. 4901.

Keywords: digital TV, service provider architecture, interactive applications

I. Introduction

Digital Television, like other media, has been originated from technological developments related to the form of disseminating information. These developments from the analogue systems allow Digital Television (DTV) systems to produce and broadcast programs with high quality of audio and video. Furthermore, the digital signal enables the transmission of data with audio and video streams [1], offering the ability to add applications, with or without interactive character.

In this context, a new and exciting universe of activities involving production of audio/visual content to include applications that allows program interaction. Interactive Digital Television System (iTV) [1], a new production chain involving the programs and applications production of interactive character, was originated from the possibility to offer applications to viewers. In this system, service providers should be concerned with the diversity of applications and services that can be offered. In this case, the viewer becomes the user of television programs.

Both DTV and iTV systems, use the same definitions and concepts. Their differences are related to the possibility of offering a richer user interaction by interactive systems. Therefore, an iTV system needs to offer an interactive channel support and modify some functionality in production (service provider) and in reception (user or Set-Top Box). These changes enable the features of interactive systems and channel [12] used to send the results of interactive applications to service provider in order to control the content of user interaction, taking into consideration that transmission occurs in the same way.

Changes in production and reception are related to data stream transmission with the audio/video stream and the way data can be offered to the user, like interactive applications associated with a program. Thus, the components in a system are represented by iTV service provider, transmission, reception (Set-Top Box) and interactive channel, responsible for quality of audio/video programs, with the possibility of multiprogramming due to signal compression and the addition of digital data streams along with audio/video.

Some iTV standards such as European (DVB), American (ATSC) and Japanese (ISDB-T), use iTV or DTV television systems characteristics adjusted to the interests of each country or region. The DVB (Digital Video Broadcasting) [7] focuses on the possibility of broadcasting programs with higher quality of audio/video since there is the possibility to add interactive applications; ATSC (Advanced Television System Committee) [8] focuses on the transmission of television programs with the highest digital quality of audio/video without concerning about offering interactivity; ISDB-T (Integrated Services Digital Broadcasting - Terrestrial) [9], basis for the Brazilian System of Digital Television (SBTVD/ISDB-TB standard), has the main objective: broadcasting programs focusing on diversity of interactive applications, offering support for portability and mobility for mobile devices.

In Brazil, the SBTVD is in implementation process from the large to the small urban centers. In the current scenario, the service providers have the capacity to produce audio/visual content, broadcast digital programs and receive the digital signal in the viewer. In small urban centers some service providers broadcast their programs with interactive applications for user interaction.

The motivation of this work is to propose interactive channel architecture based on WebServices to ensure data storage and information request based on concentrating information and services related to user interaction with interactive programs. Thus, programs can be produced with the possibility of adding interactive applications (interactive programs), promoting social inclusion and providing it to user, new and updated information and produce television content. Another motivation is to add knowledge and research in the area of interactive channel in many countries, using a case study with the SBTVD and middleware Ginga [4].

This paper presents a generic and extensible Interactive Service Provider architecture to be used in many platforms and systems, offering data storage and information request facilities based on a Service Oriented Architecture [2]. In order to support the treatment of user interaction information, an Interactive Service Provider (ISP) and an extension (API) to middleware system will be added to interactive channel. It supports interactive applications using external communication. Brazilian System of Digital Television (SBTVD) case study is used by this architecture to prove its use related to system life cycle consisting of service provider, broadcast communication, user (Set-Top Box) and interactive channel.

The paper is organized in seven sections: Section II presents some concepts of Web Services used to identify solutions and implements the architecture suggested; Section III presents the possible behaviors of interactive applications and components necessary for communication with the outside; Section IV presents the work related to channel, offering interactive services; Section V describes the architecture of interactive channel proposal, highlighting its characteristics and its operation before the storage and delivery of services; Section VI presents the Digital Television Architecture with Interactive Service Provider and Section VII presents the study case. Finally, Section VIII presents the conclusions of this work considering the Brazilian context and the work of TVDI already developed at national and international levels.

II. Web Services

The need for the provision of various services and the popularization of the Internet creates a tendency to offer services in a standardized, flexible and extensible in order to simplify the relationship between clients and servers. This relationship of interoperability can be simplified with the use of Web Services.

Web Service is a framework with the ability of developers to ensure standardized and simplified forms of providing services to end users (web applications, mobile, digital TV, among others). A Web Service proposes a framework that brings together papers, models, architectures and other frameworks, such as WSDL (Web Service Definition Language) and SOAP (Simple Object Access Protocol) to ensure provision of services.

To provide information to end users, the framework uses a service orientation [2] to describe business and problems to be solved in the real world. Because these services are offered and accessed by end users depends on the roles each Web Service in this architecture and what the goal or standard to be assigned to this architecture. The roles that can be made by Web Services are identified as service provider, service requestor, intermediaries, initial sender and ultimate receiver.

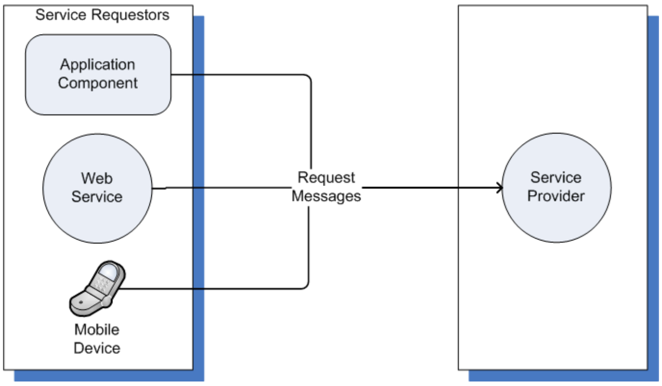


Figure 1. Service requester and service provider - adapted from [2]

A Web Service, illustrated by figure 1, takes the role of service provider (service provider) to receive service requests via external elements, comparing them to the server to a client / server communication where data is requested by a customer and information is returned by the server.

Service requestors (requester of services) are called all kinds of external elements who order a service that can be understood by a Web Service with service provider role. A Web application, or even a Mobile Web Service can be considered an applicant for service, comparing in a client / server customer calling information to a server.

