An Open-Source Component-Based Toolkit to Support E-learning Multimedia Environments

Thiago Bruno Melo de Sales,
Ivo Augusto A. R. Calado,
Manoel Teixeira de Abreu Netto,
Evandro de Barros Costa,
Henrique Pacca Loureiro Luna
Federal University of Alagoas
Computing Institute
Campus A. C. Simões, BR 104 -
Tabuleiro dos Martins
CEP: 57072-970 - Maceió - AL - Brasil
{tbms, iaarc, mtan, evandro, pacca}@tci.ufal.br

Leandro Melo de Sales
Federal University of Campina Grande
Electric Engineering Department
Av. Aprigio Veloso, 882 - Bodocongó,
CEP: 58109-970 Campina Grande - PB - Brasil
leo@dsc.ufcg.edu.br

Marcello Alves de Sales Junior
Infoserve Technologies Institute
34-36 W 32nd Street, 10th FL
New York, NY 10001

Abstract

In the Distributed Systems area, the number of Multimedia Applications has increased due to fact that it provides a better integration among their users, as well as being recognized as an essential distinctive feature for e-learning environments. However, most of the existing e-learning solutions are not open-source and are not designed with extensibility in mind, making integration with other systems difficult. In this way, this paper proposes an open-source component-based toolkit to support e-learning multimedia environments called ArCoLIVE, which offers real time multimedia services for e-learning systems.

1. Introduction

Education today plays an important role in both personal and professional life. In the global society, learning is required, not only to remain competitive in job markets, but also to enhance our individual participation in culture and society. We need to learn new skills, further training as well as newer and faster ways for them. According to *United States Distance Learning Association*, distance learning encompasses all technologies and supports the pursuit of life long learning for all [2]. Based on that, many educational institutes programs for distance learning are becoming more vital in the world of any level of education, mainly for higher education. The beneficial effects of learners interacting through online communities have been widely studied. Indeed, online discussion is argued to promote student-centered learning and therefore reasonable to suggest that the benefits of online learning should translate into improved student performance[5]. Thus, a viable and effective method that leverages the Internet to deliver anytime/anywhere learning to individual or corporate learners is also required, in order to support the fast growth of these concepts. From rich multimedia interactions to online community and discussion, e-learning technology has the power to increase the access to education and training, while reducing learning costs.

Recently, there has been an extraordinary development and a wide dissemination of network applications that provides audio and video transmission content over the internet. The newest multimedia network applications such as Voice over IP, multimedia Web-Sites, online games, conferencing, virtual worlds, and many others share the same preferences as well as time requirements, scalability and trustworthiness. Due to this explosive increasing development, it is required the creation of new tools, technics, and internet protocols in order to supply the lack of important tools and solve some problems[7].

In this context, ArCoLIVE provides an open-source component-based toolkit that is able to integrate and so, design the most proper environment as possible, enabling users exchange information in an easy way between them. The component-based development aims to build applications with a systematic development characteristic such as scalability and reuse. By using this toolkit, e-learning system developers that need multimedia and synchronous re-

Second International Conference on Systems and Networks Communications (ICSNC 2007)
0-7695-2938-0/07 $25.00 © 2007 IEEE

sources on their applications can take advantages of a set of components such as chat room, Peer-to-Peer (P2P) conversation, virtual white-board, audio/video conferencing, control connection containers, screen capture and file sharing. These components offer new network advanced services by the remote interaction and closer to real communication. With the progress in the digital media compression and multimedia networking, e-learning and media-on-demand applications are rapidly becoming more feasible and accessible. The ultimate goal of distance learning systems is to provide the remote participant most of the capabilities and experience close to the one obtained by an in-class participant. The remote participants can receive a live class feed and are also able to interact and participate in the class by asking questions.

As mentioned before, ArCoLIVE is an open-source project. For this reason, project's releases are available at a software development management system called SourceForge[12], where the scientific community can contribute on the development of such application.

The structure of this paper is organized as follows: ArCoLIVE and its architecture, which comprised of the toolkit (ArCoLIVE-T) and the server which supports it, are presented in section 2. In Section 3, we discuss the use of the ArCoLIVE toolkit in the ArCo, a virtual community framework. Sections 4 and 5 comment some related projects and future works proposals. Finally, section 6 presents a conclusion about our work.

