Chat-Based Application to Support CSCL Activities

Ricardo Tesoriero[1](✉), Habib M. Fardoun[2], and Hachem Awada[3]

[1] University of Castilla-La Mancha, Av. España S/N, 02001 Albacete, Spain
ricardo.tesoriero@uclm.es
[2] Information Systems, King Abdulaziz University, Jeddah, Saudi Arabia
hfardoun@kau.edu.sa
[3] Information Systems, Faculty of Education, Lebanese University, Beirut, Lebanon
awadahachem@hotmail.com

Abstract. The evolution of the actual labor market requires people to perform long life learning activities. As people leave the formal education system, the time and place to perform these activities varies according to their private life. Therefore, traditional learning scenarios where students are located in the same classroom as professors at the same time are not the most suitable for people that have left the formal education system. Distance learning programs enable students to carry out learning activities anytime anywhere. This article presents a chat tool that enables users to center the discussion on an artifact; where the artifact can be a report, a map, an image, a slide or any electronic document. During the discussion, users exchange messages and documents. Messages are contextualized using references to: previous messages, documents or document fragments of text and images. These graphical references linking chat messages to observations in the document improve the understanding of messages, reduce the message composition time, enable participants to focus on different threads of information at the same time and introduce 3 different types of context information into the message: temporal, conceptual and observational; which are not available on traditional chats.

1 Introduction

The evolution of the actual labor market requires people to perform long life learning activities. As people leave the formal education system, the time and place to perform these activities varies according to their private life. Therefore, traditional learning scenarios where students are located in the same classroom as professors at the same time are not the most suitable for people that have left the formal education system. Distance learning programs enable students to carry out learning activities anytime anywhere.

Computer Supported Collaborative Learning (CSCL) fosters the collaboration among teachers and students to improve the learning experience. These activities are traditionally carried out using Learning Management Systems (LMS) [6], such as Moodle [14]. These systems provide many tools to carry out

G. Meiselwitz (Ed.): SCSM 2017, Part II, LNCS 10283, pp. 161–177, 2017.
DOI: 10.1007/978-3-319-58562-8_13

collaborative activities such as email, chats, forums, workshops, surveys, questionnaires, wiki, blogs, peer evaluation, slide presentations, videos, documents, etc. These tools are an excellent option to carry out simple activities such as getting answers to questions using forums and chats, making simple decisions using surveys, presenting results in blogs, building a wiki to show findings, presenting a subject of study using slides or videos, etc.

However, there is a gap between the formulation of the problem to solve and the presentation of the conclusions of the work. This gap in the process is filled with the analyses and discussions that leaded to the conclusions. These tight-coupled activities are not well-supported by traditional LMS.

These activities involve students as well as teachers. For instance, suppose a hypothetic scenario where teachers provide students with a video, or a set of slides, containing the subject to learn. Then, students have a slot of time to ask questions that are answered by teachers, or other students in a forum. Once questions are answered, teachers ask students to write in groups research reports to evaluate the knowledge acquired by students.

From the Computer Supported Collaborative Work (CSCW) perspective, the presentation, question and answer, and writing report activities, are carried out asynchronously. However, the process involving each of them can be performed synchronously.

The presentation and question and answer sessions can be performed using audio or video conference systems. They enable students to ask questions to teachers during the presentation session instead of using a forum. Students have the opportunity to ask questions, and teachers to answer them, in the context in which the part of the subject requires extra explanation.

The writing research report group activities involve 3 phases: the learning material review (slides, electronic books, notes, annotations, etc.), the selection of findings to introduce into the reports, and the report authoring activities.

During the learning material review, students share questions and answers that use the slides as the center of the discussion. Then, students discuss about the findings to select them. Again, the discussion is centered on the documents (HTML pages, notes, spreadsheets, slides, docs, book chapters, etc.) students have found during the research process. During the learning material review and the selection of findings phases, students employ different tools to exchange information (i.e. chats, audio and video conferences, desktop sharing applications, on-line document sharing tools, etc.) Finally, students compose the research report based on the discussion carried out during the selection of findings phase using on-line document editing tools or file sharing utilities.

