PRESENTING INTERACTIVE MULTIMEDIA DOCUMENTS WITHIN VIRTUAL ENVIRONMENTS

Paulo N. M. Sampaio[1,2], Laura M. Rodriguez Peralta[1,2] and João Pedro D. Pereira[2]
[1]*Centre for Informatics and Systems of the University of Coimbra (CISUC)*
[2]*Laboratory of Distributed Systems and Networks (Lab-SDR)*
University of Madeira (UMA), Campus da Penteada 9000-390, Funchal, Madeira, Portugal

Keywords: Virtual Reality, Interactive Multimedia Documents, CSCW, Awareness.

Abstract: Different languages and tools have been used for the modeling and building Virtual Reality applications which can be applied in different domains. These tools are each time more sophisticated providing means for the author of a 3D environment to build his model intuitively. However, these languages and APIs used for building virtual environments are still limited when it comes to integrate multimedia content inside these applications. This paper presents a generic and extensible solution for the presentation of integrated Interactive Multimedia Documents within Virtual Reality applications.

1 INTRODUCTION

The integration of multimedia content inside Virtual Environments (VEs) is a promising and interesting trend in the development of Virtual Reality applications since interaction can be enhanced, and through the addition of audio and video, the user´s immersion inside the VE can be improved. Indeed, a multimedia presentation consists in the integrated presentation of different media objects (e.g., images, text, animation, etc.) where at least one of these objects is continuous (a video or an audio). The definition of a multimedia presentation can be generalized by the concept of Interactive Multimedia Documents (IMDs).

Besides Multimedia, the benefits of the Computer Supported Cooperative Work (CSCW) inside VEs should not be neglected: Cooperation motivates and increases productivity stimulating users to communicate with geographically dispersed participants of a collaborative session inside a Multimedia and Collaborative Virtual Environment (MCVE).

Many Virtual Reality (VR) systems have been proposed in the literature as collaborative environments, which were developed within different application domains: e-learning (Chee, 2001), (McArdle et al., 2004), (Halvorsrud and Hagen, 2004), collaboration among workgroups (Bochenek and Ragusa, 2003), augmented collaborative spaces (Pingali and Sukaviriya, 2003), Multimodal VR applications (Carrozino et al., 2005), among others. Indeed, the rapid prototyping, modelling and authoring of CVEs has been a major concern to many authors, as presented in (Rodrigues and Oliveira, 2005), (Osawa et al., 2002), (Ficheman et al., 2005) and (Garcia et al., 2002). Although, most of the systems propose the development of CVEs, few of them (or none) explore the presentation of integrated multimedia content inside CVEs (Walczak et al., 2006).

One of our main research interests is the proposal of a generic solution for the presentation of IMDs within virtual environments. This paper presents one representative effort in this direction, the development of an API to provide the integration of a multimedia player with a VE, and which can be easily adapted and extended for the presentation of complex MCVEs.

This paper is organized as follows: Section 2 introduces the multimedia player adopted in this project; Section 3 presents the main aspects related to the development of the proposed API; Section 4 discusses some lessons learned; Section 5 presents a case study of the application of the proposed API; Finally, Section 6 presents some conclusions.

N. M. Sampaio P., M. Rodriguez Peralta L. and Pedro D. Pereira J. (2008).
PRESENTING INTERACTIVE MULTIMEDIA DOCUMENTS WITHIN VIRTUAL ENVIRONMENTS.
In *Proceedings of the Third International Conference on Computer Graphics Theory and Applications*, pages 512-518
DOI: 10.5220/0001098805120518

2 PRESENTING CONSISTENT IMDS

The design and presentation of complex IMDs can be an error-prone task since we cannot ensure that all the synchronization constraints specified by the author can be respected during the presentation of the document. Some methodologies and techniques have been proposed to support the design of consistent IMDs (Courtiat and Oliveira, 1996), (Layaida et al., 1995), (Mirbel et al., 2000), (Jourdan, 2001). In particular, the methodology presented in (Sampaio et al., 2007) provides the formal design (specification, verification, scheduling and presentation) of complex Interactive Multimedia Documents.