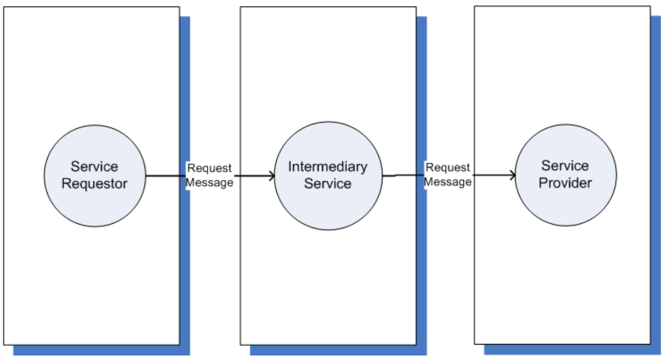


Figure 2. Roles of Web Services - adapted from [2]

The great flexibility and scalability, Web Services can take on other roles, shown in figure 2, that do not limit the framework for client / server communication. With this, another feature is the ability of Web Services with the roles of intermediaries (brokers). They can work as routers or message processor before they reach their destination, which can be classified into passive and active intermediary. The classification as passive intermediaries must not modify the SOAP message after the interpretation of its header. Intermediaries assets are different from passive intermediaries through modification of the data present in the header and message content.

Web Services and external elements can be classified as service requester. When the starting point of transmission of the message part of one of these elements, they are classified as initial senders. Unlike the original sender, the ultimate receivers are service providers considered the ultimate receiver of the message sent.

With the variety of roles exercised by Web Services and the necessity of offering services for applications, the different roles can be used to compose specific services. Service composition is a term used to describe a specific role of a Web Service, and yes, a set of Web Services with the ultimate goal of fulfilling a task. For this set meets a task, its main task is divided into sub-tasks that are specific to each Web Service. Each Web Service this composition is classified as a member (member of the composition) of the service, because they are responsible for chores or tasks related to the main task.

With the composition of services, from the union of different roles for Web Services, several models can be proposed from the services compositions of Web Services. Some models of the more common services that are viewed in the real world can be classified as business model, utilities and control. The business model represents sets of business logic, usually being used in compositions of services. To represent situations with generic features, the service model used is the utilities. Since the control model is used by Web Services responsible for controlling compositions of services.

To obtain the weak coupling (less dependence between Web Services) in their communications, Web Services use document-description services for communication between service requester and service providers.

The service description document, named WSDL, aims to describe endpoints (recipient of the messages) offering service requester logical and physical descriptions of available services enabling the routing of the message. Generally, the endpoints are represented by Web Services providers.

To describe the services present at the endpoint, a WSDL document is divided into a piece of abstract description and another description of concrete. The abstract description of a WSDL (Web Service Definition Language) contains information regarding the interface of the Web Service, describing their services and their features and parameters of input and output. Unlike the abstract description, the description focuses on concrete physical description of the technologies to be used to implement the logic in Web Services and gifts to the abstract interface has references to a physical endpoint.

Web Services provide an architecture designed to provide automated services and ensure less dependence between the services. For the logical and physical description of Web Services and services, it uses the WSDL document.

The SOAP (Simple Object Access Protocol) was developed with the aim of establishing a standardized form of communication between departments in order to be flexible and extensible.

Communication is based on sending messages such as envelopes (Envelope) and can be extended and customized to ensure flexibility and depth needed. An envelope can be divided into header (header) and body (body). The header provides independence to the posts holding all the information necessary for services included in a Web Service. This information is related to routing and processing of messages. With the description of the message being sent, the message content, covering the body of the envelope, can be delivered to its destination with the characteristics proposed by the SOAP protocol.

Other features that can be added to a SOAP message are related to the description of faults (fault), routing (path message) and sending attachments (attach). The description of faults is used when there is the need to involve a possibility of a failure in one part of the information being transmitted. The option of routing, built during the sending of messages, has to route a message departing from its initial sender to its receiving end, including intermediaries. Since the attachment option allows sending a SOAP message attachments (images, video, data, and others) using a different medium from that used for the sending of envelopes.

Using technologies of interactive television and Web Services an aggregation of both can generated a rich scenario to improve knowledge and creates a new way to produce interactive programs.

III. Interactive Service

Digital Television Systems is related to economical and governmental interests like social inclusion and democratization of information through the development of e-government, e-learning and e-commerce applications, normally classified on the Internet context as t-government, t-learning and t-commerce applications.

In order to promote all the interests of the country, the systems should enable the incorporation of interactive applications to television content, allowing the viewer's (user) interaction with applications, while iTV systems need to ensure these supports in Set- Top Box (STB), more specifically, in the middleware. Moreover, it is necessary an interactive channel to send data to the service provider transforming the viewer in a TV user.

With the possibility of interactive applications broadcaster, the iTV systems start to worry not only about the cycle of programs but also about the applications. A television program without applications has a continuous life cycle being generated, transmitted and received. When applications are broadcasted, this continuous program cycle is the same but other life cycle is included in the program. This can be classified in pseudo-interactive or interactive cycle, involving programs and applications.

A pseudo-interactive program limits the application life cycle to exchange information to STB, that can write and read data using the middleware or deliver content to users, but without any form of external communication using the interactive channel .

The interactive programs have the same pseudo-interactive characteristics. The difference is the possibility of having communication through interactive channel. This feature allows the service provider to add more functionality, allowing t-commerce, t-government and t-learning applications to be broadcasted.

A new form of communication between service provider and user becomes possible. Digital television systems must enable an interactive channel communication located between service providers and users to send and receive information. [13]. For user interaction, information can be stored and offered in content form. An Interactive Service Provider (ISP) must be added in the interactive channel [1], [13] for the service provider. In iTV systems the ISP may be located at the network or at a company hired to provide such service.

The interactive channel is designed to ensure two-way communication between service providers and users. It guarantees that data resulting from user interaction can be sent to ISP and service providers can request information related to it. New and updated content is also produced and delivered to users through television programs or interactive applications.