This article presents a tool to improve the communication between teacher and students where text documents, spreadsheets, slides and images, play an important role during the conversation. This tool enriches messages with information related to the context in which they were posted enabling users to recover the context afterwards.

This article is organized as follows. Next section presents the motivation for this work. Then, it defines the contextual chat conference tool functionality in terms of the user interface and the system architecture characteristics. Afterwards, it discusses the benefits of using this tool, and how this proposal improves learning activities. Finally, it presents conclusions and future work.

2 Motivation

In face-to-face scenarios, students and teachers share information during learning activity sessions. During these sessions, teachers share comments with students using slides, geographical maps, mathematical function graphs, equations, block diagrams, conceptual maps, etc. to expose subject contents. Besides, share comments with their colleges using practice spreadsheet reports, annotations, book chapters, etc. to fulfill teacher assignments The result of these activities usually consists only of summaries of the discussion and the conclusions.

Therefore, a large amount of information is lost during the transition between the discussion and the conclusion of the learning activity. This information could be critical in future activities where they have to answer questions such as Why did we discard this alternative? Why did we choose this one? Who proposed this alternative? Who suggested this solution? As only the summary of the discussion and conclusions where stored, it is difficult to backtrack in the discussion to answer these questions.

In distance learning scenarios, discussions are carried out in different ways; for instance, using conference phone calls, video conferences, and lately, personal computer assisted conferences (i.e. Skype) where users do not have to be grouped into a video conference room. The way teachers and students share information varies according to the media they use to communicate.

2.1 Conference Phone Calls and Audio Conferences

In conference phone calls and audio conferences, documents are printed on paper before the session.

The main problem regarding printed documents is setting the right document in the conversation context because participants have to synchronize these documents “by hand”. For example, suppose that medical students are discussing about the evolution of a patient by analyzing 3 clinical analyses. To comment a set of values that are out of the normal range in a section of a report, they have to explicitly include in the comment the context information about the section of the report the comment is about.

This overhead of information may lead to misunderstandings. Suppose that a student comments to a section of an analysis without mentioning to the analysis. As the same section is present all reports, the rest of the students may link the comment to sections of different analysis.

This situation becomes worst if students talk about different versions of images, where the section (or region) to contextualize the comment is difficult to describe.

2.2 Video and Personal Computer Assisted Conferences

In video conferences and personal computer assisted conferences, the software usually provides sharing mechanisms such as desktop sharing applications (i.e. Microsoft NetMeeting [24]) or document sharing capabilities (i.e. Google Docs [23]).

They provide participants with a synchronized view of the document that is the subject of the conversation because all participants share the view of the same artifact. However, participants dealing with different documents sharing the same view have to deal with the "document overlapping" problem where participant hide some documents to see other documents.

A compromise solution is achieved when combining document and desktop sharing applications. Participants use the document sharing application to manipulate documents, and the desktop sharing application to show their contributions to the rest of the participants. Although the "document overlapping" problem is solved, and the information overhead decreases, this approach does not provide participants with any mechanism to link document to comments. For instance, following the clinical analyses example, the information about the evolution of the levels located in different analyses cannot be explicitly linked to a comment without introducing explicit contextual information.

Sometimes conclusions cannot be obtained in a single session; therefore, it is important to record all the information during to continue with the discussion. If the media to record the information is audio or video, the analysis and review process of a conversation is not easy.

From a technological perspective, searching information on this type of media is difficult. From a semantic point of view, it is not easy to discern between information that is part of the focus of the conversation, from the information that is part of the comment context.

Although it is possible to link video and audio resources to text comments at specific times using voice recognition technologies, it is not easy to link videos to external resources such as documents, images, graphs, etc. due to the separation between the communication tool and the resource manipulation application.

2.3 Document Comment Tool

A potential solution to this problem lays on the use of comments provided by the most of document sharing applications. Comments are an interesting alternative to support the exchange of information among session participants. The main problem behind this approach is the lack of temporal context provided by conversation.