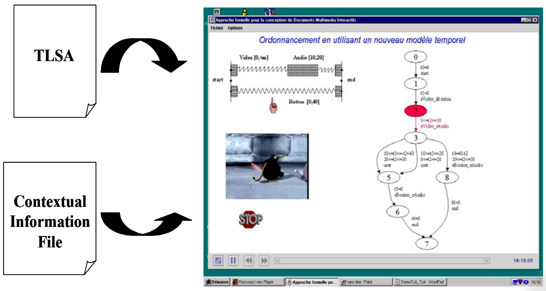

TLSA
Contextual Information File
Ordonnancement en utilisant un nouveau modèle temporel
Video [0,∞]
Audio [10,20]
start
end
Button [0,40]
STOP

Figure 1: Snapshot of the TLSA Player.

A prototype for the presentation of IMDs, called TLSA Player was implemented in this methodology. The TLSA Player relies on the TLSA (Timed Labeled Scheduling Automata), which is the scheduling graph adopted in the previous methodology and on the *Contextual Information*, which describes the non-temporal components of the document, for supporting the presentation of complex and consistent IMDs. The TLSA Player was implemented using JAVA (jdk 1.2) and JMF 2.0. Figure 1 illustrates a snapshot of the TLSA Player.

The TLSA Player was adopted in this project since it offers a flexible and open Java-based architecture which can be easily adapted to embed multimedia content inside Java3D VR environments. The integration of the TLSA Player with a VR environment is presented in the next section.

3 INTEGRATING IMD'S INSIDE VIRTUAL ENVIRONMENTS

The main idea behind this project was to implement an API that could be easily instantiated during the creation of virtual environments to render a texture-like presentation of Interactive Multimedia Documents (or a multimedia texture) on 3D objects.

Before building complex VEs, we first decided to focus on the problem (the integration of multimedia and VR) using a simple 3D object to illustrate the solution proposed. Thus, we applied a cube to be "wrapped around" with a multimedia texture.

However, the applied language and APIs (Java, JMF and Java3D) do not provide means to render a video object integrated with the rest of its related multimedia document inside a VE. Therefore, we decided to come up with an alternative solution that enabled the presentation of IMDs, including video, inside the VE. This solution consisted in extracting the video object frame-by-frame from the presentation of the TLSA Player, and then re-integrating these frames afterwards with the multimedia presentation inside the VE.

The solution proposed can be broadly applied to enrich virtual environments, where users can navigate, visualize the presentation of integrated media, and also interact with the presentation if they want. The multimedia presentation inside virtual environments can be applied in different domains such as education, virtual collaborative meetings, tourism, health care, etc. These solution and architecture proposed on this work can be adapted easily to any java-based virtual reality application. The architecture and some implementation details are described on the next sections.

3.1 Architecture

The architecture for this solution describes all of its components, their properties and the relations among these components. The architecture proposed is presented in Figure 2.

The architecture of this application is composed of three different components: (i) the *multimedia presentation*, (ii) the *media processing*, and; (iii) the *virtual reality integration component*.

The *multimedia presentation* component is implemented by the TLSA Player, and its integration with the system aims at extracting the image and video objects presented by the TLSA Player.

The *media processing* component is responsible for extracting the image and video objects being presented by the TLSA Player. These image and video objects are blended afterwards and presented into the virtual environment. This component is composed of the: *Image Extraction*, *Video Extraction* and *Image Composition* modules.

The *Image Extraction* module aims at extracting images from the presentation area of the TLSA Player. These images represent all the visible multimedia content being presented by the player with exception of the video objects.

The *Video Extraction* module aims at extracting only the video objects being presented by the TLSA Player. When a video object starts to be presented by the TLSA Player, a notification with the video´s spatial position is sent to the *Video Extraction* module to enable to extraction of the video frames to be further presented inside the virtual environment.

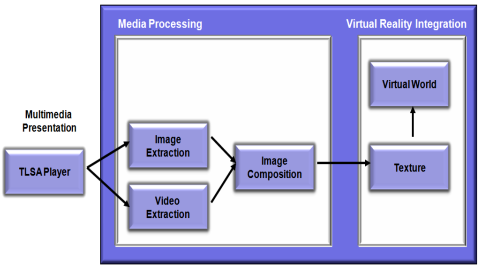


Figure 2: Architecture of the solution.

The *Image Composition* Module is in charge to blend both image objects obtained from the *Image Extraction* and *Video Extraction* modules, and to send the resulting image to be presented into the virtual environment.

We should note that when there is no video presentation on the TLSA Player, there is no communication between the *Image Extraction* and *Video Extraction* modules. Thus, the image received from *Image Extraction* module corresponds to the presentation carried out on the virtual environment.

