A Software Architecture for Adapting Virtual Reality Content to Mobile Devices

Conference Paper · June 2010

DOI: 10.1007/978-3-642-13789-1_16 · Source: DBLP

6 authors, including:

José Barbosa Dias Jr
Instituto Federal de Educação Ciência e Tecnologia do Paraná (IFPR)

Marlus Dias Silva
Universidade Federal de Uberlândia (UFU)

Luiz Fernando Braga Lopes
Faculdade Cidade Verde

Eduardo Damasceno
Federal University of Technology - Paraná/Brazil (UTFPR)

A Software Architecture for Adapating Virtual Reality Content to Mobile Devices

José Barbosa Dias Júnior
Instituto Federal de Educação, Ciência e Tecnologia de Goiás – Campus de Goiânia – Departamento IV
Goiânia - Brasil
juniorddos@yahoo.com.br

Alexandre Cardoso
Programa de Pós Graduação de Engenharia Elétrica da Universidade Federal de Uberlândia – UFU – MG
Uberlândia - Brasil
alexandre@ufu.br

Marlus Dias Silva
Programa de Pós Graduação de Engenharia Elétrica da Universidade Federal de Uberlândia – UFU – MG
Uberlândia - Brasil
marlusdias@yahoo.com.br

Edgard Afonso Lamounier Júnior
Programa de Pós Graduação de Engenharia Elétrica da Universidade Federal de Uberlândia – UFU – MG
Uberlândia - Brasil
lamounier@ufu.br

***Abstract*— This paper presents a proposal of software architecture to assist in the development of Virtual Reality applications for mobile devices based on context adaptation techniques. The aim of this solution is to offer support to Virtual Reality context appropriateness to be used by different mobile devices in accordance with their individual limitations. This adaptation is dynamically performed by the proposed solution without the need for user interaction, providing Virtual Reality content portability for mobile applications**

Keywords: virtual reality; mobile devices; adaptation 3D

I. INTRODUCTION

The evolutions in portable computer devices technology, and the advances in wireless communication have led to the emergence of a new paradigm [1]: Mobile Computing (MC).

The constant technological innovations in this area have permitted the development of a great variety of portable devices that can transport the working environment beyond "office building walls", leading to greater freedom between users and computational systems.

The emergence of different types of equipment such as mobile cell phones with Wireless Application Protocol (WAP), notebooks, palmtops, among others, has led to an increased demand for new services and applications in the most diverse areas.

According to [1], the evolution of MC has promoted advances in the development of new hardware and software technologies, opening a range of new applications based on wireless communication resources.

Among those, applications for Virtual Reality (VR) and/or Mixed Reality (MR) in the most diverse areas such as education, entertainment, visualization, etc., are in special evidence.

The portability provided by mobile devices, together with VR's more natural interaction, permits the creation of a Mobile Virtual Environment favorable to the exchange of experiences, information, images, etc, among people, in a more attractive and motivating way for the user.

However, problems specifically related to mobile devices limitations, which continue to represent a 'bottleneck' for the creation of VR mobile applications, still exist [2].

The main obstacle to be transposed in the development of applications for mobile devices (MDs) is their heterogeneity, characterized by limitations in terms of processing, memory, and battery capacity, communication bandwidth, as well as the great variety of existing software (Windows, Symbian, Palm OS, etc.) [3 and 4].

Due to those different features, it would be necessary to develop specific VR content versions for these different environments, in accordance with the characteristics of each device. However, the creation of different versions might not be viable and might not reach the "human" condition, because of the number of platforms and devices available.

Within this scenario, the development of VR applications for mobile devices becomes a challenge, especially for heterogeneous devices.

Therefore, this work has the objective of proposing an autonomous content customizing architecture for mobile devices environments, i.e., without the intervention of the user, by the means of a software infrastructure, facilitating the development of VR applications for mobile devices.

II. ADAPTATION

According to the concept presented by [5], adaptation is the capacity that an application has to adapt itself to alterations in the environment circumstances in which it is executed.

Two factors have contributed to the increased interest in the development of adaptable software [6]. First, there is the MC's paradigm, by which an application must have the capacity to adapt to several environments and MDs. The second is the development of Autonomous Computing, whose objective is to increase the number of systems that

978-0-7695-4108-2/10 $26.00 © 2010 IEEE
DOI 10.1109/CIT.2010.346

have self-management capabilities, by using all their technological infrastructure potential, reducing human administration over them.