The technologies used to ensure data transport through the interactive channel are influenced by several physical and social factors. Choosing a technology is influenced by socioeconomic differences, physical and technological limitations present in every region of the country [1], [13]. Interactive channel architecture shouldn't be restricted to a single technology of access to internet. It enables the use of different forms of internet access, considering lowest cost available in each region of the country, due to the intent of social inclusion.

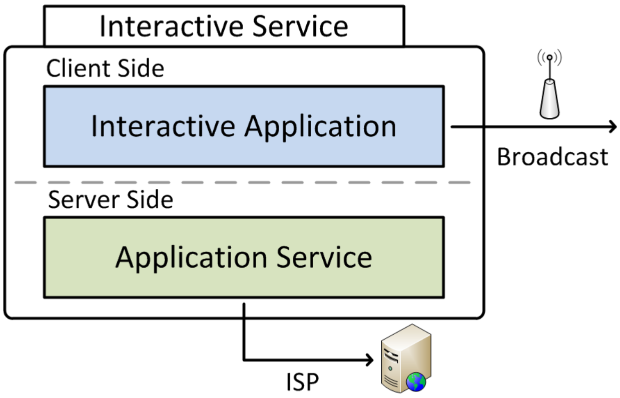


Figure 3. Interactive Service

With interactive applications broadcasting, the digital television systems have a necessity of storing information about user interaction or other useful information about service providers. Thus, programs with these features require specific forms (services) of data treatment to send and provide information request in ISP. These specific forms can be organized in an Interactive Service [12] based on a client side and on a server side as shown in figure 3.

The way to organize using client (interactive application) and server side provides relationship between application and their server functionalities. This corresponds to association originated when program and their Interactive Service are created.

The client side is represented by interactive applications that will be produced and broadcasted by service providers. This side involves all types of applications that can be associated in television context, involving digital systems that provide support to interactive applications.

Server side is represented by server applications and normally is built using technologies in Web Services context. The organization of these services is based on Service-Oriented Architecture (SOA) which has business functionalities and processes called services. SOA defines that the services are developed; deployed and integrated regardless the application and computing platform on which services will run [18]. Web Services support the interoperability of systems and application integration services through network applications, allowing the system to adopt of SOA paradigm [19].

SOA development process can be applied to two approaches: bottom-up and top-down. The bottom-up strategy is tactically focused on fulfilling the requirements for an immediate business priority especially in cases of new service creation. On the other hand, the top-down strategy analyses the business processes before developing the services [18], [20].

The logic organization of services execution is called choreography or orchestration. Choreography is the composition of business processes where the picture of a coordinator which manages and coordinates the other processes cannot be found. Orchestration is a workflow conception for the composition of services where a central coordinator manages the service calls or any internal changes if necessary [21].

This form of production involving television, interactive applications and application services enables broadcasters to concentrate on the television production programs and Interactive Services [12]. Thus, there is the possibility to create a new production line, where the television programs continue to display the audio/video applications and Interactive Services are produced and offered to users.

The next session presents some related work using Digital Television Systems, and java technology to develop their systems. These related works contribute with some features of the architecture presented in this paper.

IV. Related Works

With the adoption of the Brazilian standard digital TV (ISDB-TB) have emerged numerous studies in the area of digital television with the aim of adding knowledge and offer alternatives to this new technology. JiTV platform, developed by TVDILab (Laboratory of Interactive Digital Television) Resident PUC Minas Campus Pocos de Caldas is one of these alternatives to SBTVD.

The interactive programs on the platform JiTV (pTVDI) are structured using XML schemas that describe the stations, programs and additional information (catalog of genres, languages and closed captioning). With the information contained in XML files, media and interactive data, it is possible to adapt the content of interactive programs for viewers [10].

The goal is to provide a platform JiTV set of features that support all cycles of an interactive program from the creation and authoring to the reception by viewers, offering a set of tools for the purpose of authorship and transmission of programs and ensure the ability to viewing of programs independent of the access terminal (TV, PDA, personal computers and cellphones, among others).

At the station, part of the application offers JiTVStudio production, which is responsible for every part of production and publishing of interactive programming. In the part of the transmission is JiTVGCarrossel JiTVStreamer and which have the characteristic of organizing data for transmission of programming and interactive programs to viewers.

In the viewer JiTV platform operates on a layer of middleware offering JiTVPlayer, responsible for displaying the viewer's audio and video. Above the presentation layer are the applications residing, which are installed in the STB to be received, and the non-resident applications, which are available from station and transmitted by data carousel.

Once the platform has the capability of authoring programs, transmission and reception, the programs must have support for communication with the external environment so that data resulting from the interaction to be stored for an ISP Provider (Interactive Services). As part of work, the platform also offers a set of tools to the viewer, and issuing a return channel to support interactive applications that depend on the return channel [10] and can be visualized in figure 4.

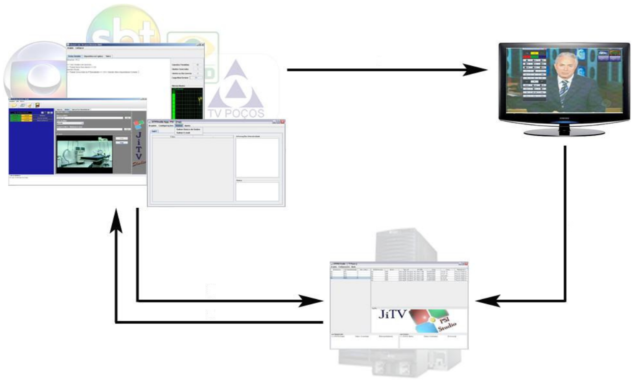


Figure 4. Structure with ISP platform JiTV

In the viewer, the interactive applications offer the feature of communication with the outside world by sending XML files containing the result of interaction with the viewer. The file is sent to the ISP to be stored and eventually used to generate information for the station.

In issuing the application JiTVStudio PSI has the feature of turning the station's need for information on data to be offered by JiTVPSI Studio. Since these data are ready they are sent to the JiTVPSI Studio, and consequently, an answer is expected that the information requested is available for use by the station.