The *Virtual Reality Integration* module is responsible for rendering and managing all the components of the virtual environment. This module is composed of the: *Texture* and *Virtual World* modules.

The *Texture* module receives continuously the images generated by the *Image Composition* module and integrates these images as textures into the virtual environment. The texture mapping process is described by "wrapping around" a 3D object with the texture of the multimedia presentation (Figure 3).

The *Virtual World* module builds and manages the virtual environment with all the 3D objects to which the multimedia textures are applied. This is one of the main modules of this architecture since it is responsible for carrying out the final integration between the *multimedia processing* module and the *virtual reality integration* module. Some implementation details and issues are further discussed on the next section.

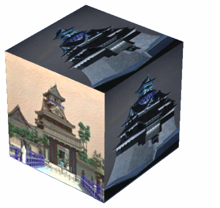


Figure 3: The texture of a 3D object.

3.2 Implementation of the Prototype

Besides some technical details of implementation, this section also focuses on the main issues that conducted the implementation decisions to the correct presentation of IMDs inside virtual environments. These issues are discussed for each implemented module on the next sections.

Multimedia Presentation Module. This module was proposed for the presentation of Interactive Multimedia Documents. At first, two solutions were considered: the implementation of a multimedia player, which would be a time consuming task, or; the utilization of a multimedia player already developed and easily adaptable to our needs. The second choice seemed the most reasonable. Naturally different players and multimedia formats were considered such RealPlayer, Flash, among others. However, these players could not be easily adapted to cope with the prototype which was developed using Java3D. For this reason, the TLSA Player was chosen due its characteristics for the presentation of correct SMIL documents (which can be easily authored), and since its Java-based code could be easily adapted to integrate the application that would provide the presentation of multimedia content inside a virtual environment.

Image Extraction Module. Due the limitations of Java and the APIs applied, the multimedia presentation could not be carried out properly within the virtual environment with the minimal level of quality expected. In fact, Java3D which is the language used for the construction of the virtual world still is not compatible with the API JMF, which used by the TLSA Player for the presentation of audio and video. For this reason, we had to propose and evaluate different solutions during the development of this module. Among these solutions, we considered:

- To export all the multimedia presentation from the TLSA Player to a file, which would be loaded afterwards by the application to be presented as a multimedia texture on the 3D object;

- To generate images associated with the multimedia presentation of the TLSA Player continuously, and render these images as multimedia texture on the 3D object.

These solutions are discussed further on the sequence.

Generation of image files. The *Image Extraction* module first extracted images from the presentation area of the TLSA Player, which were saved as JPG files to be applied afterwards for the presentation within the virtual environment.

This solution was not useful due the huge amount of disk space used to save the generated files. Moreover, the latency between the multimedia presentation of the TLSA Player and related multimedia presentation inside the virtual environment could not be neglected, most of the time leading the presentation to a deadlock.

Another disadvantage of this approach is the considerable amount of images generated, for instance, one minute presentation generated around 2000 images (with 30 fps). For these reasons, this solution was discarded.

Generation of internal images. Another solution considered was the generation of images from the TLSA Player and further rendering inside the virtual environment.

The images would be obtained by the utilization of an API which enables to take a screenshot of the presentation area of the TLSA Player at a given instant. This screenshot could then be sent to the virtual world. This API, called *printAll*, allows the capture of all the graphic components being presented by the TLSA Player sending this information to an image.

Nevertheless, the API *printAll* is not able to capture the video object being presented by the TLSA Player. Figure 4 (a) illustrates the image captured from the TLSA Player, corresponds to the dashed area.

Video Extraction Module. This module allows the transformation of each frame of a video object into an image. In order to do so, the TLSA Player provides previously the presentation position of the video so that it can be rendered on the correct position within the original presentation on the virtual environment. Note that this solution can also be applied independent of the number of video objects presented in parallel in an IMD. The video to be extracted is illustrated inside the dashed area in Figure 4 (b).