An application may be considered adaptable if it is capable of automatically changing its behavior according to its context [7].

So that VR mobile applications may be used in different devices, the content must be carefully selected and adapted to the restrictive conditions of each destination device in terms of display size, quantity of colors, available memory, etc [5-7].

Therefore, the same content directed to a microcomputer should not be sent to a mobile device such as a cell phone or a *Personal Digital Assistant* (PDA), for example. To do so, both a reduction in the amount of information as well as a change in the way data is presented must occur.

This process is presented in the working structure of this study (4.1).

III. Related Work

Projects related to the use of VR and Augmented Reality (AR) in mobile devices have started to be explored in the last years, and there are some important factors to be resolved so that these may offer better service quality and quantity to users, such as: bandwidth limitation, processing, memory, screen size, etc. [5 – 7].

Among those projects, works in areas such as Localization and Navigation [8], Education [9], Visualization and Navigation [10 and 11] are representative.

The project reported in [8], uses VR and MR for navigation. Although they are used in pre-defined different devices, it does not possess criteria for adapting content according to the devices characteristics, which also happens in most of the works reported.

According to [9], although Virtran was originally developed for cell phones, it can also be used by PDAs or Pocket PCs. However, this solution does not provide support for content adaptation, or definition of the devices' profile, as the solution proposed here.

Museum Virtual Guide permits certain personalization as, for example, the routes according to users choices [10].

However, the criteria to carry out an adaptation according to users requirements would be very complex, due to the amount of information necessary (age of the groups, preferences, groups with disabilities, etc) [10 and 12].

The personalization used by [10] is close to user preferences customization, but it tends more to the configuration of preferences reported in [12 and 13].

SignPost [11] permits differentiated visualization according to user preferences. These data are pre-defined by the application without, however, the possibility of adaptation.

IV. Proposed Architecture

The proposed architecture uses an application separated into layers. This solution permits the inclusion and the exclusion of new functionalities, processes or layers with minimal impact on the architecture. Besides facilitating and making the development process more efficient (by reusing the code), it also permits the solution to be easily improved [14].

Figure 1 shows the Adaptation Server, composed by five layers: *Connection, Group, Adaptation Management, Profile Factory* and *Persistence*.

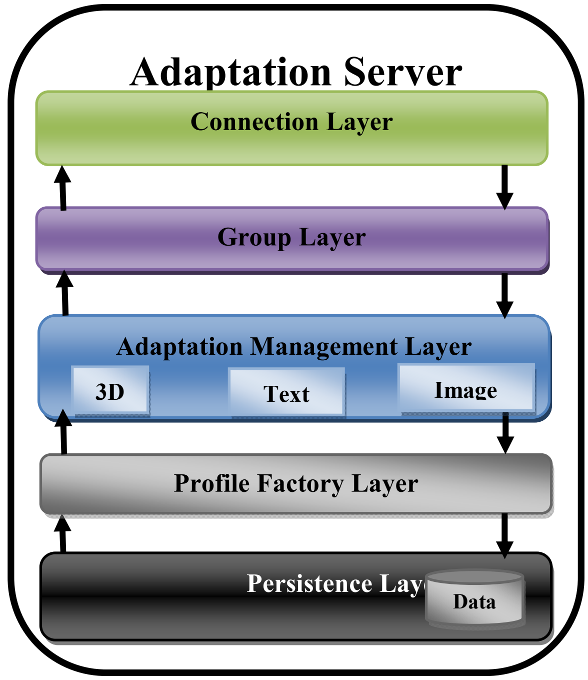

Figure 1. Proposed Architecture.

The activities executed by the layers belonging to the Adaptation Server are described as follows:

1st. *Persistence Layer*: It is responsible for storing information about devices as well as 3D content, guaranteeing their integrity.

2nd. *Profile Factory Layer*: It uses the information from the previous layer to generate the devices' profiles, in a pre-defined way. These profiles are a set of specific characteristics belonging to the devices that are used to identify them. They are generated, managed and stored by this layer. To generate these profiles, a set of pre-defined rules are used, adapted from [14], obtained from studies carried out on MDs databases, proposed by W3C [15] *Composite Capability/Preference Profiles* (CC/PP) and in the UAProf vocabulary developed by the WAP Forum group [16], which contains information on MDs characteristics.