In the part referring to the return channel exists an intermediate module between broadcaster and viewer with the ability to offer the functionality of a ISP. This module - JiTVPSI Studio - has characteristics of storage and data processing. The storage feature is responsible for receiving connections from viewers await the sending of the XML files related to the interaction and implementation to store them. Since the functionality of data processing is responsible for waiting for connections from stations asking for information relating to data (XML) stored. Thus, allowing data to be collected and information is sent to the station when requested. With these components JiTV the platform is capable of addressing all the components in a system TVDI.

For the U.S. Digital TV system ATSC-ACAP supports ensure support for interactive applications one-way and two-tay a return channel model should be proposed. The WRCs (Web-based Return Channel) architecture suggests a return channel based on the web to support the wide variety of interactions that can be transmitted via broadcast network (EPG, t-commerce applications, t-government, t-learning, among others) and, consequently, the ability of feedback through the contents of the data resulting from the interaction of viewers with interactive programs [11]. Another important feature is the ability to connect multiple viewers.

To ensure these services architecture is divided into three types of servers and a switch with large capacity distribution (layer 4 switch). The three types of servers are classified as storage servers (database servers) servers, the return channel interactivity (interactive return channel servers) and Web Services.

The first layer of working servers to store information related to non-interactive services and interactive services broadcast by the network. With this information it is possible to identify broadcast services, location and type of other servers that are related to data transmission, interactive applications, return channel, certificates of users, user accounts, system errors and the backup list, among others.

The second layer of servers is represented for the authors as the brains of interactive services of WRCs. First these servers receive data related to the fate of interactivity and running as scheduled. Another feature of these servers is the transmission of information corresponding to users' requests for Web Services.

The third layer has the characteristic of servers to manage the session control to provide support for HTTP and TCP / IP made by the terminals (STB) present in the users.

So that services can be handled on a consistent WRCs uses the EJB (Enterprise Java Beans), which helps in simplifying the development and performance of interactive applications. The choice of EJB technology is due to the fact that the choice of platform ACAP-J and can be divided into Entity Beans and Session Beans. Entity Beans are used for services related to management and system programming, are permanent and are only deleted when the program is terminated. Several programs can use this bean to be shared on the network, running remotely from the servers. Unlike Entity Beans, Session Beans, which are used for services related to television viewers, are not permanent and are usually associated with a connection.

Combining the services offered by WRCs Beans with EJB-related services to users are implemented using Session Beans and the related management and scheduling system are implemented using Entity Beans.

The validation of WRCs is performed using an interactive application that provides the actors in a television program. The interaction of information are sent to the WRCs through the network infrastructure available. To be received, the user certificate and its validity is checked. Being confirmed, the result of the interaction is stored by WRCs and returned to users via the broadcast network or an IP network. The management of WRCs can be done through a web application that allows the control of its main features.

The authors' justification for the use of Web technologies to implement this type of return channel (WRCs) is the initiative to start a connection by the user in the same way that the operation of web applications [11].

[12] suggests a framework model of service-oriented architecture for interactive applications using interactivity channel through low cost satellite transmission (project SATMODE). The framework aims at offering client and server side features to create interactive services, consisting of interactive applications and specific application services, where interfaces are used on both sides to establish a standard communication.

Besides offering some features [10], [11], [12] related with communication using an ISP and Web Services, the architecture developed in this paper is characterized by the use of a broker for specific services hub, the standardization of communication through APIs and the use of a registry server UDDI.

V. Interactive Service Provider Architecture

As a first purpose, the Interactive Service Provider (ISP) architecture offers a support of services publication, concerning an application service provider with the feature of concentrating common services in a unique localization. The second one is to offer a support to publish services outside the ISP, such as bank and some governmental services already developed. Other feature derive from the second purpose is the possibility of interconnecting more than one ISP in order to create a global communication.

This section is organized in three subsections: the first describes the generic ISP architecture with all features about publication and service access; the second describes the Client API (CISP) that ensures a standardized communication for clients; and finally, the third describes the standard message used to communicate with ISP and their few parameters.

A. Generic Interactive Service Provider

The generic Interactive Service Provider (ISP) was designed from a service-oriented architecture [2] with the aim at ensuring a generic architecture for iTV systems. It is a standardized and expandable architecture with the guarantee of service publication and centralized database of common information.

The architecture is based on a composition of Web Services (services) enabling the addition of new services without modification to the architecture and to the way to access the services. In order to meet these features, Web Services were adopted due to its independence of application and because they work with Web-based message requests [2].

In this service composition architecture, the services were rated to work with intermediaries or service providers roles [2]. The first category of services in this architecture is located between the client applications and services classified as service providers. Their role is to receive requests from client applications and forwarding requests to service providers, as a router. The second category is related to application services responsible for the storage and information delivery related with interactive application. Thus, the architecture of the ISP, in figure 5, is divided into two service layers: front-end and back-end layers.

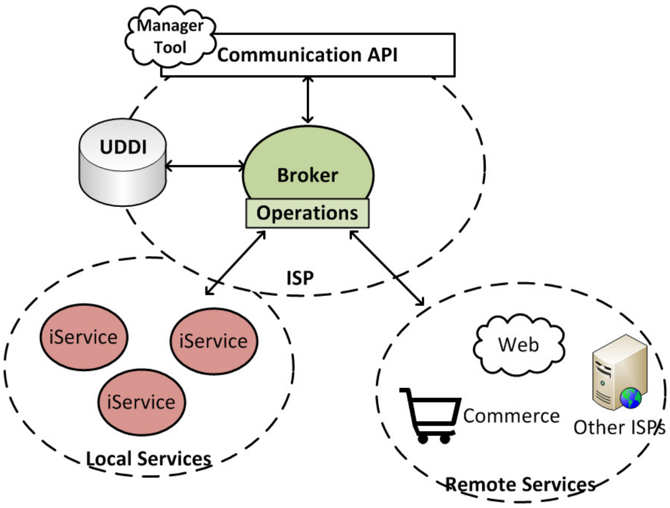


Figure 5. Generic ISP Architecture

The front-end layer is the only way clients can access the ISP. This layer offers ways to publish and access application service; and an intermediary service (broker) who acts generically offering a standard interface (form of communication) for the communication between applications and services manages the access.

