GRAPHICS ARCHITECTURE FOR MULTIMEDIA NON-DESKTOP DEVICES: A STUDY OF DIGITAL TV RECEIVERS

Pablo Cesar*

Abstract. *Non-desktop devices, in particular entertainment consumer products, are evolving into advanced interactive multimedia terminals, generating a new set of requirements for their user interface software architecture. Thus, traditional approaches used in desktop computers have to be reconsidered. This thesis proposes a graphics architecture model and presents the experiences of applying it for the design and development of a configurable digital television receiver. The implementation can run advanced multimedia services such as 3D games, Java applications, and XML based documents.*

1. Introduction

The starting assumption of this research, depicted in Figure 1, can be expressed as: currently, a number of multimedia terminals accesses through a variety of networks to a diversity of content and services. These multimedia terminals include both desktop computers and non-desktop devices. [14]

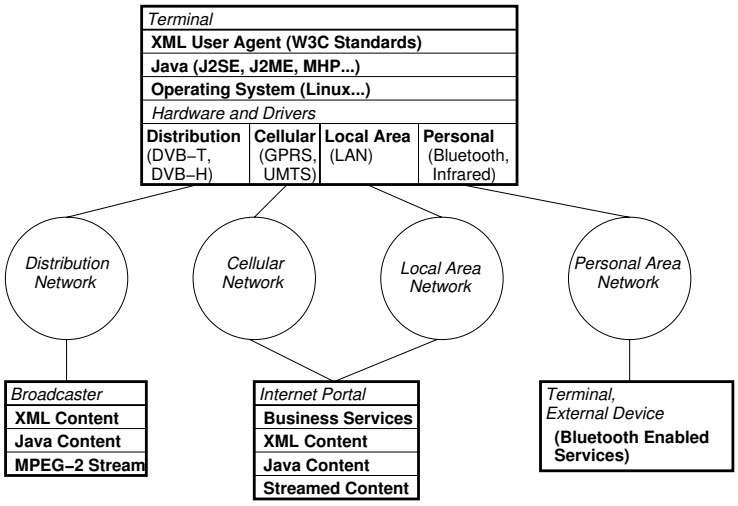


Figure 1. Device Chaos at the Turn of the Century.

This thesis categorises interactive multimedia devices based on their purpose (i.e., user expectations). First, these devices fall into the broader definition of information access systems, but in contraposi-

*Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology, P.O. Box 5400 FIN-02015 HUT, Espoo, Finland, email: pcesar@tml.hut.fi

tion to personal computers, non-desktop devices are rarely used for development. Moreover, personal computers are multipurpose platforms, while non-desktop devices can be classified as [7]: Information management systems such as personal digital assistants, person to person communication systems such as mobile phones, and entertainment systems such as DVD players, game consoles, and digital television receivers.

Because a doctoral thesis should be kept within certain boundaries, studying all kind of non-desktop devices (e.g., wereable computers, smart appliances) would be a suicidal approach. Hence, based on the platform's purpose, this thesis concentrates on digital television receivers as a representative example of entertainment oriented systems or media stations [4]. The topic is narrowed even more to their graphics architecture. Specific requirements of such devices include:

- **User interaction:** media stations usually lack of a pointing device.
- **Scene metaphor:** the screen is considered as a scene composed of different multimedia objects.
- **Presentation:** different multimedia objects such as video, 2D, and 3D graphics should be seamlessly integrated.
- **Windowing System:** because media stations show one scene at a time, a scene manager would be more appropriate.
- **Levels of abstraction:** a layered architecture is preferable, so developers can implement at any level depending on their goals and expertise area. Some examples include procedural languages (e.g., Java) and declarative languages such as eXtensible Markup Language (XML).

2. Research Problem and Background

The research problem is the definition of a graphics architecture for digital television receivers. Thus, the background includes topics such as graphics architectures (i.e., device manufacturer), content authoring formats (i.e., service provider), and standardisation procedures (i.e., independent consortium).