3rd. *Adaptation Management Layer*: This layer determines the strategies used to carry out VR content adaptation, using content adaptation techniques previously described in item 2.

To better structure this task, it possesses three internal modules, each of them responsible for specific activities: control and customization of 3D objects (3D Module), images (Image Module) and text (Text Module), using the database from the Profile Factory and Connection Layers. In possession of these data, it executes the content adaptation process, permitting that the same VR content may be used by different devices.

4th. *Group Layer*: It aims at grouping mobile clients with common characteristics to facilitate their management and control during information exchange between server and mobile clients, and so reducing the processing power necessary for managing this task and guaranteeing the integrity of the process.

5th. *Connection Layer*: It is the architecture's last layer, which is responsible for receiving, storing, managing and keeping mobile clients connections references (MDs participating in the application). It also carries out the mobile clients identification process in execution time using the information received from the Group Layer.

The solution developed uses the storage of objects that represent the target environment (3D content, technical information about devices, manufacturers, components, etc) in a database, and so enabling its instantiation in execution time without depending on a specific mobile client model. To do so, it uses a mobile client abstract model, which is the result of the information generated by the Profile Factory Layer.

This process is different from those described in other works in item 3, where objects were normally represented in the code itself, or individually instantiated for each device.

The application developed here is based on the Client-Server model used by [8], [9] and [11], being composed by a Mobile Client, which is a multi-platform client for mobile devices, and by an Adaptation Server, the software architecture responsible for customizing content for portable equipment as illustrated in Figure 1.

A. *Working Structure*

The architecture working structure, as described before, is based on the Client-Server architecture in which all processing related to adaptation is carried out by the Adaptation Server in order to keep its independence in relation to mobile clients.

The mobile client is constituted by the mobile application and the Connection Layer, responsible for performing the request and exchange of messages during the communication between client and server.

The Adaptation Server uses information in the request message coming from clients to identify and group them according to their characteristics. This process is carried out by the Group Layer, while 3D content customization is carried out by the Adaptation Management Layer, in accordance to the Mobile Client profile.

Figure 2 illustrates the proposed general structure.

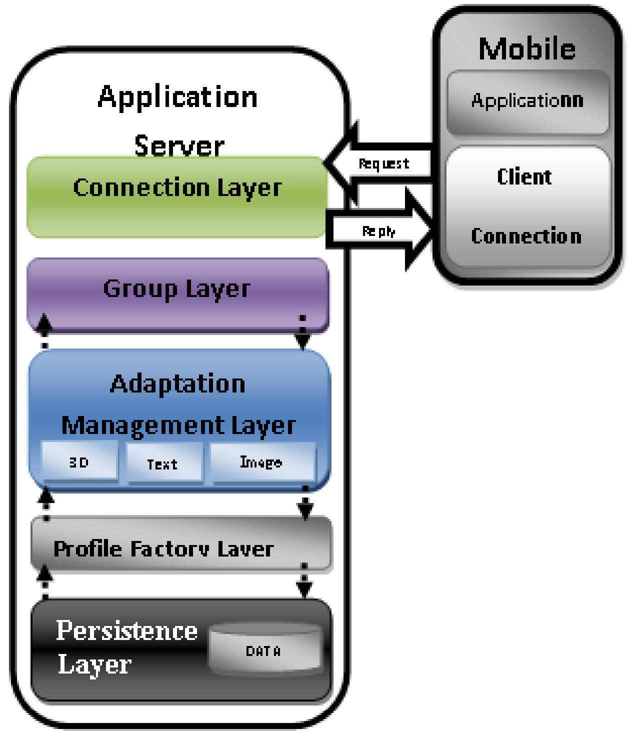

Figure 2. General Structure.

B. *Mobile Client*

The Mobile Client communicates with the server by exchanging request/reply messages. It is responsible for identifying mobile devices, collecting information about the characteristics of each device and sending them to the server in a dynamic and autonomous way. This information is used by the server to help defining the mobile device's profile.

The mobile client is also responsible for VR content requests, which are carried out by the Client Connection Layer, as we can see on Figure 3.