The broker (front-end service) is the standard service of communication with ISP providing three logic operations (postService, listServices and Route) responsible for publishing service, requesting service and information. PostService operation is responsible for ensuring forms of publishing application services available in ISP. Its operation is based on the receipt of a file, which has the service logic (classified as Local Services), storing the file on the server and registering the service on a services registry (UDDI - Universal Description, Discovery and Integration) [2]. Other functionality enables external service (classified as Remote Services) produced in other ISP, Web or Commerce Applications to be reached by ISP requested. ListServices operation offers a list of services registered in UDDI. When requested, the operation checks the services on UDDI getting the corresponding file descriptor of service (WSDL - Web Services Description Language) [2] for service information.

Finally, the Router provides a standard communication interface with interactive application services and the routing feature to forward requests to back-end layer. When a service request is received, the router operation accesses the UDDI to verify if service is available to process the request. Once it is available the request is redirected to a service that returns the response to be directed to the client.

Backend layer represents application services in ISP aimed at offering a fast way to deploy/publication and access the services. The proposal was to use this layer was proposed to be used before the great possibility of Digital TV application to produce and edit their services and applications. Thus, new developed/updated applications only need to deploy their service using broker to offer the new features. Each service in back-end layer has its own features regardless the broker. This creates the possibility of using these services in architecture.

The general ISP behavior with front-end and back-end layers, illustrated in figure 5, offers a standard communication interface with services published in the backend layer allowing different services to be deployed/published. This way clients need only to access the ISP and additional services can be added in architecture without changing the access way used by clients to access application services. To use all the features offered by ISP, consequently service broker, interactive services should be published using the ISP Web Tool. On the user side, the Client ISP API helps user to communicate with ISP to send and receive information.

This architecture was developed using Java language to services development and Axis2 Web Service engine [16] to publish services on Web. The API was developed to ensure standard communication with ISP front-end layer (broker service) offering Java and Lua support. The ISP Manager Tool was developed using J2EE technology and Java implementation of Client ISP API to communicate with broker operations.

B. Client ISP API

In order the client to easily access ISP frontend layer, an API (Client API-CISP) in Java, C++ and Lua was proposed and aggregated to architecture. Their importance to the architecture is related to simplify the client communication, abstracting to the application developer the configured layer. With this API, the application development is concentrated only on logic of application and message content.

The CISP was developed using a dynamic way to create a SOAP message offered by Axis2 engine (AXIOM) and all abstraction from AXIOM is done by CISP.

The interactive applications received by users are transmitted as data streams along with audio / video from a television program. The middleware is responsible for identifying the types of data received and to allow to be stored and offered to users. Once the applications received are considered interactive, its data are no longer treated only locally and require communication with an external environment to work correctly. The way in which data is available to the external environment must be ensured by the middleware, STB on this user, abstracting the hardware settings and ensuring a generic form of communication. To ensure this form of communication, an additional module - Interactivity Module - of interactivity should be added to the user and, respectively, for the middleware.

Interactivity Module should provide resources for developers to simplify the hardware settings and standardization of data, disregarding the need for knowledge of how the data will be sent, only worrying about the characteristics of communication and information to be sent. This module is divided into two sub-modules, one responsible for providing the developer with an API for attributes related to the configuration and application data are stored temporarily in memory or on a base of data and another - Engine Module - responsible for standardizing data and application settings offering a communication with the ISP.

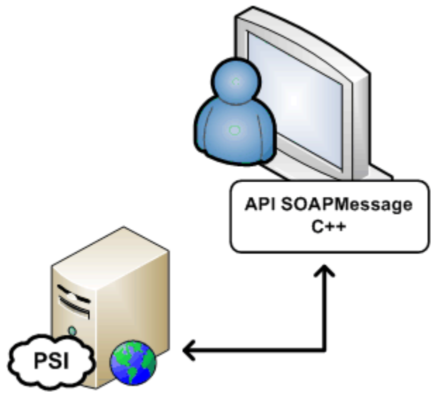


Figure 6. Communication between applications and interactive modules built-in interactive channel

Tha API module should be an extension of the middleware as a component capable of supporting interactive channel and will be responsible for delivering the developer of interactive applications several methods of assigning characteristics to outgoing messages. The objective of this module is to ensure that the developer only worry with the characteristics of the message and its contents, without worrying about shipping configurations, protocols and destination of the message. It also offers the functionality to notify the need to send data to the Engine Module, which standardizes the data. The need for confirmation of receipt, type of service (associated with the running application) and are content to send some of the information that the developer should not worry.

The Engine Module is responsible for all the logic and data processing assigned by the developer and the user interaction with the interactive application. Upon being notified of the need for early communication, the file description and location of services (WSDL) should be accessed for information where the message should be sent, ie, the location of the ISP. Once the location of the ISP is determined, the data allocated by the API Module are obtained and adapted for a SOAP message or SOAP Envelope (used as standard for communication between Web Services and their clients), where the settings are added to the message header and content to be sent is added to the body of the message. By completing the standardization process of transmission, communication with the ISP should be started following some characteristics of a specific protocol proposed that would be shown in next session. Receiving information from the ISP should also be a feature of Interactivity Module with the protocol, ensuring two-way communication between user and broadcast.

Describing the module channel, through the general operation of its features by using interactive applications is illustrated in figure 6. When an application comes into play mode and after the interaction process, the user can request that the communication process is started. To initiate communication, the application gets access to the methods available in the API using the Lua language (scripting language for accessing resources middleware) determining a set of attributes related to data regarding the configuration of the message and content of interaction. After the assignment, Engine Module obtains the location of the ISP and together make up the settings assigned to the message header. After the header is composed, the data of the interaction are added to the body of the message. As part of the final process of communication, the message is sent to the ISP.

Once the communication process has started, this module will ensure the ability to exchange messages between the interactive application and service specific present in ISP through the communication protocol being proposed.