Comments are attached to the structure of the document, instead of the temporal development of the conversation. It is difficult to follow a conversation from document comments, because the order in which comments are located is defined by the structure of the document, instead of the time the comment was created. This situation becomes worst when participants are dealing with more than one document at the same time. Each document have its own structure.

Another issue to take into account is the relationship between comments. In forums, participants are able to link posts to other posts to represent a relationship between them. For instance, a participant posting a question usually finds the answer to the question in a nested post. Therefore, during a meeting, participants are able to refer comments that are related to other comments that are not the actual subject of the conversation.

For instance, suppose that a group of medical students are discussing about the set of symptoms affecting a patient that are linked to clinical analysis results. At the beginning of the conversation, a student suggests a medical diagnostic assay to diagnose the potential disease causing of these symptoms. To keep in mind which symptoms the student is referring, the comment about the medical diagnostic assay is linked the comment related to the symptoms. As the conversation goes on, the patient exhibit new symptoms, and new alternatives about the disease are taken into account. At the end of the discussion, students can easily relate the assay to diagnose assay to the symptoms and then, relate the assay to the clinical analysis results. Thus, comments have temporal and subject contexts at the same time.

We summarize the analysis of the meeting scenarios as:

1. Lack of information regarding the learning activity process
2. Overhead of information related to the need to introduce context information into comments
3. Difficulty in the analysis and review of conversations using audio and video conferences
4. Lack of relationship on communication channels. While resources are managed by document sharing applications; conversations are carried out using conference systems
5. Lack of temporal awareness when using "the comment tool" on document sharing applications during a conversation, due to the comment dependency on the document structure
6. Observations cannot be easily related to other observations due to the lack of a mechanism to support temporal and semantic linking at the same time.

This article presents a contextual chat conference tool that enables participants to create references to different parts of meeting documents to cope with the problems mentioned in the previous paragraphs.

3 The Contextual Chat Conference Tool

The Contextual Chat Conference Tool (CCCT) is inspired by chat conference applications that enable a set of participants to exchange text messages and anchored conversations [4]. This proposal links messages to documents, fragments of text, regions of images, mixed parts of a document including regions of images and fragments of texts, and other messages.

The remaining of this section exposes the most relevant characteristics of the CCCT user interface supporting the reference mechanism that links documents to messages. Besides, it describes the architectural characteristics of the system used as a platform to support the exchange of messages as well as documents.

The CCCT employs a multi-user communication channel where all the information is shared by all the chat participants that are connected to a channel or chat room.

The information density of a channel of information defines the unit of information (i.e. character, line, or message) chat participants exchange. The density of the information that flows through a CCCT channel is the message. Therefore, the message composition process is private, and no one, except for the author, is able to see any message until it is posted through the communication channel.

As, The CCCT is a Groupware application that can be classified using to the space/time classification matrix proposed in [10]. From the temporal perspective, this application support both, synchronous and asynchronous work sessions. It supports synchronous work sessions because it enables participants to exchange messages in real-time. Besides, it supports asynchronous work sessions because conversations are persistent to enable the analysis, review or continuation of work sessions. From the space perspective, as the most of chat systems, CCCT enables in-situ as well as geographical distributed conversations.

The extension to the classification proposed in [17] adds 3 new characteristics to classify CSCW systems: Information sharing, Communication and Coordination. This application is a *communication* tool that supports the *information sharing* by enabling participants to exchange documents and messages linked to portions of these documents. Although it could be used as a *coordination* tool (you can use this tool to schedule the next steps of a rehabilitation process); it was not intended to play this role.

According to [20], there are two types of elements that are part of a computer supported communication: the artifact and the prose. While the prose expresses an observation textually, the artifact is the center, or the focus, of the conversation. In CCCT, the role of the artifact is played by documents (such as presentations, reports, images, etc.), and comments linked to document regions play the role of the prose (i.e. messages).