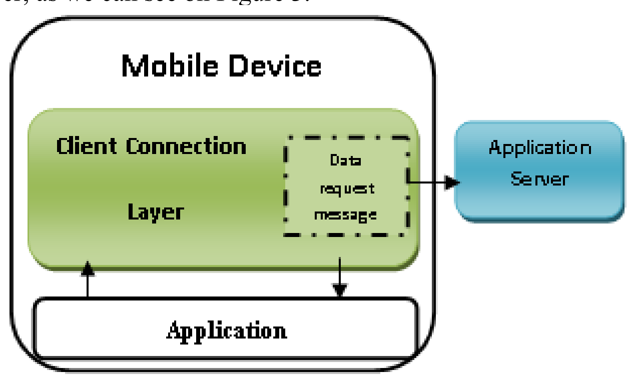

Figure 3. Mobile Client Working Structure.

The Client Connection Layer is also responsible for all the communication performed with the application server. It receives customized information coming from the server and controls data exchange between them.

Data processing is totally executed in the server as not to overload the mobile devices, which have limited processing capacity. The Adaptation Server working structure is detailed in the next item.

C. Adaptation Server

The layers present in the Adaptation Server are generally responsible for receiving data requests coming in from mobile clients, as well as their validation. With these data, the server carries out the identification of participating devices, which allows for their classification according to pre-defined profiles stored in the Persistence Layer.

Identification, requests control and all other activities necessary for the performance of communication with mobile clients are executed by the Connection Layer.

Mobile clients are divided into groups created and managed according to their common characteristics. This process reduces the "quantity" of devices managed by the server, facilitating the application of rules and controlling the alterations in the state of these devices, which improves management and reduces the structure complexity, as well as the processing necessary for executing this step.

Most part of mobile clients management is carried out in groups, but in some cases there is a need to control specific events (alterations, requests, etc.). This process is also executed by the Group Layer, which controls the specificities necessary to keep the integrity of the structure.

The adaptation Server working structure and the exchange of messages among its layers are illustrated in Figure 4.

The Profile Factory Layer receives messages with information from devices and with VR content from the Persistence Layer, and uses these data to generate profiles according to each device individual restrictions.

VR content adaptation policies are applied by the Adaptation Management Layer, which uses a set of rules and the information on the pre-defined profiles to carry out content customization in accordance with the characteristics of each mobile device. The policies used in this layer prevent, for instance, that the same content with the same characteristics is sent to a notebook and to a cell phone.

Content is dealt with separately in this layer. 3D objects, text and image have specific modules responsible for applying the adaptation policies required, based on their complexities and different adaptation rules.

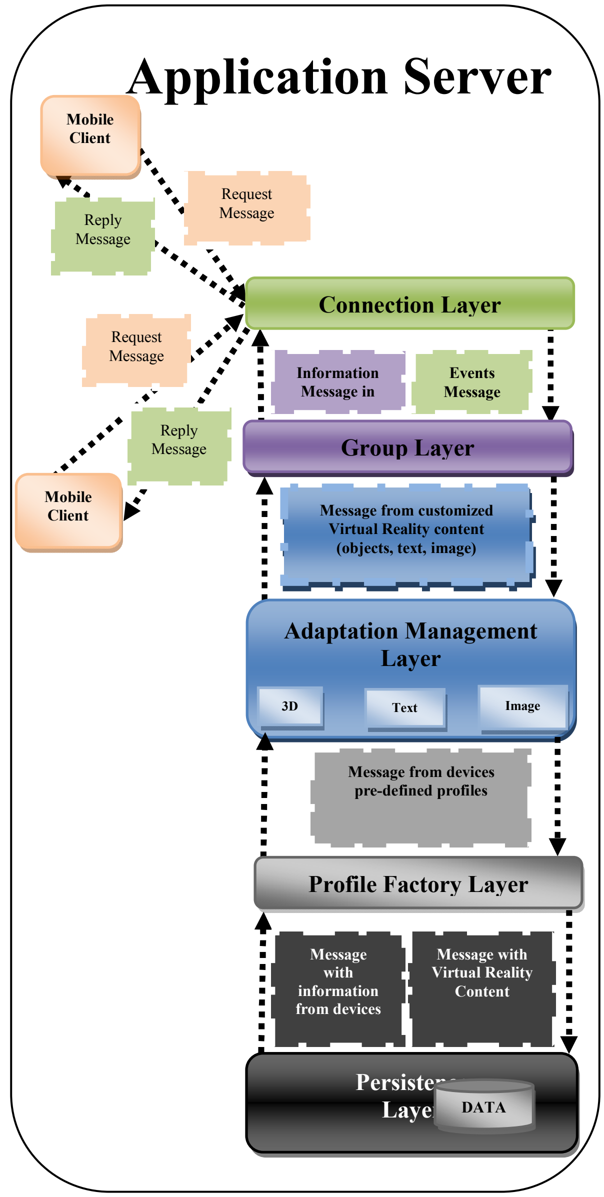

Figure 4. Adaptation Server Working Structure.

