Scientific Research and Essays Vol. 7(19), pp. 1857-1863, 23 May, 2012
Available online at http://www.academicjournals.org/SRE
DOI: 10.5897/SRE12.022
ISSN 1992-2248 ©2012 Academic Journals

Full Length Research Paper

Distributed multimedia presentation control – A system perspective

Md. Anwar Hossain[1]*, Jorge Parra[2], Dewan T. Ahmed[1], Atif Alamri[1], Ahmed Ghoneim[1] and M. Shamim Hossain[1]

[1]College of Computer and Information Sciences (CCIS), King Saud University, Riyadh, Kingdom of Saudi Arabia.
[2]IKERLAN-IK4 Technology Research Center, Spain.

Accepted 10 May, 2012

In many meeting scenarios, such as in a conference, delivering multimedia presentation is an important activity that is supported by different presentation systems. Most of the current presentation systems provide a tightly-coupled solution where the model (source presentation), the view (rendering), and the controller (mouse/keyboard) work as a single unit of interaction. Although this approach is easier to be familiar with, it does not provide any option to separate these aspects of the system. Separating these aspects (loosely-coupling) on the other hand could, a) allow a presenter to use other interaction devices such as a cell phone to navigate the presentation slide, b) facilitate the mobility of the viewer who can roam around and still follow the presentation if needed, and c) enable multiple people to interact with the presentation, which is often required in a collaborative session. In this paper, we show the design and development of such a system, which is based on service-oriented-architecture and follows a model-view-controller like design pattern. Our experiments show applicability of the approach and assure its flexibility.

Key words: Presentation control, slide navigation, service-oriented architecture, model-view-controller, discoverable web services.

INTRODUCTION

In a meeting scenario, presentation is one of the most important tasks a presenter does. In traditional presentation system, the presenter usually uses mouse/keyboard or infrared controller to navigate through the presentation slide. However, the overall presentation system is tightly coupled, where the source presentation (model), the renderer (the view), and the interaction device (controller) work as a single unit, which basically restricts the presenter to use the given system and oblige an audience to view this presentation from the system-adopted renderer (e.g. projector). The presenter might want to control the navigation of the slides from his/her cell phone, while the audiences might want to follow a presentation from their own device remotely or check two or more presentations simultaneously, which is out of the scope in the existing systems.

There are many works that address different issues related to multimedia presentation. For example, Java Application Sharing in Multi-user INteractive Environments (JASMINE) collaborative system (Shirmohammadi et al., 2003) allows the participants to actively communicate among them in a collaborative session, while a moderator controls the session when sharing a presentation. The JASMINE system is not particularly targeted for multimedia presentation rather sharing presentation with online participants. In Wang and Fox (2004), the authors demonstrated a presentation sharing approach between remote clients using a message brokering approach by sending text messages for specialized Impress presentation navigation. They require that the Impress presentation software be installed in all clients.

*Corresponding author. E-mail: mahossain@ksu.edu.sa.

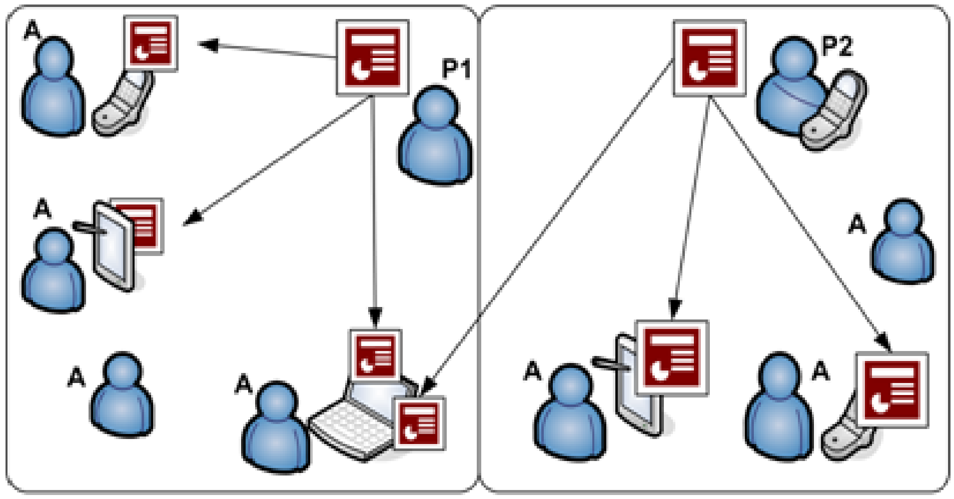

Figure 1. Example conference presentation scenario.