3.1 The Contextual Chat Conference Tool User Interface

The Fig. 1 depicts the 5 interaction areas the CCCT user interface: the Artifact Interaction Area (AIA), the Prose Review Area (PRA), the Prose Arte fact Connection Area (PACA), the Prose Composition Area (PCA) and the Presence Awareness Area (PAA).

The PAA provides participants with 2 interaction components: the user presence list, where participants see the rest of the participants' state (Away, Busy, On line, Free to chat, etc.), and the participant state selector where they set their own state.

The PCA enables participants to compose the prose to be sent to the chat conference channel. It also enables them to change the message font family, size and color as well as set the bold and italic font attributes.

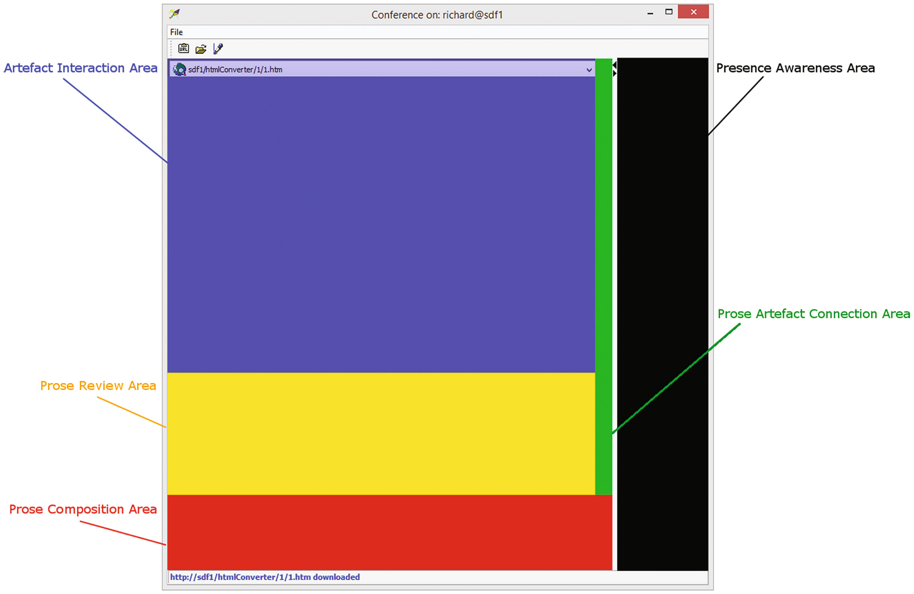

Conference on: richard@sdf1
File
sdf1/htmlConverter/1/1.htm
http://sdf1/htmlConverter/1/1.htm downloaded
Artefact Interaction Area
Presence Awareness Area
Prose Artefact Connection Area
Prose Review Area
Prose Composition Area