Myers proposed a graphics architecture for desktop devices, layered into Windowing System, Toolkit, and High Level Tools [11]. This architecture divides the screen into different regions, or windows, and implies the performance of different activities simultaneously (e.g., surfing the Web and writing a document). Since media stations are intended to visualise one context at a time (e.g., watching television or playing a game), the concept of windows becomes unnecessary. Closely related to the underlying architecture, current systems are built around the Windows, Icons, Menus, and Pointing (WIMP) paradigm. This thesis argues that media stations do not follow the WIMP paradigm because of its content (i.e., video and advanced user interfaces) and its temporal based nature. The argumentation is based on Chorianopoulos [4] and SIGGRAPH'90 workshop on no-WIMP interfaces [6] findings.

In order to simplify the service developer task and increase the amount of available content, the adoption of standardised content formats is needed. These content formats can be categorised as procedural and declarative. Procedural languages are those, in which the developer codes how to solve a problem, while using declaratives ones she only states what has to be solved. Hence, because declarative languages have a higher level of abstraction, their expressional power (i.e., ceiling) and

difficulty to learn (i.e., threshold) are lower. Moreover, there are two kinds of procedural languages: intermediate (e.g., C Language) and interpreted (e.g., Java). Procedural - interpreted languages use the virtual machine concept (i.e., own computer abstraction), therefore they are more interoperable and safer to distribute than intermediate ones.

Currently, the main concern regarding digital television is to assure a smooth transition from its analogue counterpart, for which the definition on time of a viable, open, and platform independent middleware (i.e., horizontal market) standard is a key issue. Such standard should, as well, take into account that the transition will follow gradual steps and that different user groups will have different demands. With this goals in mind, different regional consortium have proposed their own standards, such as Digital Video Broadcasting (DVB) defining Multimedia Home Platform (MHP). In order to provide a worldwide solution, DVB has published Globally Executable MHP (GEM) [5]. GEM is a valid starting point, which takes into account the american and japanese initiatives, but only considers a procedural interpreted language, DVB - Java (DVB-J). Hence, the declarative environment and advanced functionality (e.g., 3D graphics) remain to be standardised.

In conclusion, this thesis studies the graphics architecture for digital television receivers and proposes a model taking into account its specific requirements. Then, the model is validated by analysing its suitability for other media stations and by implementing, as a proof of concept, a configurable digital television receiver called Ubik. Moreover, Ubik permits to propose extensions to the GEM standard and make recommendations to manufacturers.

3. Methodology

This thesis was carried out in a ”step-by-step” approach. First, the author analysed the state of the art in digital television environments, reviewed the related literature, and examined existing standards and solutions (e.g., Tivo or WebTV). Then, when some missing parts of the puzzle were identified, the author studied different alternatives and proposed a solution, following the Concept Implementation research method. In order to review its progress, the intermediate results were submitted to International Conferences. Finally, all these results were integrated and a system was constructed as the final experiment of this thesis and presented as a Journal paper [3].

This thesis will include, apart from a 70 pages introductory part currently being written, a compendium of nine publications. In order to provide a clearer understanding of the journey followed during the research, Figure 2 depicts the relationship between the publications and the milestones. The first publication, P1 [17] proposed a system software architecture for digital television receivers, while the second, P2 [15], described a system integration model capable of running Java based services. Even though they might not be completely relevant for this thesis, they highlight the basic starting problems studied in the other publications. First, a digital television oriented graphical user interface toolkit (e.g., taking into account remote control input), presented in P3 [2], is needed for software developers. Second, a declarative profile intended for simple applications development (e.g., information services) was proposed in P4 [16] and P5 [10]. Because of the temporal nature of digital television, an implementation of Synchronized Multimedia Integration Language (SMIL) was reported in P4 and its potential use cases were analysed in P5. The major problem of SMIL language is its limited interaction capabilities, thus P7 [9] proposed a XML Compound Documents profile (SMIL+XForms), in which XForms permits developers to define advanced user interaction. Third, in order to study current digital television limitations, the author contributed in the implementation of Java services delivered to commercial receivers, which results were presented in P6 [8]. Finally, once