The MultiPresenter system (Lanir et al., 2008) demonstrates a presentation system for large display spaces. Among many other features, the system allows changing the flow of a presentation and interactively manipulating the content during presentation. The NextSlidePlease (Spicer and Kelliher, 2009) is a framework that allows structured authoring of presentation content and formatting the content with a view to increase the interactivity between presenter and audiences.

Anderson et al. (2004) develops a tablet PC based lecture presentation system to leverage the best of existing computer-based and manual presentation systems, such as the addition of improved handwriting features. In Myers (2001), the author showed the application of handheld device to better control the presentation flow, which is also consistent in our approach.

A low cost mobile pointing device is introduced in Jantz et al. (2007) that utilizes LED and camera to recognize the pointers and accordingly navigate the presentation slide or draw on the presentation wall. Another work (Zhang et al., 2002), broader in focus, proposed NetMedia as a middleware that allows a client to retrieve customized multimedia content from multiple servers through synchronized streaming.

Unlike these approaches, we focus on a presentation scenario and show how the presentation can be controlled from a presenter's handheld device and the audiences can get the snapshot of the ongoing presentation in their devices without interrupting the presenter.

Our contribution in this paper is twofold: first, we propose the design of a distributed and pervasive presentation application that allows the separation of a presentation view from the navigation control. This approach enables the presenter to control the presentation from his personal handheld device. Second, we provide a mechanism to share the multiple presenters view with the audience in their own device. In addition, we also enable the support of controlling the presentation by many parties, when such an application is used in a collaborative session.

The remainder of this paper is organized as follows. We first provide a motivating scenario, followed by the proposed system design, implementation, experiments, discussions and future work directions.

MOTIVATING SCENARIO

Before delving into details of our proposed application, we present a motivating scenario where the application will find its applicability. Consider for example Figure 1, which shows two ongoing presentations in two rooms. In the rooms, there are audiences with and without their own devices. One of the presenters (P1) uses the existing conferencing control (e.g. mouse and keyboard) to navigate the presentation slide while the other (P2) controls the presentation from his smart phone. The audience can either follow the main presentation view directly in the room projector or they can choose to view the main presentation from their own device. In this figure we also state that an audience can simultaneously follow two presentations from the two rooms. Having this kind of arrangement in a conferencing session would provide better experience to the presenter as well as to the audience due to the flexibility it provides. It also enables a remote client to view an ongoing presentation, while he/she is unable to attend physically. In this paper,

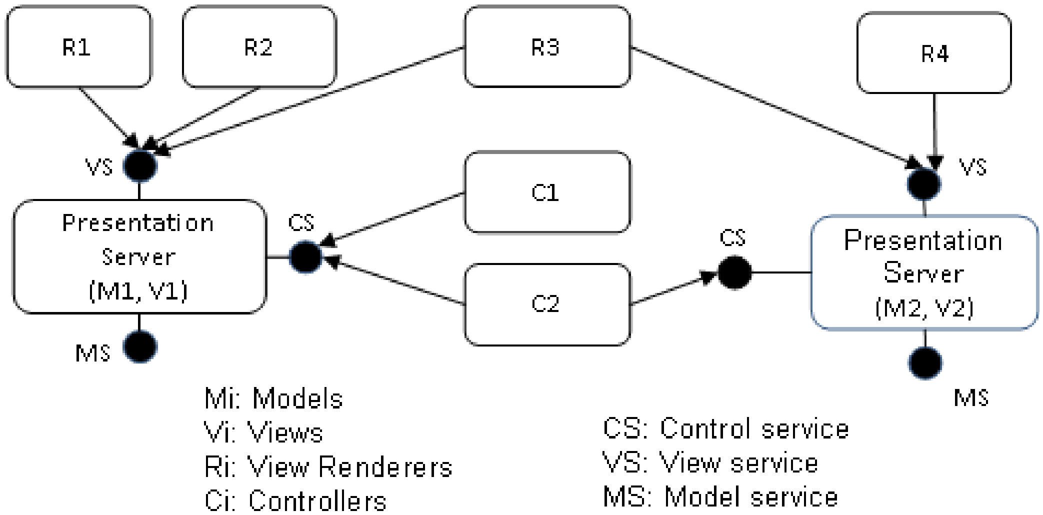

Figure 2. Separation of View-Renderer-Controller in a presentation system.

we aim to realize this scenario.