V. PROTOTYPE WORKING STRUCTURE

In the prototype that was developed, the mobile client requests a connection with the Adaptation Server, which is responsible for establishing and managing the connection and identifying the device.

After the connection is established, the mobile client initiates the VR content request process by using the options on the client application menu. To execute request activities for images and 3D objects, the client application carries out the steps as follow:

Image: In this step the mobile client sends a message to the Adaptation Server in the format of text. This message contains the following format <data.image.position>. In possession of this message, the server verifies in the database the image requested, and apply the adaptation rules for images to adapt it to the characteristics of the requesting client, and sends a reply (adapted image). The same takes place in case of requests of texts, the message changing to <data.text.position>.

3D objects: To request 3D objects, objects in the M3G format are being used in this work. The process is basically the same as described above, the message being changed to <data.m3g.position>. The server receives this request and applies the necessary content adaptation rules and sends the adapted object to the requesting mobile client.

Adapted M3G content is made available by the means of a Web Apache server, through an address informing where the requested data is stored, as illustrated in Figure 5.

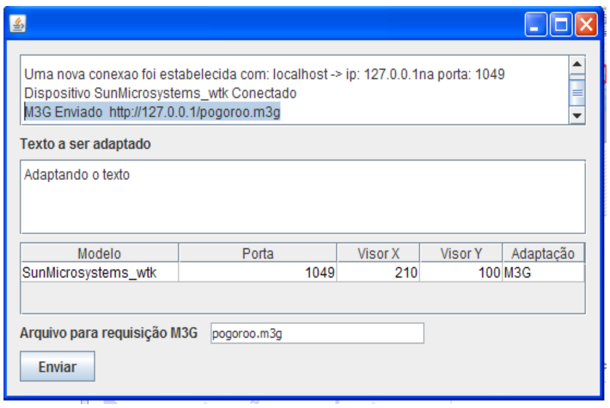

Figure 5. Example of an M3G file.

After this process, the mobile client will receive the object requested, as demonstrated in Figure 6.

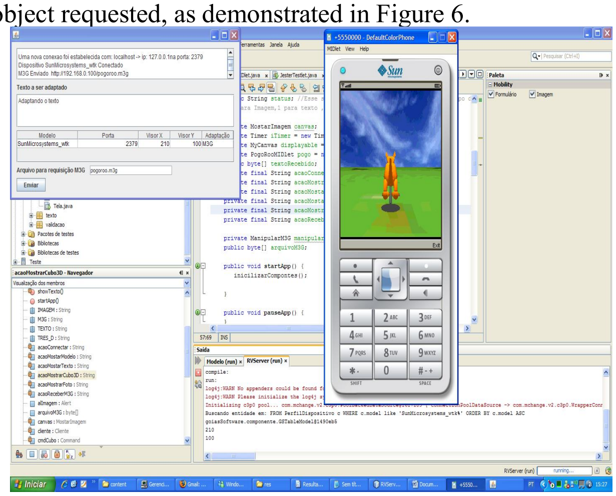

Figure 6. Mobile Client using an adapted M3G file.

The proposed structure permits that several mobile clients are part of the application, independently of their specific characteristics. This client may have the possibility of requesting one or more VR contents.

VI. ARCHITECTURE IMPLEMENTATION

The architecture was implemented by using a Java 2 Micro Edition (J2ME) platform, with a Mobile Information Device Profile (MIDP) 2.0 version, and a Connected Limited Device Configuration (CLDC) 1.1 version [17].

The choice for this platform was based on its portability and the support offered to a great number of devices [9 and 14]. Another motivating factor for the choice was the existence of the Mobile 3D Graphics (M3G) package, defined by JSR-184 [17].

M3G permits the elaboration of interactive applications that use 3D graphic resources for devices with limited processing power. It also defines a standard file format to be used in M3G applications [9].

The proposed solution was developed using the concept of *Design Pattern*, proposed by [18]. The development of layers facilitates the improvement of the solution, according to [14].