2 The *ArCoLIVE*: Live Internet Videoconference Environment

The ArCoLIVE environment is essentially composed of components based on the JavaBeans technology, a component-based architecture for the Java 2 Platform. As already mentioned, the component-based development approach aims to allow the creation of components that can be reused by other applications. Furthermore this enables the free market of an already tested solution, since these components were all used during the development of them. According to [11], the use of component based development has been successfully applied in distributed systems, giving an effective mechanism for software reuse. Our focus is to develop components to be used as synchronous tools in many of these systems and provide a mechanism to enable the addition of new components. However, some of the ArCoLIVE units, such as ArCoLIVE Media Server, were also developed to support these components in order to build more sophisticated applications.

2.1 Architecture

The ArCoLIVE architecture is divided into three main layers: application, server and persistence, as shown in Figure 1. The ArCoLIVE-T is implemented in the application layer, where components interact with each other. The ArCoLIVE-S layer (see Section 2.3), is capable of handling the components requests and responses, user authentication and provides interface between the application and persistence layers, all based on the ArCoLIVE protocol, described in Section 2.3.

The persistence layer is used to store participant profiles, service features like descriptions, permissions, services instances and some media properties for media-on-demand service.

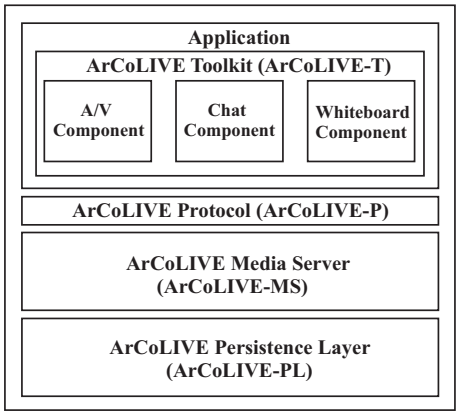


Figure 1. ArCoLIVE Architecture

2.2 Toolkit

The ArCoLIVE toolkit, or shortly ArCoLIVE-T, is composed of 11 JavaBeans components. Some of them are still under development. In what follows, we briefly describe the ArCoLIVE components.

- **ArCoLIVEConnection** - this component supports two well-known protocols for data transmission - UDP (User Datagram Protocol) and TCP (Transmission Control Protocol). For audio and video transmission transport, we use RTP (Real Time Protocol) over UDP and for the rest, control connection and file transfer, for instance, we use TCP. One important property of this component is the multicast support by using the UDP protocol.
- **ArCoLIVEConnectionManager** - it is used when we want to get connection reports. The connection state information are obtained through the RTCP (Real

Time Control Protocol). This component provides data about how many RTP datagrams were sent and received as well as the datagram lose rate.

- **ArCoLIVEPlayer** - it is an audio/video player that plays local media. The main feature of this component is to abstract the source where the stream comes from i.e. a file, a webcam or a microphone plugged in the computer. This component can interact with ArCoLIVERecordMedia, in the case when its user wishes to save that stream.

- **ArCoLIVENetworkPlayer** - this component is capable of playing an audio/video stream that is transmitted by the ArCoLIVE Server.

- **ArCoLIVEProcessor** - this component handles some media control and processing aspects like, format compression, audio/video quality and dimension.

- **ArCoLIVEChat** - this component provides resources to be used in text-based conversation among the participants. The conference manager can moderate the chat session by using the moderated property implemented inside this component.

- **ArCoLIVECaptureDevice** - this component is responsible for managing media-capture devices connected to the local system. By using this component, it is possible to obtain the list of all media devices installed, as well as to get the resource address to use one of the listed devices.

- **ArCoLIVEParticipant** - this component represents a conference participant. One participant can talk in a chat at the same time as he/she can receive an audio/video stream from the server. Participant profile information such as name, email and authentication data can be obtained through this component.

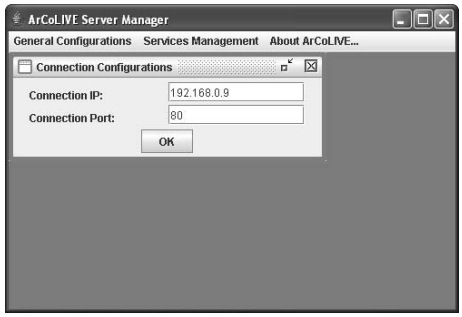
- **ArCoLIVERecordMedia** - the user can use this component in order to save a given media stream while he/she receives it. It is also possible to specify the local destination path where the recorded media will be stored along with its format.