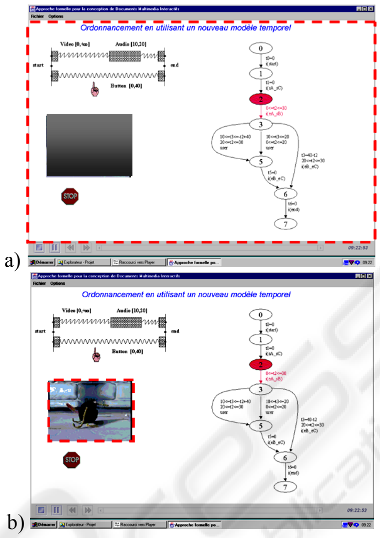

a)
Ordonnancement en utilisant un nouveau modèle temporel
b)
Ordonnancement en utilisant un nouveau modèle temporel

Figure 4: (a), Image generated with printAll, (b)The video to be extracted from the TLSA Player.

Image Composition Module. This module blends the images generated by the *Image Extraction* and *Video Extraction* modules. As we can see in Figure 5, after capturing a video frame corresponding to a temporal position of the TLSA Player, another image is also captured corresponding to the same temporal position of the multimedia presentation of the TLSA Player. These images are blended by the *Image Composition* module and sent afterwards to the *Virtual Reality Integration* Module.

Texture Module. This module receives as a parameter the images sent by the Image Composition module, and produces a multimedia texture to be presented on a 3D object. The streaming of images and the presentation on the 3D objects is managed by the Virtual World module.

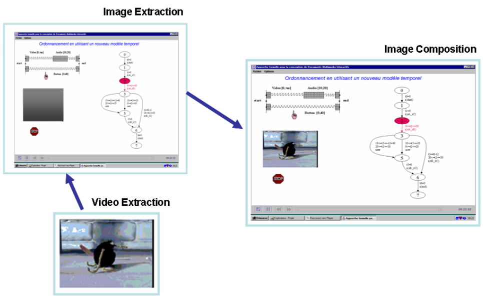


Figure 5: Image Composition module.

Virtual World Module. The main concern for the implementation of this module was to propose a solution to sequence all the images generated by the *Image Composition* module and to manage their presentation inside the virtual environment continuously with minimal delay as possible.

At first, a potential solution was to apply threads for the presentation. However, that would represent an extra burden to the system and more latency for the streaming of images. To solve this problem, we applied an API called *Renderer* which allows the presentation of continuous media (video, in particular) without affecting the performance of the presentation. Despite, we were not handling video directly, but images, we needed to synchronize these images in a frame-based sequence. Indeed, Renderer made the process of acquiring and rendering of images much easier and faster.

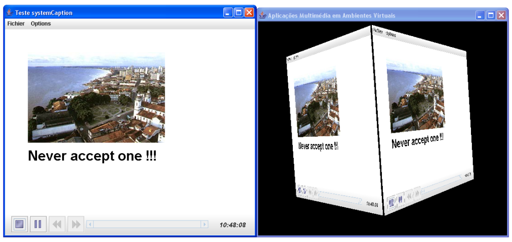


Figure 6: Execution of the application.

Figure 6 illustrates the execution of the implemented application, where we can observe the TLSA Player presenting the IMD at the left side, and this IMD also being presented on the virtual environment at the right side.

4 LESSONS LEARNED

The efficacy of the solution proposed for the presentation of IMDs within virtual environments can be guaranteed by the perfect integration among the multimedia player (TLSA Player), the *Media Processing* and *Virtual Reality Integration* modules.
Different IMDs were created and tested within the virtual environment with the application developed, and some conclusions can be taken from these tests:

- Utilization of a multimedia player that can be easily adapted for the presentation inside virtual environments: Our research group had already previous experience with the TLSA Player, and besides promoting the presentation of correct documents, this tool has a java-based code which could be adapted straightforward to needs of the application being developed;
- Possibility of choosing the document to be presented: The presentation of the IMD can be naturally configured on the TLSA Player, and this presentation will be carried out on the virtual environment;
- Extensibility: The application developed can be extended to support the rendering of the multimedia texture on any surface inside a virtual reality application;
- Performance of the presentation: The application was developed to support the continuous presentation of IMDs within virtual environments. However, some minor failures and delay of milliseconds still can take place without affecting the global quality of the presentation.
- The occurrence of failures is due the huge resource utilization by the application, and the latency is due the different operations that must be carried from the acquisition of images, integration of the video and the multimedia presentation and the composition of multimedia textures inside virtual environments.