C. Standard Message

In order to standardize the communication between client applications and ISP, some configurations need to be added in a SOAP message [2], so that the broker can understand the message and redirect to application service. These messages, or SOAP envelop, are divided in a header, in which application services information are configured, and a body containing the message content.

Some formatting or mandatory fields should compose the message being sent. Through the header, the application must specify which application services and operation of the service you want to access, enabling the message to be interpreted and routed through the broker. The data is sent in the body to be stored by the service or request information.

This standard message is an essential component for understanding the requests by the ISP, more specifically by the broker, even allowing applications to communicate with ISP. The message doesn't need to be configured if CISP is used. Reminding that de API abstracts all communication configurations.

The importance of the protocol is due to a communication performed in a correct and safe will result in ensuring the integrity of data transferred.

The purpose of this standard message is based on the transport protocols TCP and UDP for applications that require confirmation of data integrity and for those that do not need to use an application-level standard message that ensures the necessary characteristics for the transport data.

For communication to be in accordance with the Brazilian standard in which specifies how the communication should be made of the applications using the interactive channel, the standard message to be proposed should be concerned with the forms of connection, data integrity and how the data will be transported. This standard message can be used by all client applications (interactive applications, Web applications, Mobile, and others) who wish to communicate with the ISP.

Interactive applications in the use the interactive channel should possess the following features: connection, data transfer and disconnection [1], [13]. The connection and disconnection are implemented by the middleware notifies the physical means to connect and disconnect the application of an external environment. For other applications that request services to ISP, the forms of connection and disconnection are implemented according to the technology used.

After the step of connecting and disconnecting are defined, the integrity of data transport and application execution are important characteristics of communication. The ISP and interactive applications may have different interests with the messages used for communication. Some will require confirmation of receipt of the data to complete the application, while others will only send data and will be finalized without acknowledgment.

This standard message uses SOAP messages to convey the settings needed for communication and data generated by applications or by ISP.

Through the header of a SOAP message, the client (or middleware ISP) shall specify the need for confirmation or otherwise of completion of the application, if the message is the request (meaning client / ISP) or response (meaning ISP / client), the type of service and whether it is a message of completion of application.

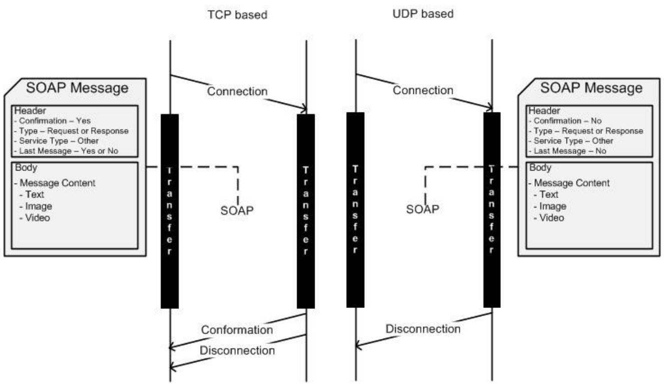


Figure 7. Standard message-based TCP and UDP

As illustrated in figure 7, communication using this protocol can be done in two ways. One with the possibility of confirmation of completion of implementation of the ISP and the other coming no matter when an application must be completed. With confirmation, the standard message must have the features enabled in the confirmation header, which allows the ISP to know whether a message should be returned to the application after receiving a request. If it is enabled, after all messages between the application and ISP are exchanged, a confirmation message is sent to the application notifying the end of the communication. In the case of a communication without confirmation, the communication between application and ISP runs in the same way, differing by the absence of a completion message for the application, enabling the completion of the application at any time of his execution.

VI. Digital Television Architecture with Interactive Service Provider

The ISP architecture was planned to be used with iTV systems, to support interactive services composed by interactive applications and their application services. In iTV context, using ISP architecture requires some parts of architecture to be added in iTV architecture to offer interactive service support. These parts are covered for additional ISP elements in service provider and user side. Figure 8 shows the iTV architecture with ISP.

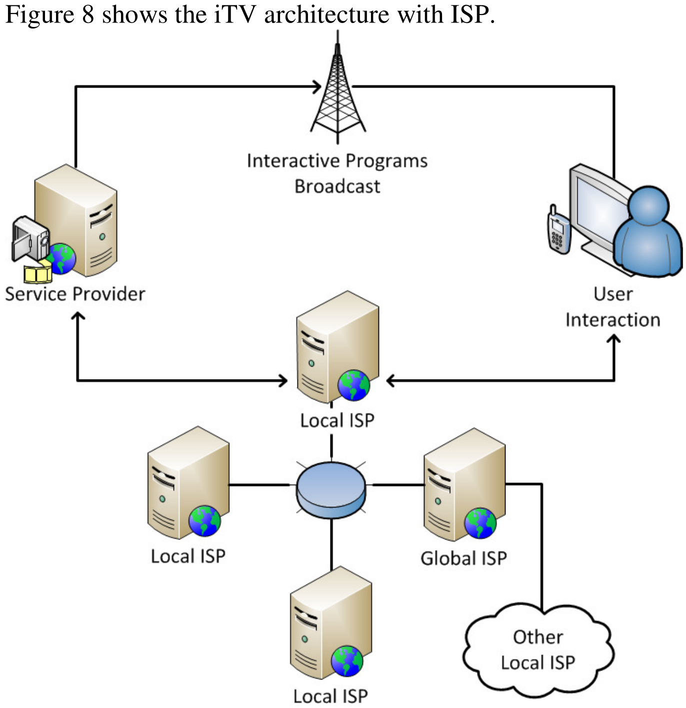


Figure 8. Interactive Digital Television Architecture with ISP

On the user side, the focus is on the interactive applications communication with ISP to send and receive data. To ensure this form of communication, developers need to use supported languages implemented in CISP ensuring correct communication with ISP with interactive applications developed in their language. Normally, the CISP is used with imperative language responsible for ensuring the logical behavior of application. Some applications have declarative and imperative languages working together and some have only imperative language working alone.