- **ArCoLIVEFileShare** - this component is used to transfer (upload or download) files between the conference participants. After a participant specifies which file he/she wants to send to other participants, the component starts accepting download connections and sends the specified file as to the other participants that had requested it.

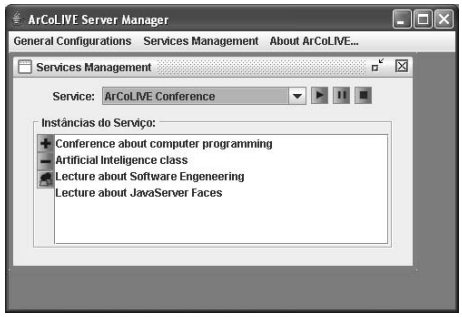
- **ArCoLIVEWhiteBoard** - this component provides a virtual white-board abstraction. A teacher can use it in order to facilitate a lecture explanation, for instance. The conference administrator may also control who is able to write in the virtual board.

2.3 Server

In other to provide request and response support for the components presented in Section 2.2, we offer the ArCoLIVE media server. The ArCoLIVE-MS, in short, which enables user group communication through a network, particularly through the Internet. The goal of the server is to centralize all users' requests/responses and broadcast them as quickly as possible to the remote peers



(a) ArCoLIVE setup connection



(b) ArCoLIVE services manager

Figure 2. ArCoLIVE administrator

The server is organized in services, such as videoconference or chat room. For each service, the server keeps all service instances associated with it and which participant is responsible to manage them. By service instance, its an example of a lecture that takes place in a classroom where the teacher is explaining about biology (one instance of video-conference, chat room or other service that may be required) at the same time that other events can also be transmitted, each one a new service instance associated to that event. The Figures 2(a) and 2(b) respectively show the server connection setup and service manager screen, in where it is possible to:

1. Manage a particular service (start, stop and change its profile);
2. Manage the participants associated to that service;
3. Drop a user from a conference;
4. Disable a user to talk in the chat.

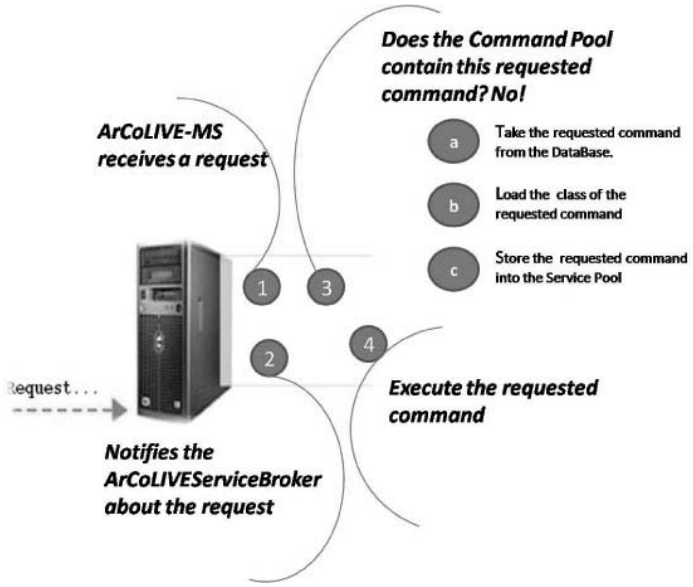


Figure 3. ArCoLIVE-MS dynamic loading service

The ArCoLIVE-MS services are based on the Command design pattern[6], which will give uniformity on their executions through a parameterized way on the service's invocations.

The Figure 3 shows how the ArCoLIVE-MS services are invoked.

1. A request is received by the ArCoLIVE-MS.
2. ArCoLIVEConnection notifies the ArCoLIVEServiceBroker about the request. The ArCoLIVEServiceBroker will determine which command must be executed.
3. ArCoLIVEServiceBroker will verify if the command is already stored in a service pool. If so, step 4 is called. Otherwise:
   (a) A query is done to the database to get the information required to load the command;
   (b) The command will be loaded and prepared to be executed;
   (c) The command loaded will be stored in the service pool to avoid process, such as query to the database, class loading, command pool storage, one more time.
4. The service requested from the ArCoLIVE-Client is executed with proper parameters sent by the remote peer.

The dynamic loading of the ArCoLIVE-MS services can offer a robust extensibility and some advantages can be mentioned, such as:

- New services can be added in runtime.
- ArCoLIVE-MS allows an easily way to add new services just (1) implementing an interface (ArCoLIVEService interface), (2) specify the ArCoLIVE protocol for that service and (3) putting some service's characteristics in the database.
- If a bad-projected service is added, it will not cause problems to others services, since the services execution are independent from each of the other.