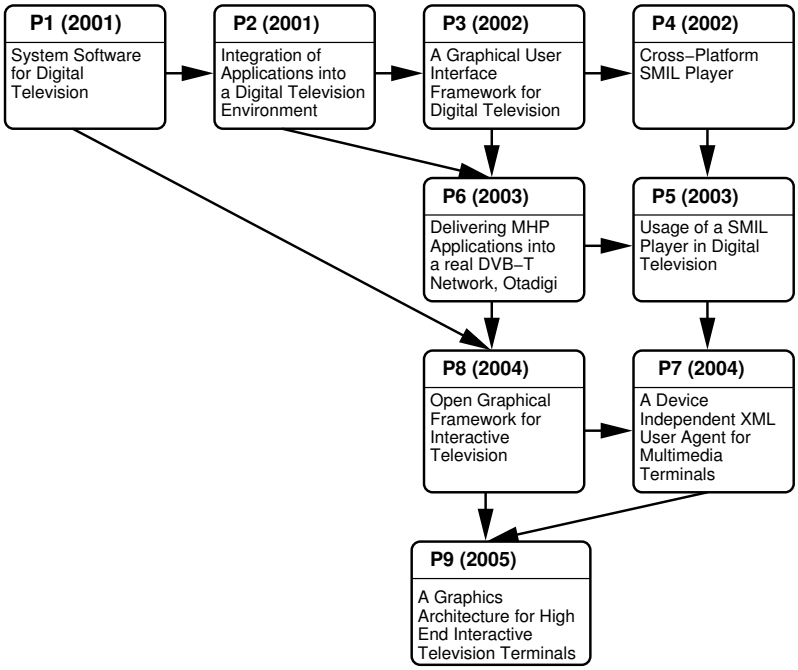


Figure 2. Relation of Publications in the Thesis and Milestones.

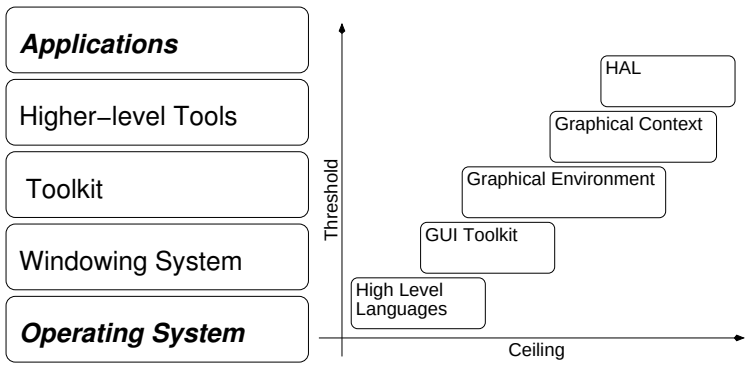
the author had experimented with the practical issues, a layered model for digital television receivers was proposed in P8 [1], which was extended with a more comprehensive analysis and evaluation in P9 [3].

4. Results - Contributions

This thesis proposes a graphics architecture model for digital television based on the requirements presented in Sections 1. and 2. This model is based on Myers' architecture [11], but replaces the Windowing System by a scene manager and explicitly integrates different multimedia objects (e.g., video and 3D graphics), as depicted in Figure 3. The proposed model is layered and takes into account the threshold and ceiling, introduced by Myers in [12], of each layer:

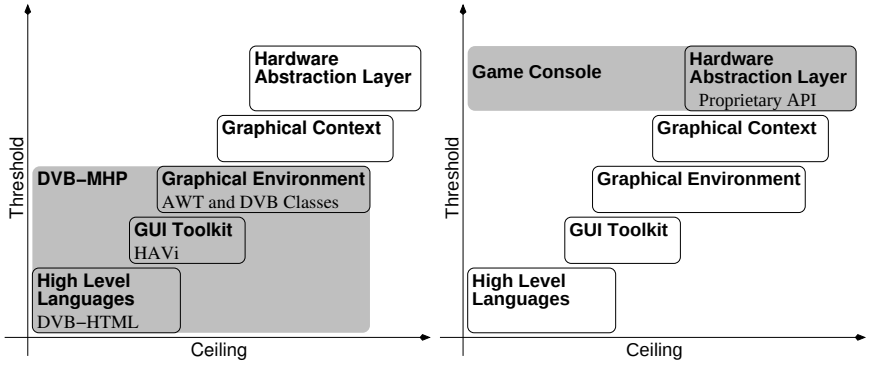
- **Hardware Abstraction Layer (HAL):** renders the final graphics output.
- **Graphical context:** a cross-platform abstraction of the rendering region or canvas as defined by Olsen [13]. In addition it, provides graphics primitives for multimedia objects.
- **Graphical Environment:** controls different graphical contexts.
- **GUI toolkit:** a set of "ready-made" user interface widgets.
- **High Level Languages (HLL):** to develop simple services (e.g., XML languages).

The model can be generalised for other media stations such as game consoles. An MHP compliant digital television receiver implements the Graphical Environment and GUI Toolkit as DVB-J, while



(a) (b)

Figure 3. Comparison beetween (a) Myers' [11] and (b) Proposed Graphics Architecture (Note the reversed order of the layers because of the threshold and ceiling measures).

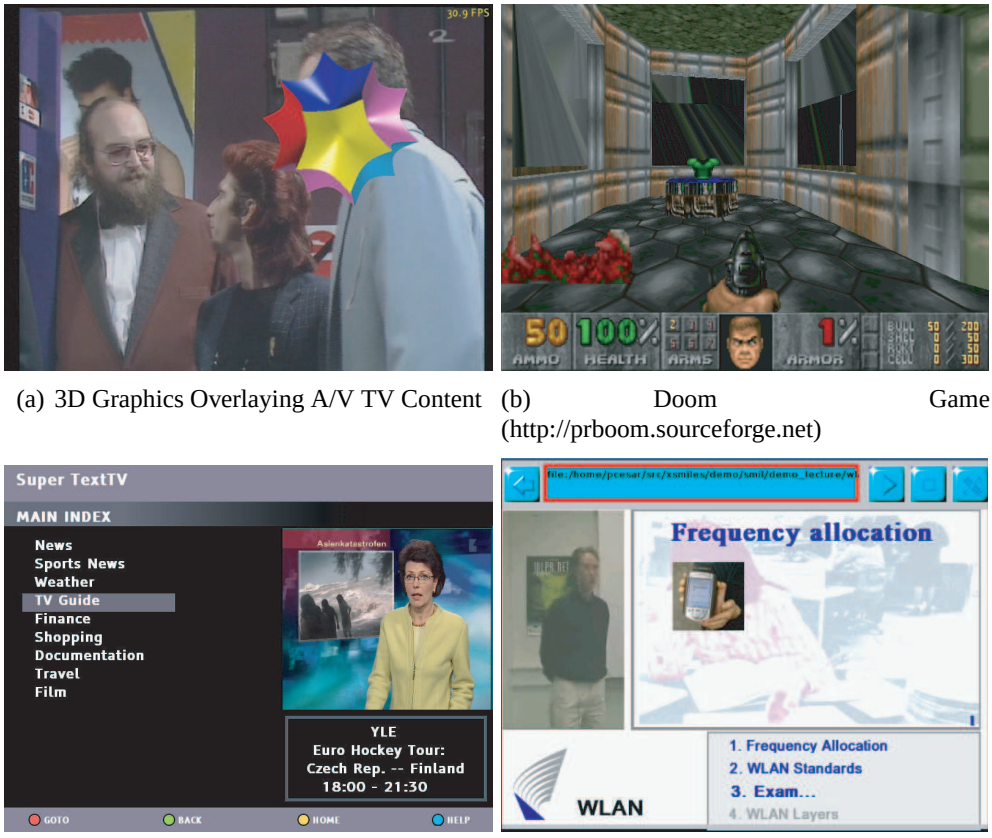


(a) Digital Television. (b) Game Console.