PROPOSED DESIGN

Here, we illustrate the design of the proposed presentation application. The core idea of our approach is to define a View-Renderer-Control logical component separation scheme to enable distributed rendering and control of a presentation view. We note an analogy of our design with the well-known Model-View-Controller (MVC) pattern (Gamma et al., 1994) that makes a logical separation of the model, the view and the controller in order to increase its flexibility and extendibility. MVC allows different views to be created for the same data model. However, in a presentation scenario there is a single view per model, which could be rendered simultaneously in different devices and also be controlled by many clients if necessary. Our proposed design facilitates such view sharing and control. Figure 2 shows a schematic view of this separation based on multiple presentation servers that have their own data model and view (e.g. M1, V1). In this figure we show that R1 and R2 are rendering the same view V1, R4 is rendering V2 and R3 is capable of rendering V1 and V2 simultaneously or alternatively. We also show that each presentation view can be controlled by a single or many controllers, for example, C1 and C2 is capable of controlling V1.

We now identify the basic requirements of the proposed distributed multimedia presentation system:

i) The presentation data creates a single/unique view (REQ1)
ii) A single presentation view can be rendered in several displays (REQ2).
iii) The presentation slide navigation can be controlled from several devices (REQ3).
iv) The pointer can be controlled from several devices (REQ4).
v) All the functionalities should be handled-enabled independently (REQ5).

In this research, a detailed design of the system was provided. First we introduce the proposed system model; then we describe Web Services as communication middleware solution; and finally we justify the use of a service discovery protocol to minimize system configuration tasks.

System model

Here the design of the system in terms of different interface and component definition required for View-Renderer-Control logical distribution approach was described. In order to fulfill the requirements we define the following separate (REQ5) service interfaces as depicted in Figure 3.

REQ1: ISourceAllocator: allows setting the source data for the presentation (e.g. a ppt file).
REQ2: IViewer: allows renderer devices to get the current presentation view to be rendered locally.
REQ3: INavigator: defines a set of methods to navigate through the presentation (e.g. StartPresentation and NextSlide) from the user’s handheld device.
REQ4: IPointer: defines the required methods to control the pointer in terms of pointer position and clicks (e.g. mouse movement position, left button click, right button click…).

According to the required functionalities, the presentation server is composed of two separate entities that manage different aspects: a) IViewer and IPointer are screen related features that are implemented by a ScreenController entity, and b) ISourceAllocator and INavigator are presentation related functionalities that are implemented by a PresentationController entity that internally uses a specific presentation application manager (such as MS PowerPoint).

Based on the class and interface definition in Figure 3, we provide a component view of the system in Figure 4. Here, the main PresentationServer component contains the entities that implement the previously stated service interfaces. We also define individual client/controller components for each of the proposed functionalities. Thus, a SourceAssigner component will be responsible for assigning the presentation data to the PresentationServer component using the ISourceAllocator interface; a NavigationController component will interact with the server to send navigation commands according to the INavigator interface; controlling the pointer in the presentation view will be supported by a PointerController component that interfaces with the server using IPointer; and finally, a ViewRenderer component will be responsible for acquiring the current presentation view from the server in order to enable distributed view rendering. In this case, the ViewRenderer will ask the PresentationServer to provide the