The server is capable of handling all kind of service requests and react according to them. The main services provided reflects the main components presented in the previous section: video-conference, chat room, file sharing, virtual white-board and screen capture. To make a standardized communication between client and server and also for extensibility purposes, we have specified and implemented the ArCoLIVE-P. Figure 4 shows the ArCoLIVE protocol stack, where the service protocol is responsible to handle a general user request, determine the service and forward this request to a specific protocol.

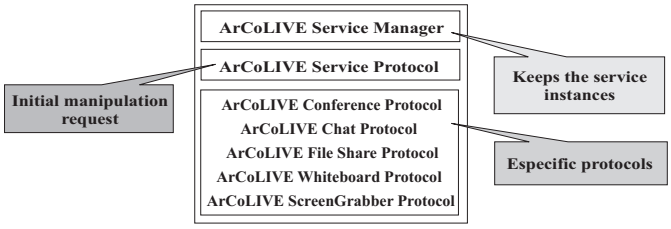


Figure 4. ArCoLIVE communication protocol

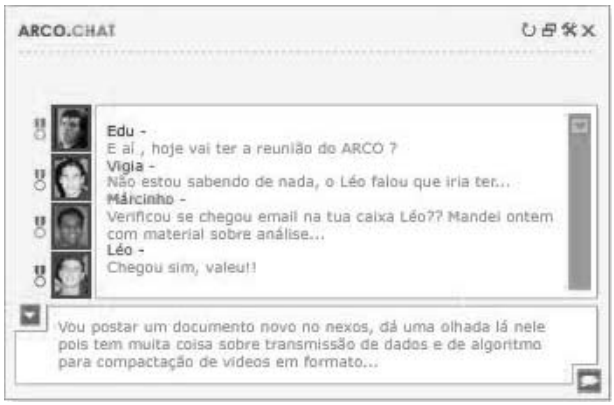
3 Showcase: ArCoLIVE in action

In order to validate some ArCoLIVE components, a study case was done in a Virtual Learning Environment (VLE).

A VLE is a software system designed to facilitate teachers in the management of educational courses for their students, especially by helping teachers and learners with course administration. The system can often track the learners' progress, which can be monitored by both teachers and learners. While often thought of as primarily tools for distance education, they are most often used to supplement the face-to-face classroom.

A free open source VLE known as ArCo[4], whose main goal is to support e-learning through the usage of virtual community environment, has been used to test and run ArCoLIVE multimedia components. Figures 5(a) and 5(b) show the **ArCoLIVEPlayer** and **ArCoLIVEChat** components in action on such VLE.

(a) **ArCoLIVEPlayer**



(b) **ArCoLIVEChat**

Figure 5. ArCoLIVE components example

4 Related Work

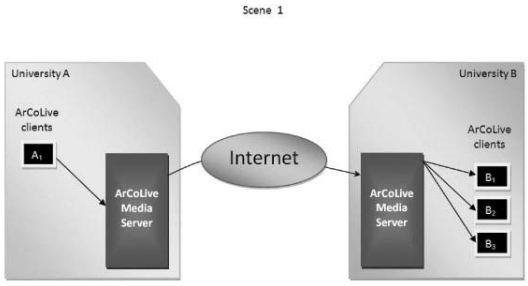
In Sakai Project[10], a wide variety of tools that can assist in the e-learning process is listed, from which some of them can also be found implemented in the ArCoLIVE offerings. However, one of the main ArCoLIVE's focus / propose, multimedia resources usage, like conferencing, doesn't make part of such application, only content storage. In T-Cube[9], an interesting solution is shown for real time screen capture, but this tool restricts the system for only this task and it is not extensible. The Dokeos[1] application, an e-learning system just like Sakai, although providing video-conference functionality, it is not open-source.

Other solution is the Flash Media Server[3] (FMS) and Media Components (MC) from Macromedia. It composes a toolkit that aims to build multimedia applications. This toolkit provides functionalities to handle MP3 and FLV streaming media. Despite the development application facilities, the main disadvantages that may be mentioned are: (1) FMS and MC are not free software, as a result it is an uncompromising solution from a developer point-of-view, preventing them of customizing or extending their application and redistribute it, (2) although FMS supports AVI format, it uses a proprietary audio and video format for media, this means that if we want to integrate other third-party solutions it may not be possible, and (3) FMS users must run the FMS Server on their server in order to handle components requests, which uses a proprietary protocol and, as a consequence, they must be bought in order to be used.