The communication services between the Adaptation Server and the Mobile Clients were developed based on the use of the Transmission Control Protocol (TCP) and Internet Protocol (IP) set, known as TCP/IP, [19].

An important non-functional requirement for the architecture is that the Mobile Client is multi-platform, which is one of the factors that have also influenced in the choice for the J2ME platform for the development of this solution. Once this requirement is fulfilled, the proposed solution can be used by a great variety of devices.

VII. TESTS CARRIED OUT ON THE PROTOTYPE

For the assessment of the proposed architecture, tests with MDs with different characteristics and categories were carried out to check the reliability of strategies used by the proposed solution for the adaptation of VR content for mobile clients. Thus, tests sending the same VR content to 3 (three) different mobile clients in an individual way were performed.

The results obtained with these tests can be seen on Figures 7, 8 and 9.

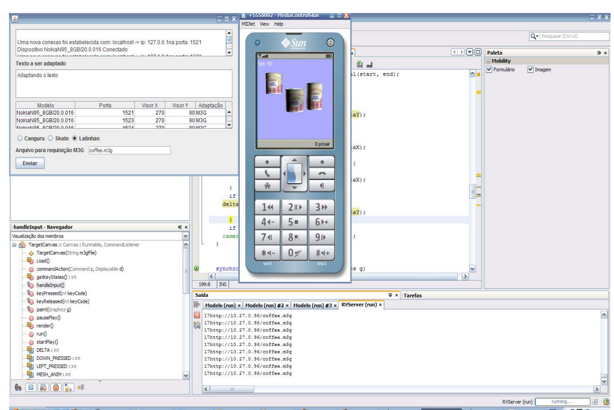

Figure 7. VR content adaptation for Nokia N95.

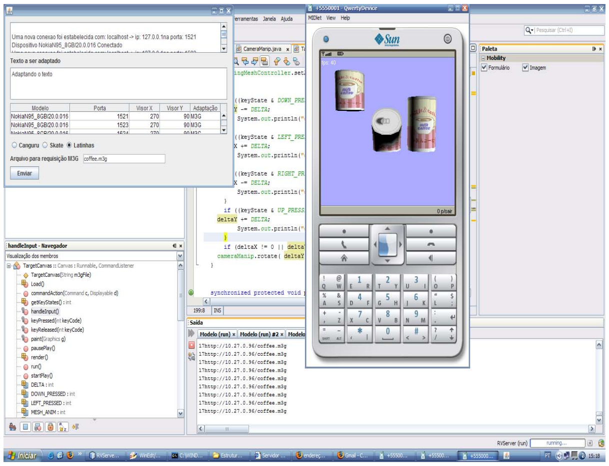

Figure 8. VR content adaptation for BlackBerry.

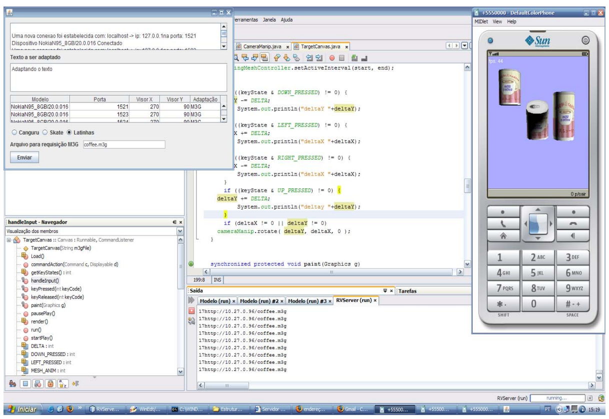

Figure 9. VR content adaptation for Sony Ericsson S400.

From Figures 7, 8 and 9, it can be observed that the results obtained by the adaptation strategies were in accordance with each mobile device individual characteristics.

These results also demonstrate that the adaptation techniques used, together with the pre-defined profiles generated by the Profile Factory Layer, achieved relevant results.

Another test was carried out by sending the same VR content to the 3 (three) MDs mentioned before at the same time, in order to test connectivity and management capacity of several devices by the proposed architecture. The results of this test can be seen on Figure 10.