Since the applications were developed using CISP, all information resulted from user interaction will be sent to the ISP and will be received by the broker and redirected to the correct application service in the backend layer.

On the service provider's side, the interests are involved in how to obtain information resulting from the user interaction with interactive applications to produce new/updated content and new applications.

Therefore, the service providers need to organize their environments with two main features: support to publish application services and a way to access these services to obtain processed information. The publication of application services can be done through the ISP Manager Tool using the postService offered by broker. To obtain information about published services and operations, listServices can be used. Finally, to request information about result and processed data related in user interaction, a request message can be sent using the route operation that will be responsible for finding and redirecting the message for an application service and return the answer about the message requested. The communication process can be done by CISP.

With the ISP architecture the service providers can obtain new interactive applications and real-time update by the interactive channel information that can be used for the production of new content production.

VII. Case Study

Based in the Presidential Decree No. 4901 [5], the assumptions of the Brazilian government is not restricted to transmission of digital content, like many other systems. The focus of the system is to offer the user the better interaction with programs through justification to prove social inclusion and democratization of information in all the country. Brazilian government likes to develop an objective system focusing the needs specified in decree [5] and offering higher quality of video and application execution.

In Brazilian context, in which SBTVD focus is on developing social inclusion programs, this work presents the development of an interactive application for people with special needs in order to enable the visually impaired to participate in the process of technological development of Interactive Digital Television in Brazil [3].

The technologies used in the development process are in agreement with Brazilian system and need to be tested. A case study of an interactive application questionnaire for visually impaired on client side and the ISP architecture are used to guarantee the access to the interactive application external communication.

The interactive application gives the user several questions on different topics with the narration aid to support visually impaired people. When the application is received, the user can select the desired genre and browse the questions. The figure 9 illustrates the following scenario: the user watches TV when the system inquires him about three themes (in the left) or shows him about the results of the quiz (in the right). In the end of the interaction, the answers for the questions are forwarded to the ISP. The response processed by the ISP has a set of information to display to the user as showing true questions, the total score of true questions and what is your ranking based on other users.

Figure 9. Interactive Application

The client side of this case study uses Ginga-NCL implementation to offer interactive support of NCL [15] applications and Lua imperative language with CISP that offers standard communication with ISP. When application is received by middleware, the user needs to interact in order to show something. Interacting with the application some question genres are able to be selected and start all questions about selected genre. Interactive options are showed using NCL and the sound used to ensure visually impaired are started through Lua. This situation is illustrated by figure 9 in the left picture. After user interaction, the information about all questions answered are sent using ICSP Lua implementation, providing all communication support with interactive service and all answers are returned to user through application and sound support.

ISP architecture is used to ensure external communication for questionnaire application. The first step in developing the service was executed by a Java developer, generating logic of service and using the ISP Manager Tool, offered by ISP to deploy and publish the service on Web. With the deployed and published service, clients can use all features offered by the service.

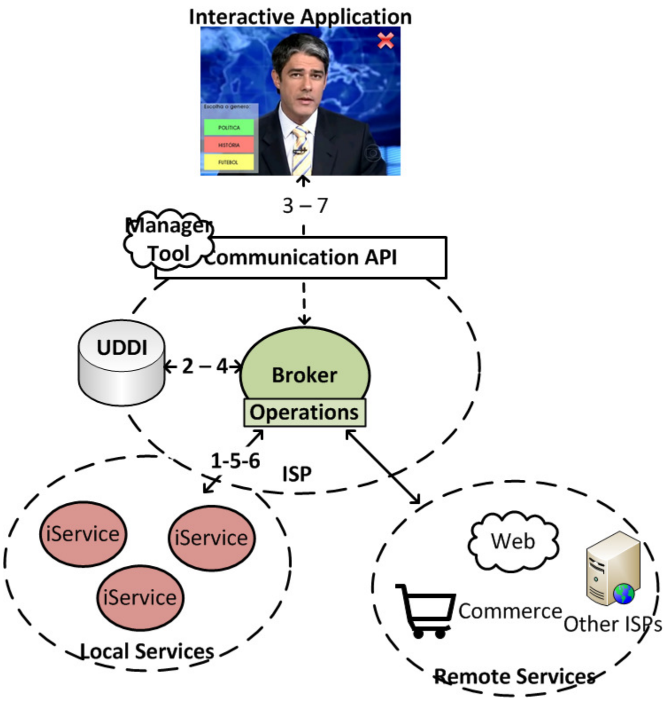


Figure 10. ISP Architecture in the Case Study

General behavior represented by figure 10 uses all architecture modules presented in this paper. The general behavior can be described in some steps:

1. Indicates when the service provider developer publishes the application service after its development using ISP Manager Tool;
2. The service is registered in UDDI in order to be verified by route operation to redirect incoming requests;
3. After user interaction, a message is sent to ISP (broker servicer) to be redirected to an application service and stored;
4. When the request is received from broker, a consultation is made in UDDI to verify service existence;
5. If service is registered, the message is redirected to application service to be processed;
6. When application service receives the message, the content is processed and returned to broker with the information requested;
7. The broker finishes the process redirecting the processed message to interactive application to show information.

In high level behavior, when user interacts with the application, some sound notification is started, complementing text information shown on screen, describing menus, alternatives and answers. When interaction is completed, the resulting data containing the chosen genre and its answers are sent to the ISP. To obtain the information resulting from the interaction, the service processes the information providing a final response to the user containing the correct answers, the final score and the rank to be compared with other users.

A simulation processes was executed with some conditions in resident lab in Federal University of São Carlos, and applications, like all interactive applications, has a life cycle defined by the way that directly interact with the user. To be produced by the service provider together with the PSI, the application is sent to users via an IP network by simulating a network broadcast. The STB receives the data streams containing audio / video / data delivery data for the middleware that notifies the user of the existence of an interactive application. After being available, two options are offered to the user, choose to send their data or dismiss the application. When choosing from available data, they are sent and stored by the specific service waiting for the station servicing to obtain data for information is produced, released and sent. This information request is made web support allowing data analysis and, consequently, the use of information for which the broadcaster has control of sitting on their programs.