Fig. 1. Contextual Chat Conference Tool user interface interaction areas

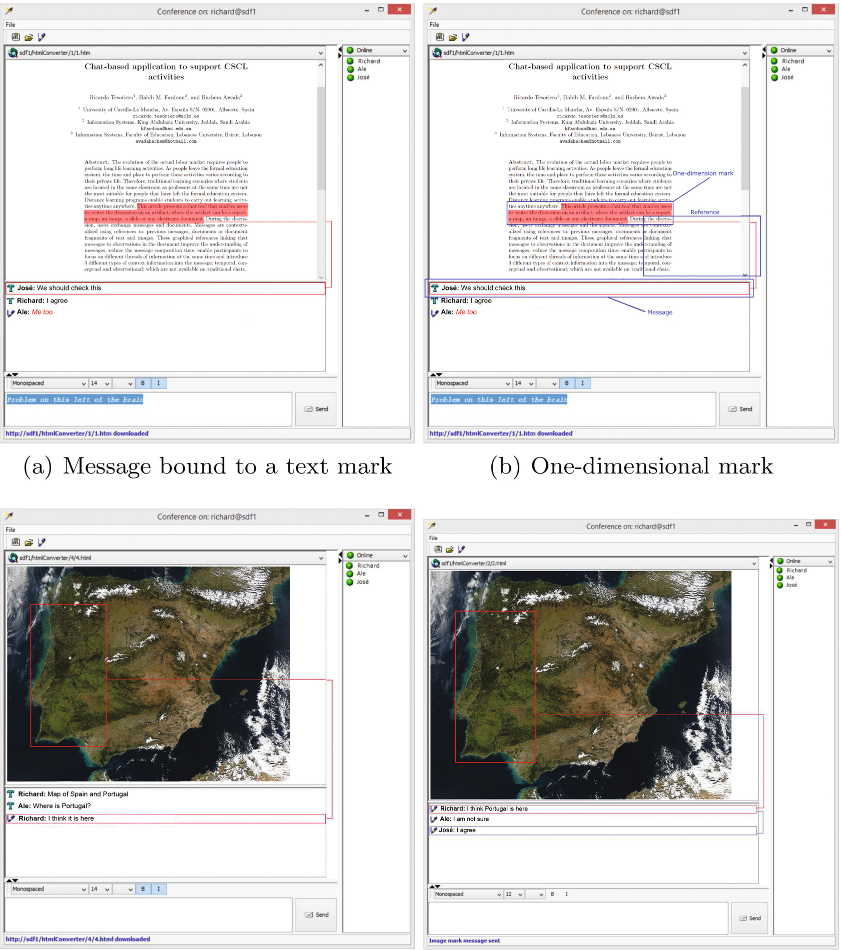
The PRA shows the messages sent by chat participants. When participants click on a message, the reference to a message, or a document region, is displayed accordingly.

The AIA displays the artifact of the conversation related to message observations. It consists of two parts: the document view area showing a document, and the document selection area that enables participants to select the document to be displayed on the document view area. This area renders documents and marks, where a mark is mechanism to identify fragments of documents. The type of mark depends on the media to be referred (i.e. text or image).

The application supports different types of media resources included in the artifacts (i.e. documents); for instance, plain text and compositions of texts and images. These resources are embedded into documents of different formats (ANSI Plain Text [19], Graphics Interchange Format [5], Joint Photographic Experts Group [8], Portable Network Graphic [26], Rich Text Format [3], Microsoft Word format [13], Microsoft Excel [11], Microsoft PowerPoint [12] and references to static HTML [25]).

Regarding messages, the CCCT enables chat participants to improve the message expressiveness by setting the message text attributes, such as the font family, size, colors, bold and italics styles.

Messages can be linked to a document, a region of a document or another message. These links consist of 3 parts: the source, the reference link and the target. While the source is always a chat message, which is defined when the message is composed by the author, the target could be another message, a document or a region of a document. The link reference is represented by a line that binds both, the source and target elements.



Chat-based application to support CSCL activities

Ricardo Tesoriero[1], Habib M. Fardoun[2], and Hachem Awada[3]

[1] University of Castilla-La Mancha, Av. España S/N, 02001, Albacete, Spain
ricardo.tesoriero@uclm.es
[2] Information Systems, King Abdulaziz University, Jeddah, Saudi Arabia
hfardoun@kau.edu.sa
[3] Information Systems, Faculty of Education, Lebanese University, Beirut, Lebanon
awadahachem@hotmail.com

Abstract. The evolution of the actual labor market requires people to perform long life learning activities. As people leave the formal education system, the time and place to perform these activities varies according to their private life. Therefore, traditional learning scenarios where students are located in the same classroom as professors at the same time are not the most suitable for people that have left the formal education system. Distance learning programs enable students to carry out learning activities anytime anywhere. This article presents a chat tool that enables users to center the discussion on an artifact; where the artifact can be a report, a map, an image, a slide or any electronic document. During the discussion, users exchange messages and documents. Messages are contextualized using references to: previous messages, documents or document fragments of text and images. These graphical references linking chat messages to observations in the document improve the understanding of messages, reduce the message composition time, enable participants to focus on different threads of information at the same time and introduce 3 different types of context information into the message: temporal, conceptual and observational; which are not available on traditional chats.