Figure 4. Different Configurations of the Model for Media Stations Examples.

a game console provides those layers closer to the hardware and can support, for example, high level languages to get on-line help for a particular game [1]. These configurations are depicted in Figure 4.

Moreover, the model is validated by the development of a reference implementation of a digital television receiver called Ubik[1]. Because it is not the same to implement a 3D game than a information service, the layers of the model are ordered based on their threshold and ceiling. Hence, a developer can decide which of the layers is more appropriate for her goals. For example, in Ubik, developers can provide 3D graphics objects overlaying the television content (i.e., Hardware Abstraction layer) [3], games using the Graphical Context layer [3], Java based services (i.e., GUI Toolkit layer) [1], and World Wide Web browsing (i.e., High Level Languages layer) [9], as depicted in Figure 5. Regarding its performance, Ubik consumes less than 32MB to run DVB-J applications over the TV content, a common hardware configuration for limited resources devices. However, a more comprehensive analysis can be found in [3].



(a) 3D Graphics Overlaying A/V TV Content (b) Doom Game (http://prboom.sourceforge.net)

(c) DVB-J Based Teletext (d) E-learning Portal.

Figure 5. Services Running in Ubik.

Finally, Ubik permits to study the evolution of digital television receivers, to propose extensions to current standardisation initiative, and to make recommendations for manufacturers. Figure 6 shows the typical configuration of a digital television receivers. It includes drivers for network accessibility, native support for multimedia objects, a procedural language intended for complex applications (i.e.,

[1]Publication P9 [3] provides a description of the specific packages used, a design rationale, and a comprenhesive analysis of the system

high ceiling and threshold) in the form of DVB-J, and declarative support for simple services (i.e., low ceiling and threshold) by integrating a XML user agent. Moreover, the software architecture is divided into two main groups: broadcast and interactive television. These groups can be further divided into profiles as specified by MHP. At the procedural level, this thesis proposes a new profile, High End Interactive, in which 3D graphics support is included. In addition, it recommends to implement it as a thin layer wrapping OpenGL for Embedding Systems functionality. At the declarative layer, this thesis proposes two different profiles, instead of the MHP solution, DVB- Hyper-text Markup Language (DVB-HTML). First, eXtensible Hyer-Text Markup Language (XHTML) + Cascading Style Sheets (CSS) for defining the structure and look of limited interactive applications (e.g., Teletext). Second, SMIL + XForms for more complex applications (e.g., E-learning portal). Finally, even though current research on multimedia concentrates on MPEG-4, this thesis considers its size, unmaturity, and threshold as too strong drawbacks.

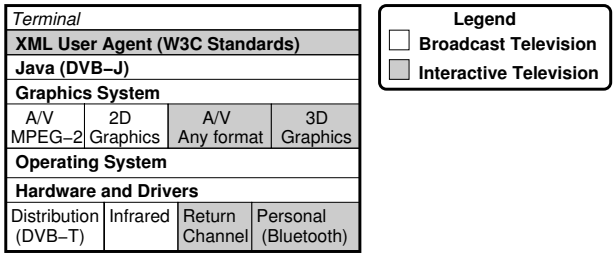


Figure 6. Configuration of Digital Television Receivers.