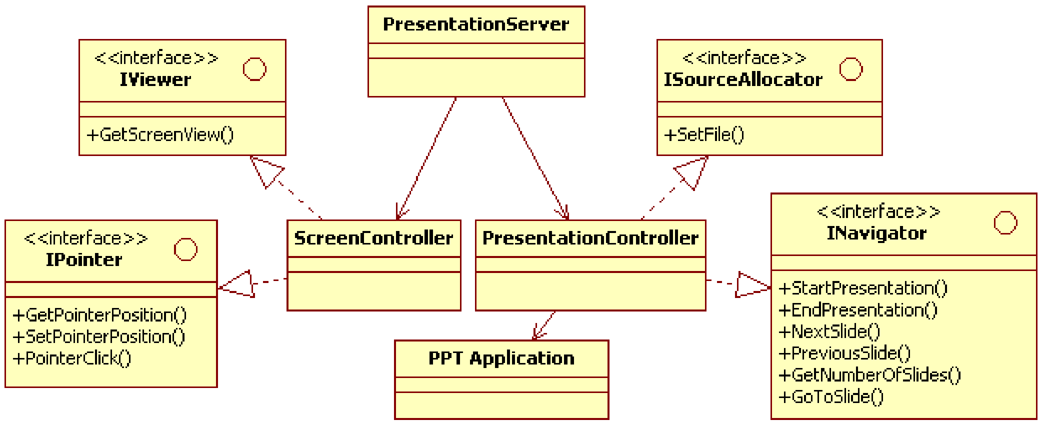

Figure 3. Interface and class definition of the system.

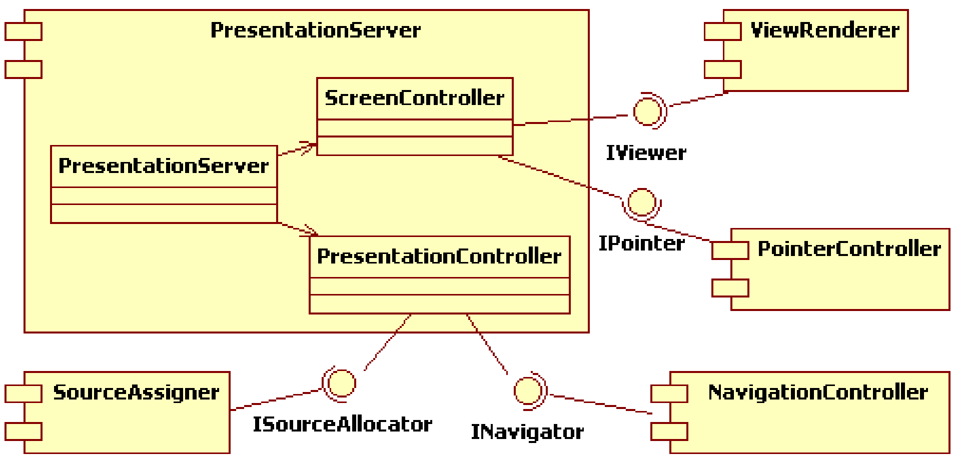

Figure 4. System component model.

presentation view according to its own rendering capability and the PresentationServer will repurpose the view and send it to the renderer. More detail on this will be given in the implementation section.

The main advantage of designing separate interfaces and client/control components is the ability of deploying any set of control components in networked devices having different access and control capabilities. Consequently, taking benefit from this logical separation, presenters could have access to all the control services while audience could have a restricted set of control capabilities. Therefore, the set of control components that are deployed in the participants' devices establishes their interaction capabilities. Furthermore, the number of participants (even sharing the same role, e.g. several presenters) is not limited because the control components can be deployed simultaneously in separate devices, enabling interactive or collaborative presentations.

Communication Infrastructure

In order to distribute the components shown in Figure 4, we need to establish a message interchange platform. Taking into account the service oriented approach, the network support, and the loose-coupling needs we adopted SOAP/XML Web Services as the communication middleware. Thus, PresentationServer component exposes the designed interfaces as Web Services, while control components implement clients to interact with the PresentationServer. The methods defined in the proposed service interfaces are mapped to Web Service operations.

Component discovery

Presentations and meetings are distributed in dynamic environments where participants (presenters and audience) can carry their own devices running client/control components, which will need to find and locate the PresentationServer component. This dynamicity makes component discovery a basic feature for successful system integration. Web Services Dynamic Discovery (WS-Discovery) (Beatty et al., 2005) defines a multicast discovery protocol that enables Web Service clients to locate Web Services. Thus, making use of the WS-Discovery protocol, the participants though a user-friendly interface can search for available PresentationServers in the network that are identified by type and/or scope attributes. We define the PresentationServer type, and identify each PresentationServer instance with scope information. Clients can multicast a Probe message with specific service scope and service type information, and

Table 1. Implemented client roles.