José: We should check this
Richard: I agree
Ale: Me too

Monospaced
14
B
I
Problem on this left of the brain
Send
http://sdf1/htmlConverter/1/1.htm downloaded

(a) Message bound to a text mark

José: We should check this
Richard: I agree
Ale: Me too

(b) One-dimensional mark

sdf1/htmlConverter/4/4.html
http://sdf1/htmlConverter/4/4.html downloaded

Richard: Map of Spain and Portugal
Ale: Where is Portugal?
Richard: I think it is here

(c) Message bound to an image mark

Richard: I think Portugal is here
Ale: I am not sure
José: I agree

(d) Chain of linked messages

Fig. 2. Contextualized chat message samples

Some examples of marks on documents are exposed in Fig. 2. The Fig. 2(a) shows a message bound to a text mark. If the target is a fragment of a document, we employ a mark to set the limits of the region. Thus, we define 2 types of marks: one-dimensional marks and two-dimensional marks.

One-dimensional marks enable chat participants to create lineal marks to enclose parts or sets of lines. This type of mark is the most suitable to mark paragraphs of text or parts of them (see Fig. 2(b). On the other hand, two-dimensional marks enable participants to mark sections of documents as rectangles. This type of mark is the most suitable to mark images (see Fig. 2(c)). Finally, the Fig. 2(d) shows a message to message reference link, where the chain of linked messages shows message marks.

3.2 The Contextual Chat Conference Tool System Architecture

A client-server architecture is employed to implement the system. Clients enable participants to exchange documents and messages linked to fragments of these documents through 2 servers: the Document Converter Server (DCS) and the Message Delivering Server (MDS).

While the MDS is in charge of delivering chat information among participants (messages, presence awareness, etc.), the DCS is in charge of delivering documents to chat participants. Besides, as the client supports different document formats, the DCS is in charge of converting all document formats to HTML to allow the client to treat all documents homogeneously.

Before linking a message to a fragment of a document, participants have to share a document. To share a document, the system executes 2 steps: the document conversion step and the document dissemination step.

Document Conversion. The document conversion step is carried out by the DCS. The Fig. 3 shows the document conversion process which starts when a participant selects a document to share. Then, this document is sent to the conversion service as the body of an HTTP [15] POST request.

The request is received by a Web service implemented using the Java Servlet Technology [9], and it is stored to be processed in an Apache Tomcat [2] Web server running on a Microsoft Windows Server 2003 [21]) operating system.

Once the document is stored, it is converted into HTML according to the document format. Simple documents, such as ANSI text, PNG, GIF and JPG files, are embedded into an HTML file. However, complex documents such as Microsoft Word, Excel and PowerPoint are converted to HTML files using the COM API of native applications [22]. To use the COM API from the Java

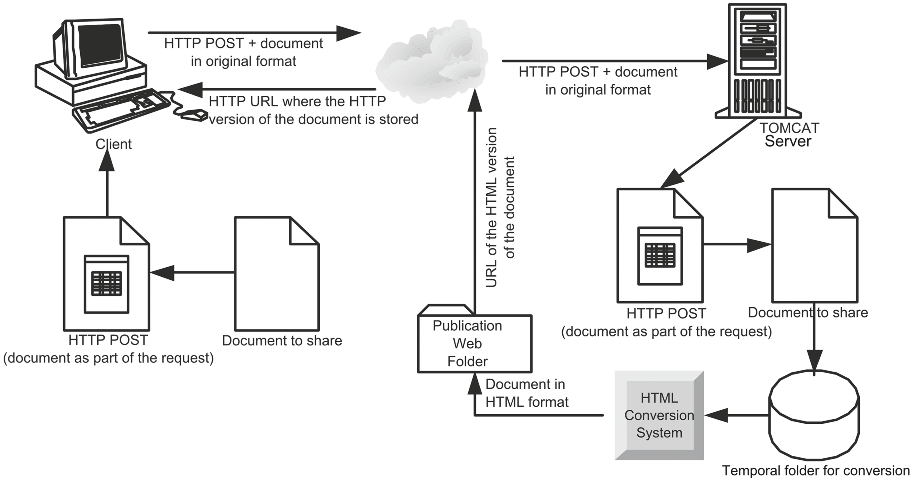


Fig. 3. The document conversion and dissemination processes

programming language, we have employed the JACOB (JAva COM Bridge) API [1] which is based on the JNI [16] (Java Native Interface) technology.

