Game With Middleware Ginga

Iran Cidade
I.A. Cidade
ialvarenga@id.uff.br
UFF – RJ

ABSTRACT

This article presents a game made with middlwarre Ginga, a multimedia tool from the Nipo-Brasileiro de TV Digital Terrestre (ISDBTB) and ITU-T Recommendation for IPTV services, developed by PUC - RJ and UFPB [1]. This game features an entertainment application to be applied to Digital TV or Personal Computer that has the Ginga application installed.

KEYWORDS

Ginga, NCL, middleware, declarative environment, digital TV, hypermedia

1 Introdution

Digital interactive TV has brought facilities to viewers through middlware like Ginga. The viewer can access the content according to their choice. Middlware like ginga is something innovative and has integration for different media contexts and various applications with digital TV in addition to being free software [3]. Ginga's declarative environment, called Ginga-NCL, is open source and was developed by the laboratory TeleMidia from PUC-Rio in partnership with UFPB. To implement Ginga-NCL as a presentation machine and open source on Linux and Windows platforms, it can be accessed on the Ginga website . The known tools for developing applications with Ginga are NCl composer and NCL eclipse. Ginga is the name of the open Middleware of the Brazilian Digital TV System. The Ginga architecture and facilities were designed to be applied to broadcasting systems and terrestrial broadcasting receivers and IPTV systems [1].

The NCL language (Nested Context Language) [5], the base language of Ginga-NCL, is a structure-based language, an XML application, for authoring interactive multimedia content. A hypermedia document is generally made up of nodes and links. Nodes represent abstractions of the media used in the document in addition to providing additional information, such as information about its presentation.

The links perform spatial or temporal synchronization between the nodes that make up the document. The NCM (Nested Context Model) model has the power and flexibility of a hypermedia document. The NCM extends the definition of nodes into two types, content nodes and composition nodes. A content node carries information about a medium used by the document, while a composition node has a set of content nodes or

other composition nodes and a set of links, being used to give structure and organization to a hypermedia document.

The NCL language treats documents in the NCM definition, where the hypermedia document is formed as an organization of nodes and links. To develop a game in NCL Ginga, the challenge is to place the media contexts with the formatting of a game, and not a multimedia presentation. For this purpose, the game called *Mediaball*, has several penalty shootout screens filmed and archived as videos, to later be inserted into a game context. Each time a video is shown, the penalty taker takes a shot and the *Mediaball* player has three seconds to press a button below the presentation with his guess as to whether the ball will go to the left of the goal, to the right or to the middle of the goal. This game stimulates sensory perception while the player interacts with the buttons, creating the perspective of a media context with the rules of a game. More information about Ginga can be found at
[10].

2 - Architecture and tools

Ginga's architecture as well as its installations are aimed at transmission systems and receivers for terrestrial transmission (over-the-air). But nothing prevents it from being applied to other transport systems such as satellite, cable TV and IPTV [3]. The Brazilian digital TV system has a layered architecture where each layer offers services to the next higher layer and offers services to the lower layer.

Figure 2: GingaNCL architecture [11]

Ginga middleware architecture can be presented in three modules:

• Ginga-CC (Common Core): has services for the presentation machine and execution machine.

• Ginga-NCL (declarative): where declarative applications of the language are developed with XML.

• Ginga-J (procedural): consists of the environment that supports the development of procedural applications

The tools used are composed of NCL-Eclipse and NCL-Composer, where they can be developed both in the language XML [4] using NCL-Eclipse, which uses a graphical and textual language. The NCL Composer architecture is based on a minimal core, which manages the messages exchanged by different plugins, and which can be extended by third-party plugins. Alternative or third-party plugins can be new views (graphic or textual) to aid authoring, or new data generators in a given syntax for transfer to various digital TV systems, or in a given storage format [5]. Currently, there are several plugins developed by third parties to facilitate the creation of interactive domain-specific applications.

NCL Composer is a very useful tool for visualizing media nodes, their relationships, connectors and descriptors. As illustrated in figure 1, the *Mediaball* game has a media node for presentation and from a descriptor that relates each stored video to an event related to a connector, which in this case is a button pressed by the player, which at the time of media presentation, you will press a button corresponding to the displayed image. NCL Eclipse is another tool used to desenvolver o código em XML [4].