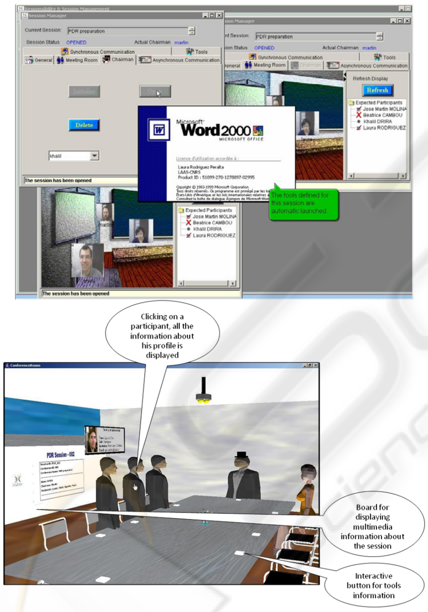
Different applications were developed, and they had a reasonable performance with the proposed solution. One of these applications aimed at implementing a virtual meeting room which would provide information about the members of a collaborative session, and about their collaborative work, as an awareness tool, as described on the next section.

5 IMPLEMENTING AN AWARENESS TOOL FOR MCVE'S

One of the problems initially presented in CSCW is the absence of context among the participants of a collaborative session, which occurs when a team group does not know what a particular member is doing, the global situation of the work and where a particular activity integrates on the entire system (Schmidt, 2002). According to (Gutwin and Greenberg, 2002), the term "awareness" is used as a label designating various, more or less specified, practices through which cooperating actors, while engaged in their respective individual activities and dealing with their own local urgencies and troubles, manage to understand what their colleagues are doing (or not doing) and to adjust their own activities.

After authentication, the proposed awareness tool provides the visualization of all the events or actions during an active session (on-line) in a virtual meeting room . This application was integrated with the developed prototype for the presentation of multimedia content inside the VE. The prototype was validated by a Distributed System Engineering (DSE, contract IST-1999-10302) project scenario: the collaboration revision phase in a space engineering system program. Figure 7 (a) illustrates some of the DSE collaborative tools.

The developed awareness tool provides the online session state description in a virtual world which can be visualized by a generic web browser. This browser can be integrated to the online session manager system without developing extra software for viewing the current online session state. This visualization is very helpful to maintain the current session state awareness for all the connected users. As a consequence, users are able to coordinate their common tasks easier through the active tools during a synchronous session.


Current Session: PDR preparation

Session Status OPENED

Actual Chairman martin

The session has been opened

The tools defined for this session are automatic launched.

Session Manager
Synchronous Communication
Tools
General
Meeting Room
Chairman
Asynchronous Communication
Initialize
Open
Delete
khalil
Microsoft Word 2000
Refresh Display
Refresh
Expected Participants
Jose Martin MOLINA
Beatrice CAMBOU
Khalil DRIRA
Laura RODRIGUEZ
ConferenceRoom
Clicking on a participant, all the information about his profile is displayed
Board for displaying multimedia information about the session
Interactive button for tools information

Figure 7: (a) DSE collaborative tools; (b) Displaying information about the profile of a participant.

In this tool, the participants are represented by 3D avatars and other information related to their profile (such as: name, email, organization, role, IP, etc.) is represented by multimedia objects (image, text, audio and video) in a virtual meeting room. Also, the proposed awareness tool provides a description of the tools that the participants are working on and sharing.

The virtual meeting room is composed of a set of chairs, a table, a board (where a multimedia texture created by the Texture module is rendered in real time). This tool allows the user to navigate in the meeting room, through the X, Y or Z axis, using the mouse and the keyboard.

Some of the possible interactions inside the virtual Meeting Room are: clicking on a particular participant, all the information about his profile is displayed on a window or on the whiteboard (Figure 7 (b)); clicking on one of the ten buttons on the table, which are next to the participants, the information about the tools he is using is displayed. This information about a session is presented in real time after a user interaction.

6 CONCLUSIONS

The development of this prototype allowed the proper presentation of multimedia content inside virtual environments, as proposed initially. Unfortunately, the platforms and languages available for building virtual worlds still do not support the presentation of integrated presentation of IMDs. For this reason, alternative solutions had to be proposed according to the needs and capabilities of the platform used to build the 3D environment. Besides presenting the multimedia content inside VEs, the API developed in this work can still be extended and applied in complex VR applications.

REFERENCES

Chee, Y.S.: Network Virtual Environments for Collaborative Learning. Invited talk. In Proceedings of ICCE/SchoolNet 2001—Ninth International Conference on Computers in Education , Seoul, S. Korea. ICCE/SchoolNet (2001) 3–11.