Role	Source selector	Navigator	Pointer	Viewer
Presenter	X	X	X	X
Restricted Audience				X
Collaborative Audience			X	X
Co-presenter		X	X	X

the PresentationServers in the network that match the request should respond to client with their location information. Hereafter, client/control applications can send control commands to the PresentationServer using this service location information.

IMPLEMENTATION

A proof of concept of the proposed application design has been implemented to demonstrate and validate our objective. Hence, this paper provides details of the implemented prototype.

A Presentation Server application has been implemented in C# language using Microsoft .NET Framework. The Microsoft PowerPoint 12.0 Object Library has been used to control a Microsoft PowerPoint application in the presentation server.

We have also implemented different client applications for various client roles. Table 1 shows the components deployed in each client application according to the client role. These client applications have been developed on Microsoft .NET Compact Framework using C# language and run on HTC Touch Cruise Smart Phones.

We used .NET Deployment Framework tools developed in the IST Amigo Project (Amigo, 2001) to implement the Web Services and Clients, supporting WS-Discovery protocol.

For the distribution of presentation view, we implemented a configurable polling mechanism instead of sending a continuous stream aiming at keeping a lightweight communication channel. Client applications periodically (we used a 1 second interval) asks for the current presentation view of the server and specifies the desired image height and width supported by the respective client devices.

In response to the client request, the server captures a presentation snapshot and adapts it to the requested size. This snapshot is serialized and sent to the client in the response message.

Note that developing mobile phone-based applications is very popular nowadays due to the increased pervasiveness of such devices and the enhanced experience it provides to its users. This justifies our choice of using mobile phone-based client to interact with the presentation view. Special attention was paid to mobile client's usability and readability. In addition, the use of NET platform for our implementation is for demonstration purpose only, other programming paradigm could also be chosen.

EXPERIMENTAL RESULTS

We conduct experiments with the developed application in terms of several test cases, followed by a discussion of the results in light of the test cases.

Test case realization

We defined five test cases and explain how we realized these test cases in weekly meeting conducted in our lab.

Test case 1: Load a presentation from a smart-phone to the presentation server and control the presentation from this phone.

To realize this scenario, we deployed the PresentationServer service in a desktop machine, where a projector is connected. We deployed the client services in the phone, which supports .Net Compact Framework. The client application was capable of discovering the presentation service using the approach described in "Component discovery". We used the source allocator interface in the phone (Figure 5) to set a source PowerPoint presentation file and started the presentation to render it in the PresentationServer machine as well as on the phone GUI. Further navigation of the presentation (e.g. pointing to a highlighted term, moving to the next slide, previous slide and so on) was performed from the phone GUI.

Test case 2: Allow many audiences to view the current presentation from their own device.

This test case was realized by deploying the client application in two PDAs and three smart phones. However, in this case we restricted the newly installed clients' capability to only view the presentation. One presentation server was setup as in test case 1. While the presenter was navigating the presentation from his phone, the audiences (with the new client installations) were also able to discover the PresentationServer service from their own client interface and could start to view the current presentation. The client view was updated according to the view update interval (1 s). Here giving restrictive access to the audience was done to ensure that they do not interface the presenter's interaction and control.

Test case 3: Allow an audience to follow a specific presentation from another room remotely.

In this case, we followed the same procedure as in test case 2, except that an audience with his device was located in a different room than the actual presentation location. As the remote client and the PresentationServer were in the same network, the remote client could easily discover the presentation server and follow the presentation.

Test case 4: Allow an audience to view two presentations that are being presented from different rooms.

Figure 5. An unrestricted view of a client.

This resembles a conference scenario where there are many presentations going on simultaneously in different rooms. To emulate this, we installed the PresentationServer service in two machines. In this case, an audience from his client interface could discover those two presentation servers (Figure 5). This allowed him to alternatively invoke both presentations from his device.

Test case 5: Allow multiple people to present a common presentation one after another from their handheld devices.

This test case is defined to explain how the proposed system can support collaborative viewing and simultaneous controlling of the presentation. To demonstrate this, we installed one PresentationServer service and three clients without any restrictions in the client devices. Therefore, all the three clients could discover the PresentationServer service and could navigate the same presentation from their own device while presenting their part. At this stage, the navigation and pointer control collision that may arise in this scenario due to multiple-party interaction are resolved manually by the presenters themselves. However, as all the presenters will have the same unrestricted view in this test case, their client application will automatically obtain the updated presentation status (such as current slide being displayed), which will ensure smooth navigation.