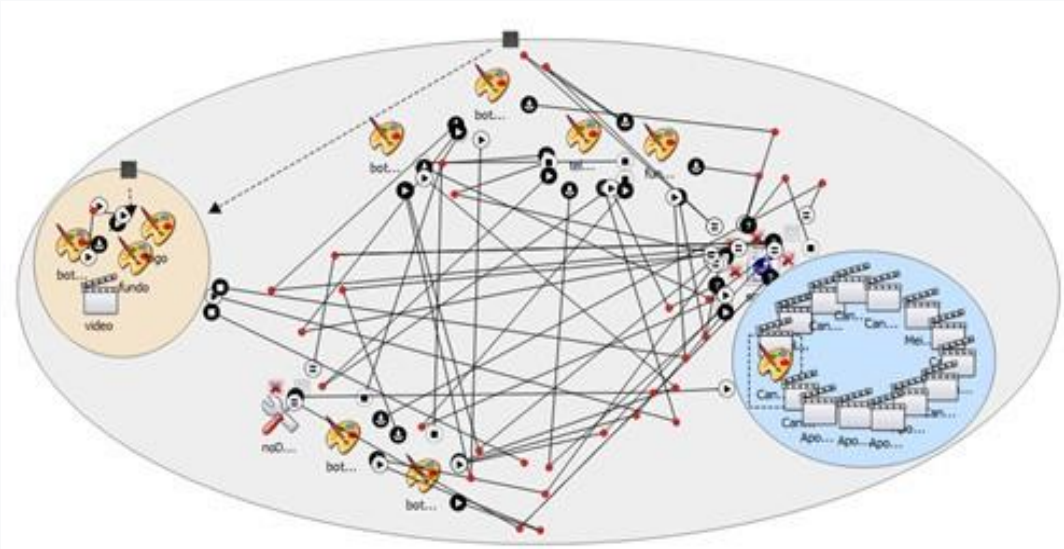

Figure 3: Mediaball diagram generated by NCL Composer.

The NCL code is written in XML (Extensible Markup Language), as XML is not a language used for a specific application, it is only used as a metalanguage to create a markup language. Declarative languages are easier to use and do not require in-depth knowledge of programming logic like procedural languages. You don't need so many lines of code, just statements of tasks to be performed. Hypermedia documents presented

in Ginga must have spatial and temporal synchronization to present their media content, which is why the NCL standard was adopted for Ginga. Animations, graphics collisions, text manipulation, all the mathematics involved in the process of digital TV systems are carried out by a procedural language. The Ginga-J procedural environment adopts Java as its procedural language [6].

3 Game Mediaball

Mediaball was written using the NCL-Eclipse tool to create connectors, media nodes and context. The node and event diagrams, as well as the technical layout layout can be viewed in NCL Composer. What makes it possible to create game dynamics is the Lua[9] script language embedded in the NCl code. The entire game engine is written in NCLua language, which allows you to manipulate media nodes together with the NCL language, hence defined as NCLua. This language allows the creation of media events the <region> tag defines the regions where the media will be presented within the context. Media Nodes<media>: These can be videos, texts, scripts that must be presented. Descriptors <descriptor>: They link the media nodes to the region of the screen that will be displayed, they also define how it will be presented. Region<region>: Indicate the position of the screen where the media will be presented.

Figure 4: Programming structure in the NCL language.

The <descriptor> descriptors, see figure 4, are also important in the NCL language as they describe how media events will be presented on the screen, as well as their characteristics, navigation, display time as referenced in the NCL documentation [10].

The content of a hypermedia document is represented by elements called media, the media represents each node of the document informing the descriptor to which it is related. According to the NCM [6] document, a medium must be within a composition node called context. Nested Context Model (NCM), extends the concepts by increasing the power and flexibility of a hypermedia document. NCM extends the definition of nodes into two types, content nodes and composition nodes. A content node brings information about a medium used by the document, while a composition node has a set of content nodes and/or other composition nodes and a set of links, being used to give structure and organization to a hypermedia document, i.e. can be well illustrated in the NCL Composer diagram, see figure 3.

The logic of this game is carried out by a script written in Lua language that will select the image to be displayed. An important feature of the NCL language is that it allows hybrid applications with media events associated with a procedural language. In this case, the *lua* script, in addition to sorting the events, will also post them. The *lua* language in this case is also written with event posting syntax, so it was easily

incorporated into NCL. LUA is a procedural programming language, dynamically typed, with automatic memory management and automatic garbage collection (“garbage collector”).