McArdle, G.; Monahan, T.; Bertolotto, M.; Mangina, E.: A Web-Based Multimedia Virtual Reality Environment for E-Learning. Proceedings Eurographics 2004, July 2004, Grenoble, France.(2004)

Halvorsrud, R.; Hagen, S.: Designing a Collaborative Virtual Environment for Introducing Pupils to Complex Subject Matter. NordiCHI '04: Proceedings of the third Nordic conference on Human-computer interaction. Tampere, Finland, (2004), 121-130, ISBN: 1581138571.

Bochenek, G.M.; Ragusa, J.M.: Virtual (3D) collaborative environments: an improved environment for integrated product team interaction?: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003. (2003), 10 pp.

Pingali, G.; Sukaviriya, N.: Augmented Collaborative Spaces. In proceedings of the 2003 ACM SIGMM workshop on Experiential telepresence, International Multimedia Conference. Berkeley, California, (2003).

Carrozino, M.; Tecchia, F.; Bacinelli, S.; Cappelletti, C.; Bergamasco, M.: Lowering the development time of multimodal interactive application: The real-life experience of the XVR Project.: In ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, (2005).

Rodrigues, S. G. & Oliveira, J. C.: ADVICe - um Ambiente para o Desenvolvimento de ambientes VIrtuais Colaborativos. XI Simpósio Brasileiro de Sistemas Mutimídia e Web - WebMedia2005, Poços de Caldas, MB, Brasil, (2005).

Osawa, N.; Asai, K.; Saito, F.: An interactive toolkit library for 3D applications: it3d. In proceedings of the workshop on Virtual environments 2002, ACM International Conference Proceeding Series; Vol. 23, Barcelona, Spain, (2002). 149 - 157

Ficheman, I. K.; Pereira, A. R.; Adamatti, D. F.; Oliveira, I. C. A.; Lopes, R. D.; Sichman, J. S.; Amazonas, J. R. A.; Filgueiras, L. V. L.: An interface usability test for the editor musical. In proceedings: International Conference on Enterprise Information System - ICEIS 2005. Miami USA v. 5, (2005) 122-127.

Garcia, P.; Montalà, O.; Pairot, C.; Skarmeta, A.G.: MOVE: Component Groupware Foundations for Collaborative Virtual Environments. Proceedings of the 4th international conference on Collaborative virtual environments, Bonn, Germany, (2002) 55 – 62.

Walczak, K., J. Chmielewski, M. Stawniak, S. Strykowski., Extensible Metadata Framework for Describing Virtual Reality and Multimedia Contents. Proceedings of the 7th IASTED International Conference on Databases and Applications DBA 2006. Innsbruck, Austria¸ (2006) 168-175.

Courtiat, J.-P., Oliveira, R. C., Proving Temporal Consistency in a New Multimedia Synchronization Model. In: 4th ACM Multimedia'96, Boston, USA, (1996) 141-152.

Layaida, N., Keramane, C.: Mantaining Temporal Consistency of Multimedia Documents. In: ACM Workshop on Effective Abstractions in Multimedia, San Francisco, (1995).

Mirbel, I., Pernici, B., Sellis, T., Tserkezoglou, S., Vazirgiannis, M.: Checking Temporal Integrity of Interactive Multimedia Documents. In: VLDB journal, (2000).

Jourdan, M.: A formal semantics of SMIL: a Web standard to describe multimedia documents. In: Computer Standards and Interfaces, 23 (2001), (2001) 439-455.

Sampaio, P.N.M.; Rodriguez Peralta, L.M.; Courtiat, J.-P.: Designing Consistent Multimedia Documents: The RT-LOTOS Methodology. In Proceedings of the 5th International Conference on Formal Modelling and Analysis of Timed Systems- Formats´2007. Springer-Verlag in Lecture Notes in Computer Science, Salzburg, Austria, LNCS 4763, ISSN 0302-9743. October 3-5, (2007) 290 – 303.

Schmidt K.: The Problem with "Awareness. Computer Supported Cooperative Work. The Journal of Collaborative Computing, vol. 11, (2002) 285-298.

Gutwin C., Greenberg S.: A descriptive framework of workspace awareness for real-time groupware. Computer Supported Cooperative Work. The Journal of Collaborative Computing, Vol. 11, (2002) 3-4.