DISCUSSION

The test cases we realized show that the application has great potential in several presentation scenarios. It provides the flexibility of controlling a presentation from different client devices and allows following up a presentation from within the client devices. However, we have the following observations on the proposed system:

i) With the increased number of client devices that render the same presentation view, the network may suffer from bandwidth problem. Therefore, in the current implementation the main presentation view is captured as images for rendering in the client devices instead of streaming. As a result, the system is not able to show the animation within the presentation.

ii) Although providing presentation view along with presenter's speech (audio) to the mobile device was our overall goal, the current version of the prototype does not yet support recording and transferring the presenter's speech to the client. However, to the presenter, the system supports easy navigation and control of the presentation that does not require audio. To the audience, if they are in the same place as the presenter is, the audio is not required either. Of course, in some scenarios audio would be much appreciated.

iii) As the client is the user's handheld device, readability of the presentation may slightly suffer due to the limited screen landscape. The users can minimize this effect by hiding the menu items from the main presentation screen while a presentation is going on.

iv) The system supports navigation and controlling of presentation view from multiple devices (as stated in REQ3 and REQ4), which apparently may introduce collision. This is handled by implementing the client roles defined in Table 1. Basically, in a general presentation scenario the presenter will have the full control and the audience will have the view access, while in collaborative presentation scenario the co-presenters will be able to navigate and control the presentation as they like.

CONCLUSIONS AND FUTURE WORK

In this paper, we have illustrated the design and development of a multimedia presentation system, which allows the separation of presentation navigation control from the view and supports rendering of the same view to multiple client devices. We have conducted the experiment with the developed prototype application using a number of test cases, which shows the applicability and potential of the proposed system. Our future work will focus on the support of different mobile platforms, enable presenter's voice transfer to mobile clients, and improve the view rendering mechanism. In addition, we would like to apply the proposed technique in different application scenarios such as collaborative learning environment.

ACKNOWLEDGEMENTS

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through the research group **project No RGP-VPP-049.**

REFERENCES

AMIGO: http://www.hitech-projects.com/euprojects/amigo/. Last accessed on 28 Sep. 2011.

Anderson R, Anderson R, Beth S, Steven AW, Tammy V, Ken Y (2004). Experiences with a Tablet PC based Lecture Presentation System in Computer Science Courses. Proc. 35th SIGCSE Tech. Symp. Comput. Sci. Educ. (SIGCSE'04), 36(1): 56-60.

Beatty J, Gopal K, Devon K, Thomas K, Brad L, Bryan R, Christopher St. J, Jeffrey S, Guillaume S, Doug W, Jack W, Yevgeniy Y, Prasad Y (2005). Web Services Dynamic Discovery (WS-Discovery). Available at, http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf. Last accessed on 10 Dec. 2011.

Gamma E, Richard H, Ralph J, John V (1994). Design Patterns: Elements of Reusable Object-Oriented Software, Second Edition, Addison-Wesley Professional.

Jantz K, Gerald F, Lars K, Raúl R (2007). A Low-Cost Mobile Pointing and Drawing Device. Proc. ACM Workshop Educ. Multimed. Multimed. Educ. (EMME), Germany.

Lanir J, Booth KS, Tang A (2008): MultiPresenter: A Presentation System for (Very) Large Display Spaces. Proc. ACM Multimed. Conf., Vancouver, Canada.

Myers BA (2001). Using Handhelds and PCs Together. Commun. ACM, 44(11): 34-41.

Shirmohammadi S, EL saddik A, Nicolas DG, Ralf S (2003). JASMINE: A Java Tool for Multimedia Collaboration on the Internet. Multimed. Tools Appl., 19(1): 5-19.

Spicer R, Kelliher A (2009). NextSlidePlease: Designing Tools for Creating Flexible Presentations. Proc. NSF Workshop Media Arts, Sci., and Tech. (MAST).

Wang M, Fox G (2004). Design of a Collaborative System. Proc. IASTED KSCE Conf., Virgin Islands.

Zhang A, Yuqing S, Markus M (2002). NetMedia: Streaming Multimedia Presentations in Distributed Environments. IEEE Multimed., 9(1): 56-73.