Figure 5: Mediaball Game Screen Generated with NCL Eclipse .

The game consists of drawing pre-recorded images in mp4 format and then inserting them into the media context using NCL tools. In this game screen, the penalty taker will approach the ball and the Mediaball player will have 3 seconds to press a button and decide which way the ball will go. At the end of the screens, a total of 10, the player will check his hits. The interesting thing about this game is that the media content can be easily manipulated following the middlware interactivity concept.

4 Conclusion

The purpose of this work is to awaken the possibility of using these applications by the academic community researching middlware applications and digital entrepreneurship in the games industry. The middlware went through an approval process to be used on TVs in Brazil [7].

Ginga's architecture also favors the creation of applications for digital TV, the simple change from the analogue to digital system allowed much more than an improvement in image and sound quality. Digital TV allowed the flexibility to expand the system's functions through applications built on the basis of a standard system of

references. These applications are computer programs resident in a receiving device or originating from data multiplexed with the main audio and main video of a television program [2].

Both controlling access to content and its protection are the facilities presented, educational services (T-learning), health services (T-health), even the distribution of electronic games [2, 8]. This demand involves more computing professionals, application developers can also cope with this growing sphere of the television market.

Few scientific or academic works are developed in this area, it is important to think that the telecommunications industry is quite significant [12]. The opportunity to create funding for research into digital TV is very interesting, but little publicized.

Programmers and developers have more of this area of activity in computational sciences and related areas. Taking advantage of academic research to develop another Brazilian product is very motivating and guarantees growth in the telecommunications sector for Brazil

REFERENCES

[1] ABNT. 2015. Televisão digital terrestre — Codificação de dados e especificações de transmissão para radiofusão digital Parte 2: Ginga-NCL para receptores fixos e móveis — Linguagem de aplicação XML para codificação de aplicações. Disponível em <https://www.abntcatalogo.com.br/norma.aspx?ID=329288>, Acesso em : 16 jul. 2023.

[2] Marco Aurelio Migliorini Antunes. 2015. Desenvolvimento de um protótipo de aplicações interativas para TV Digital no Middleware Ginga com objetos NCL. (2015).

[3] Eduardo Cruz Araújo and Luiz Fernando Gomes Soares. 2014. Designing iDTV Applications through Interactive Storyboards. In *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*. 5–12.

[4] Philip Bohannon, Juliana Freire, Prasan Roy, and Jérôme Siméon. 2002. From XML schema to relations: A cost-based approach to XML storage. In *Proceedings 18th International Conference on Data Engineering*. IEEE, 64–75.

[5] Guido Lemos de Souza Filho, Luiz Eduardo Cunha Leite, and Carlos Eduardo Coelho Freire Batista. 2007. Ginga-J: the procedural middleware for the Brazilian digital TV system. *Journal of the Brazilian Computer Society* 13, 1 (2007), 47–56.

[6] Sayonara Leal. 2019. Controvérsias na padronização do middleware de interatividade da TV digital brasileira: democracia técnica à prova do dispositivo de normatização da inovação Ginga. Sociedade e Estado 34 (2019), 49–83.

[7] João Paulo Bulhões Nogueira Martins et al. 2020. A nova televisão: estratégias, desafios e tendências da distribuição digital não linear no mercado brasileiro. (2020).

[8] PUC-Rio. 2006. Manual de referência de Lua 5.1. Disponível em http://www.lua.org/manual/5.1/pt/.

[9] PUC-Rio. 2007. Site oficial sobre o Ginga.Disponínvel em <http://www.ginganc l.org.br/pt-br/sobre>.

[10] Luiz Fernando Gomes Soares, Marcio Ferreira Moreno, Carlos de Salles Soares Neto, and Marcelo Ferreira Moreno. 2010. Ginga-NCL: Declarative middleware for multimedia IPTV services. *IEEE Communications Magazine* 48, 6 (2010), 74–81

[11] Nara Idelfonso Souto and Cláudio Márcio Campos de Mendonça. 2013. TV DIGITAL NO BRASIL: estudos científicos x aplicação no mercado. *1984-4204-Digital Object Identifier (DOI): http://dx. doi. org/10.21714/raunp.* 5, 2 (2013), 99–108