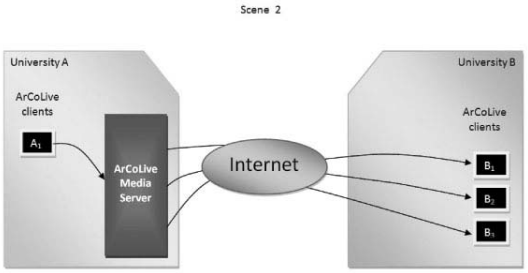
5 Future Works

Since the beginning of the ArCoLIVE project, the ArCoLIVE multimedia components-based toolkit works with JMF Framework[13] to render audio and video, but this framework is exceeded and has not been supported by Sun Microsystems. Hence, an open-source project with the goal of providing a replacement/alternative to JMF, called FMJ[8] has been studied to be integrated with the ArCoLIVE multimedia components.

In order to offer two or more ArCoLIVE Media Server communication, content reflection service has been studied, which can allow the ArCoLIVE clients to connect to the next ArCoLIVE Media Server and get the media content. For instance, suppose a conference between two universities A and B, each of them having an ArCoLIVE Media Server, as shown in Figure 6(a). Now, suppose that one client connected to the server A sends a stream. This stream will be transmitted (by server A) through the internet as an only one stream to the server B. As a consequence, this stream will be reflected at server B to all clients connected to this server. Therefore, its not required that server A has to send more than one stream to clients connected to server B as shown in Figure 6(b). This mechanism will decrease the traffic of network packets therefore the stream can be sent without consuming a wide bandwidth of the network.



(a) ArCoLIVE Reflection Service



(b) ArCoLIVE non-Reflection Service

Figure 6. ArCoLIVE Reflection and Non-Reflection Services schemes

6 Conclusion

In this paper, we presented a set of component-based toolkit with the main goal to support the development of multimedia applications. The increasing need for tools which can give support to multimedia systems, specially to support VLE, and the lack of open-source and extensible multimedia applications has motivated the development of ArCoLIVE.

Due to the high level components independency and extensible media server, it is possible to use them in any e-learning environment. In order to do this, such environment must provide a proper extensible service interfaces. Additionally, the usage of ArCoLIVE toolkit should result in a low cost, extensible and flexible software solution.

References

[1] Dokeos pt. http://www.dokeos.com/pt/. Last access on March, 2007.

[2] United states distance learning association. http://www.usdla.org/. Last Access on March, 2007.

[3] Adobe. Flash media server. http://www.adobe.com/products/flashmediaserver/. Last access on March, 2007.

[4] H. O. Almeida, L. E. F. Tenório, E. B. Costa, N. M. Barbosa, F. M. Bublitz, and A. A. Barbosa. Um arcabouço de software livre baseado em componentes para a construção de ambientes de comunidades virtuais de aprendizagem na web. 2004.

[5] J. Davies and M. Graff. Performance in e-learning: Online participation and student grades. *British J. Educational Technology*, 36:657–663, 2005.

[6] E. Gama, R. Helm, R. Johnson, and J. Vlissides. *Padrões de Projeto: Reutilizáveis de Software Orientado a Objeto*. Bookman, 1st edition, 2000.

[7] J. F. Kurose and K. W. Ross. *Computer networking third edition a top-down approach featuring the Internet*. Addison-Wesley-Longman, 2005.

[8] K. Larson. Fmj: Freedom for media in java. http://fmj.sourceforge.net/. Last access on March, 2007.

[9] M. Ma, V. Schillings, T. Chen, and C. Meinel. T-cube: A multimedia authoring system for elearning. 2003.

[10] U. of Michigan. Sakai project. http://sakaiproject.org/. Last access on March, 2007.

[11] M. Pasin. *Réplicas para Alta Disponibilidade em Arquiteturas Orientadas a Componentes com Suporte de Comunicacão de Grupo*. PhD thesis, Federal University of Rio Grande do Sul - UFRGS, 2003.

[12] SourceForge. Sourceforge. https://sourceforge.net/. Last access on April 2007.

[13] A. Terrazas, J. Ostuni, and M. Barlow. *Java Media APIs, Cross-plataform Imaging, Media and Visualization*. SAMS, 1st edition, 2002.