In conclusion, the contributions of this thesis can be divided into theoretical, standardisation, and commercial value. First, theoretically, it defines a graphics architecture model, taking into account the special restrictions of media stations. Second, it analyses, in depth, digital television standards and proposes extensions such as 3D graphics support. Finally, commercially, this thesis studies the evolution of digital television receivers and provides a reference implementation of all the covered topics. The implementation can enhance television content by overlaying 2D and 3D graphics, running Java services, and displaying XML based documents.

Acknowledgments

The author would like to thank Nokia Oyj Foundation, HeCSE, and the partners of BROCOM project for providing financial support. In addition, I would like to thank my supervisor, Professor Petri Vuorimaa, for his patience, trust, and constant support.

References

[1] P. Cesar, J. Vierinen, and P. Vuorimaa. Open graphical framework for interactive tv. In *Proceedings of the 5th International Symposium on Multimedia Software Engineering*, pages 21–28. IEEE, 2003.

[2] P. Cesar and P. Vuorimaa. A graphical user interface framework for digital television. In *Proceedings of the 10th International Conference in Central Europe on Computer Graphics, Visualisation and Computer Vision*, pages 1–4. WSCG, 2002.

[3] P. Cesar, P. Vuorimaa, and J. Vierinen. A graphics architecture for high end interactive television terminals. *Submitted to ACM Transactions on Multimedia Computing, Communications, and Applications*, 2005.

[4] Konstantinos Chorianpoulos. *Virtual Television Channels: conceptual model, user interface design and affective usability evaluation*. PhD thesis, Athens University of Economics and Business, May 2004.

[5] DVB. Digital video broadcasting (dvb); globally executable mhp (gem). ETSI TS 102 819 v1.2.1, May 2004.

[6] M. Green and R. Jacob. Siggraph '90 workshop report: Software architectures and metaphors for non-wimp user interfaces. *ACM SIGGRAPH Computer Graphics*, 25(3):229–235, July 1991.

[7] Uwe Hansman, Lothar Merk, Martin S. Nicklous, and Thomas Stober. *Pervasive Computing Handbook*. Springer-Verlang, Berlin (Germany), 2001.

[8] C. Herrero, P. Cesar, C. Herrero, and P. Vuorimaa. Delivering mhp applications into a real dvb-t network, otadigi. In *Proceedings of the 6th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services*, pages 231–234. IEEE, 2003.

[9] M. Honkala, P. Cesar, and P. Vuorimaa. A device independent xml user agent for multimedia terminals. In *Proceedings of the 6th International Symposium on Multimedia Software Engineering*, pages 116–123. IEEE, 2004.

[10] J. L. Lamadon, P. Cesar, C. Herrero, and P. Vuorimaa. Usages of a smil player in a digital television broadcast system. In *Proceedings of the 7th International Conference on Internet and Multimedia Systems and Applications*, pages 579–584. IASTED, 2003.

[11] Brad Myers. Graphical user interface programming. In Allen B. Tucker, editor, *CRC Handbook of Computer Science and Engineering*, chapter 48. CRC Press, Inc., Boca Raton (FL), second edition, 2004.

[12] Brad Myers, Scott E. Hudson, and Randy Pausch. Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction*, 7(1):3–28, March 2000.

[13] D. R. Olsen. *Developing User Interfaces*. Morgan Kaufmann, San Francisco (CA), 1998.

[14] D. R. Olsen. Interacting in chaos. *ACM Interactions*, 6(5):42–54, November 1999.

[15] C. Peng, P. Cesar, and P. Vuorimaa. Integration of applications into a digital television environment. In *Proceedings of the 7th International Conference on Distributed Multimedia Systems*, pages 266–272. Knowledge Institute System, 2001.

[16] K. Pihkala, P. Cesar, and P. Vuorimaa. Cross-platform smil player. In *Proceedings of the 1st International Conference on Communications, Internet and Information Technology*, pages 48–53. IASTED, 2002.

[17] G. Sivaraman, P. Cesar, and P. Vuorimaa. System software for digital television applications. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 784–787. IEEE, 2001.