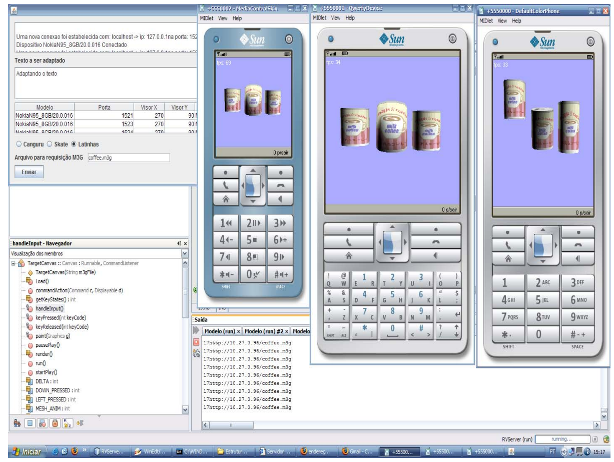

Figure 10. VR content adaptation management results for several devices.

The results obtained in this test show the architecture's management capacity when different devices request the same content at the same time.

The architecture kept the integrity of the VR content requested, even with the application of different strategies in the content (due to MDs specific characteristics).

It is important to point out that testes with different VR contents for each device mentioned above were also carried out, as illustrated on Figure 11.

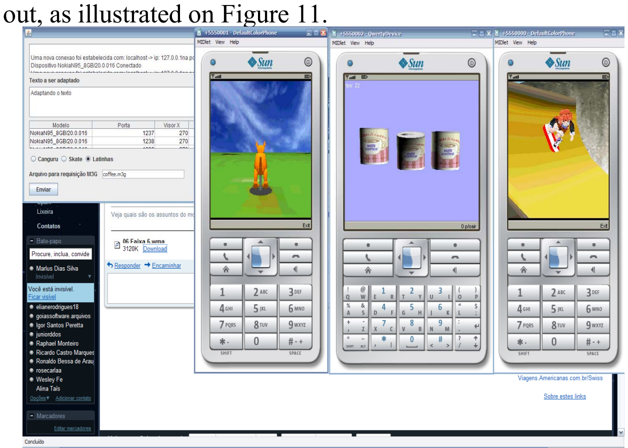

Figure 11. VR content adaptation management results for several devices.

The results obtained in this test demonstrate the efficiency in managing several Mobile Clients. Each client received one different VR content, at the same time, according to their individual characteristics.

In the next item, the conclusions of this work are presented, as well as future work.

VIII. Considerations and Future Work

This work presented a proposal of Software Architecture to provide support to VR content adaptation for heterogeneous mobile devices.

The solution presented permits that devices' characteristics are acquired in an autonomous way, and based on the devices' pre-defined profiles it supplies a solution for VR content adaptation.

The solution presented also permits the management of "N" mobile devices, with different architecture and platforms.

According to the tests carried out, besides supplying support to VR content and devices adaptation, the solution described also keeps the integrity of connections, contents and transmitted information, facilitating the management of VR applications that use one or more mobile clients.

The tests have also shown that without the application of adaptation rules, VR content loses its characteristics, i.e., falls outside MDs standards.

Another relevant result obtained by this architecture is the maintenance of the original content functionalities, due to the management of mobile clients individual characteristics. In other words, the activities (movements, luminosity, etc.) designed in the original three-dimensional scenario are executed in the mobile clients. According to [20], interoperability is measured by the degree by which Information Systems are capable of coordinating and collaborating among themselves.

The use of M3G files format to execute the application of adaptation rules helps the customization process, since M3G is a data format developed for mobile platforms, supplying a native support for 3D data presentation.

The use of CC/PP and UAProf specifications favors the architecture's evolution. Besides supplying an important database to execute device identification, it also goes through updating of their information. It is important to point out that the proposed solution permits the registering of new devices manually, so that it does not exclusively depends on CC/PP updating.

The architecture development using the layers concept will facilitate its use by other applications and its improvement, since code usability makes the development process speedier.

As future work, we can point out the use of different 3D files format, since the structure developed allows for that.

Another interesting application would be the use of context techniques to identify users preferences and characteristics, such as those used by [10], which could also be plausible, since the architecture developed provides support to multiple users.

REFERENCES

[1] Rocha, T., Toledo, Maria B. F. "Mecanismos de Adaptação para Transações em Ambientes de Computação Móvel" in IEE Latin América Transactions, Vol 5, Nº 8, December 2007.