This case study is represented by interactive application development followed by service development and publication. After development process, the interactive application is able to be broadcasted and executed in the middleware Ginga implementation Ginga-NCL.

VIII. Conclusion

The purpose of this work is to present architecture able to offer a concentration of services related to ISP by service providers. Other feature is ensuring the generality to use the architecture in several iTV systems and maintain the generality offering generic support to access from other client application and systems, different from Digital Television Systems.

The main motivation of this work is related to Brazilian context that is based on the governmental interest mentioned in the Presidential Decree No. 4901, which highlights the intention of ISDB-TB to provide the social and cultural inclusion of the country. This work is characterized to afford interactive channel architecture to support interactive applications that uses Brazilian middleware Ginga.

As future work, the architecture has two interesting points to improve. The improvements are related to the capacity of offering a deployment support for interactive services, not only for application services. This would allow interactive applications to be stored together with their services and when service providers want to broadcast them, they request the interactive applications. The other improvements are related to the capacity of supporting a lot of connections at the same time and if the architecture will support this. The studies need to be made in the clustering area to distribute the architecture in other nodes. Some contribution related to development language is how to improve time communication between clients and ISP architecture. With this, JSON communication can be tested and applied in client API and broker service to experience how it works.

Thus, we conclude that this work offers an architecture that presents the main contribution, offering a generic and expandable ISP for the publication of application services, allowing multiple services to be added without the architecture and form of access to be altered.

Acknowledgment

Capes for the financial support to Gabriel M. Prado.

References

[1] Edward, M: “iTV Handbook: Technologies & Standards”. Prentice Hall. (2003).
[2] Erl, T.: “Service-Oriented Architecture (SOA): Concepts, Technology, and Design”. Prentice Hall. (2005).
[3] Fernandes, J.; Lemos, G.: Interactive Television in Brazil: System Software and the Digital Divide. EuroiTV, (2007).
[4] Soares, L. F. G.; Rodrigues, R. F.; Moreno, M. F.: “Ginga-NCL: the Declarative Environment of the Brazilian Digital TV System”. JBCS. (2007)
[5] Brazilian Presidential Decree n° 4901. 2003. Available in: http://sbtvd.cpqd.com.br/downloads/decreto_4901_2003.pdf (2009)
[6] ABNT NBR 15607-1: “Norma para Televisão Digital Terrestre - Canal de Interatividade - Parte 1: Protocolos, Interfaces Físicas e Interfaces de Software”. Available in: http://www.forumsbtvd.org.br/materias.asp? id=112 (2009)
[7] DVB: “European Digital Television Standard”. Available in: http://www.dvb.org (2009)
[8] ATSC: “American Digital Television Standard”. Available in: http://www.atsc.org/cms (2009)
[9] ISDB-T: “Japanese Digital Television Standard”. Available in: http://www.dibeg.org (2009)
[10] Santos Junior, J. B. A., I.C. ; Morselli Junior, J. C. M. ; Teixeira, F. C.; Prado, G. M.; Ávila, P. M.: “Back Channel in Interactive Digital Television Systems: Strategies for Prototyping Applications Using an Interactive Service Provider”. ICEIS, v. 1. (2009)
[11] Seok, J. M.; Choi, J. H.; Choi, J. S.: “The Design of Web-based Return Channel System in the Data Broadcast Services”. ICACT, v. 2. (2005)
[12] Milos, F.; Basic, H.; Zagar, M.: “Model of Application Framework for Resource Limited iTV Systems”. Information Technology Interfaces, 2006. 28th International Conference. p. 633 -638. (2006)
[13] Manhães, M. A. R.; P. J. Shieh.; et al.: “Canal de Interatividade em TV Digital”. Caderno CPqD Tecnologia, v. 1. (2005)
[14] LUA. Available in: <http://www.lua.org>. Accessed in: 05/06/10
[15] NCL - Nested Context Language. Available in: <http://www.ncl.org.br>. Accessed in 1/07/10.
[16] Axis2 - Web Service Engine. Available in: <http://ws.apache.org/axis2>. Accessed in 10/06/10.
[17] Axiom – XML Object Model. Available in: <http://ws.apache.org/commons/axiom>. Accessed in 10/06/10
[18] Erl, T. SOA Principles of Service Design (The Prentice Hall Service-Oriented Computing Series from Thomas Erl). Prentice Hall PTR. 2007.
[19] Booth, D., et. al., Web Services Architecture. In World Wide Web Consortium (W3C), February 2004.

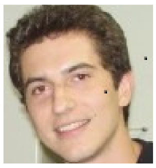
[Online]. Available: http://www.w3.org/TR/ws-arch/. [Accessed: August 2010].

[20] Meijler, D. T., Kruithof, G., Nick, V. B., Top Down Versus Bottom Up in Service-Oriented Integration: An MDA-Based Solution for Minimizing Technology Coupling. In ICSOC, November 24 2006, pp. 484-489. Available at http://www.springerlink.com/content/11462582829728m7/fulltext.pdf.

[21] Andrews, T., et al., Bussiness Process Execution Language for Web Services Verion 1.1, 5 May 2003. [Online]. Available at: http://www.ibm.com/developerworks/library/specification/ ws-bpel/ [Accessed: August 2010].

[22] Fernandes, J.; Lemos, G.: Interactive Television in Brazil: System Software and the Digital Divide. EuroiTV, (2007)

Author Biographies



Gabriel Massote Prado, Poços de Caldas / MG, May,25th, 1987. Graduate in Computer Science at Pontificia Universidade Catolica at Poços de Caldas. Now is attending the Master in Computer Science Program at Federal Universiy – São Carlos / SP, Brazil.

Sergio Donizetti Zorzo , Araras/ SP, October 27th, 1956. Bacharel in Computer Science / Federal University of Sao Carlos. Master in Computer Science at Institute of Computer Science – University of Sao Paulo / campus São Carlos. Ph.D. in electrical engineering at Polytechnic School / University of Sao Paulo. Associate Professor Federal University of Sao Carlos.