[2] Alcantara, Afonso. Principais desafios da computação móvel. Disponível em <http://www.javasoft.com.br/articles/ desafios_comp_movel.pdf >. Acesso em: 20 jan. 2008

[3] Hao-hua Chu , Henry Song , Candy Wong , Shoji Kurakake , Masaji Katagiri, Roam, a seamless application framework, Journal of Systems and Software, v.69 n.3, p.209-226, 15 January 2004 [doi>10.1016/S0164-1212(03)00052-9]

[4] Viana, Windson ; CAVALCANTE, Paula ; CASTRO, R. M. C. . Mobile Adapter: Uma abordagem para a construção de Mobile Application Servers adaptativos utilizando as especificações CC/PP e UAProf. In: XXV Congresso da Sociedade Brasileira de Computação, 2005, Sao leopoldo, RS-Brasil. Anais do XXXII SEMISH.

[5] Raatikainen, Kimmo E. E.; Middleware for Mobile Aplications Beyond 3G. International Conference on Intelligence in Networks (SMARTNET 2002),7.,p3-18, abril de 2002.

[6] Mckinley, P. K.; Sadjadi S. M.; Kasten E. P.; Cheng, B. H. C. Composing adaptive software. IEEE Computer, v.37, n.7, 2004. p.56-64.

[7] Henricksen, K.; Indulska, J. Adapting the web interface: an adaptive web browser. In: AUSTRALASIAN USER INTERFACE CONFERENCE (AUIC.01), 4, 2001, Gold Coast, Queensland, Austrália. Proceedings... Gold Coast, Queensland Austrália: IEEE, 2001.

[8] LOCUS - Desenvolvimento de Ferramentas de Localização de contexto para Serviços de Informação Móvel. <http://www.locus.org.uk>. Acesso em: 05 de Set. 2008

[9] Marçal, E., Filho. Tese de Mestrado. VirTraM: Um Framework para o Desenvolvimento de Treinamentos Utilizando Realidade Virtual. Universidade Federal do Ceará, 2005.Versão eletrônica <mdcc.ufc.br/component/option,com_docman/task,doc_download/gid,85/>. Visitado em: 03 de Abr. 2008.

[10] Damala, A. Augmented Reality Based User Interfaces for Mobile Museum and Exhibition Guides, CAA 2007, Computer Applications and Quantitative Methods in Arcaheology. Deutsches Archaeologisches Institut, Berlin, In Press.

[11] Schmalstieg, D., Wagner, D. First Steps Towards Handheld Augmented Reality. I n: INTERNATIONAL SYMPOSIUM ON WEARABLE COMPU TERS (ISWC), 7., 2003, USA, Proceedings Washington: IEEE Co mputer Society, 2003. p. 127-137.

[12] Proctor, N., 2004. Access in hand: providing deaf and hard-of-hearing visitors with on-demand, independent access to museum information and interpretation through handheld computers. In: J. Trant and D. Bearman (Editors), Museums and the Web 2005, Toronto, Canada. www.archimuse.com/mw2005/papers/proctor/proctor.html. Acesso: em 10 Set. 2007.

[13] Bowen, J.P. and Filippini-Fantoni, S., 2004. Personalization and the web from a museum perspective. In: J. Trant and D. Bearman (Editors), Museums and the Web 2004, Toronto, electronic version. http://www.archimuse.com/mw2004/papers/ bowen/bowen.html. Acesso: em 10 Set. 2007.

[14] Ito, Giani C. Uma Arquitetura para Geração de Interfaces Adaptativas para Dispositivos Móveis. Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada do Instituto de Pesquisas Nacional de Pesquisas Espaciais – INPE. São Paulo, 2007.

[15] W3C - WWW CONSORTIUM, Site do fórum de desenvolvimento de tecnologias para Web. Disponível em:<http://www.w3c.org >. Acesso: em Jan. 2008.

[16] Site da especificação XML User Agent Profilling Specfication (UAProf). Disponível em: <http://www.openmobilealliance.org/ >. Acesso: em Jan. 2008.

[17] Sun (2004) "Java 2 Platform, Micro Edition (J2ME)". Disponível em http://java.sun.com/j2me /index.jsp. Acessado: em Jul. 2008.

[18] Gamma, E., Helm, R., JOHNSON, R., VLISSIDES, J. Design Patterns – Elements of Reusable Object-Oriented Software. Editora Price, 1995.

[19] TANENBAUM, A. S. Redes de Computadores. 4. ed. São Paulo: Campus, 2003.

[20] BEYNON-DAVIES, P. Information Systems - An Introduction to Informatics in Organizations. Palgrave, Bath, 2002.