Once the conversion process is finished, HTML files are published on the HTTP server. The response to the request sent from the client contains the URL of the HTML version of the document.

Document Dissemination. Once the document conversion process is finished, the document dissemination process wraps the URL referring to the HTML version of the document within a message, and it is delivered to the rest of the participants through the MDS.

Although the document information is delivered as a message, this information is not shown on the PRA. Instead, it is shown on the combo box that is part of the AIA, where participants choose the document to manipulate in the AIA canvas. Thus, participants are able to select the document from the combo box, and any fragment of the selected document, from the AIA canvas.

The MDS is a Jabber server based on the eXtensible Messaging and Presence Protocol (XMPP) [18]. This protocol allows the bi-directional communication between the server and all clients using XML. The implementation employed to deploy the CCCT is the Openfire [7].

To support the exchange of documents and message references, we have implemented an extension of the XMPP.

The main advantage of the XMPP protocol is the capability to define message extensions to enrich the information sent and received by chat participants.

The Fig. 4 shows the XML document sent by sender@server.com to the room@server.com channel. The Fig. 5 shows the XML that is received by receiver@server.com from the room@server.com. To extend the message contents with extra information we employ the 'x' tag. In order to process the 'x' tag, it defines the xmlns attribute which identifies the type of information that is contained with the tag. The Fig. 6 shows an example of a message received by receiver@server.com that uses a message extension to include information about the time the message was sent because it was not delivered on time.

Therefore, we have defined message extensions using 'x' tags to exchange messages including links to documents, fragments of documents and messages.

```
<message from='sender@server.com/home' id='hfsd76fds'
 to='room@server.com' type='groupchat'>
<body>This is a message</body>
<html>
 <body>
  <p>This is a <b>message</b></p>
 </body>
</html>
</message>
```

Fig. 4. XML content of an XMPP message to be sent

```
<message to=’receiver@server.com/work’ id=’hfsd76fds’
 from=’room@server.com/sender’>
<body>This is a message</body>
<html>
 <body>
  <p>This is a <b>message</b></p>
 </body>
</html>
</message>
```

Fig. 5. XML content of an XMPP message received

```
<message id=’xmkos2yh’ from=’room@server.com/sender’
 to=’receiver@server.com/work’>
<subject>Welcome!</subject>
<body>Welcome to Jabber!</body>
<x xmlns=’jabber:x:delay’ from=’source@server.com/home’
 stamp=’20011206T18:22:09’>
Offline Storage</x>
</message>
```

Fig. 6. XML content of an XMPP message with extensions

The Fig. 7 shows a sample of a message extension that delivers the location of the Testing.doc document that was converted to HTML and stored at the http://localhost:8080/ConverterWEB/files/3/3/3.html. This extension is identified by the http://www.richard.org/docMessage XML namespace. It defines the url, description and action nested tags.

```
<x xmlns=’http://www.richard.org/docMessage’>
 <url xmlns=’http://www.richard.org/docMessage’>
  http://localhost:8080/ConverterWEB/files/3/3/3.html
 </url>
 <description xmlns=’http://www.richard.org/docMessage’>
  Testing.doc (1/1)
 </description>
 <action xmlns=’http://www.richard.org/docMessage’>add</
    action>
</x>
```

Fig. 7. XML content of an XMPP message with a document extension

The url tag defines the location of the HTML version of the document.

The description tag contains information about the original version of the document; for instance, it contains the document file name and the page number of the document. This information is useful because most documents

(PNG, JPG, GIF, Microsoft Word, etc.) are stored in single files; however, presentations in Microsoft PowerPoint and spreadsheets in Microsoft Excel are stored in multiple files (one file for each slide/sheet). Therefore, as the message format supports only one page for each message, the converter delivers as many messages as HTML pages were the result of the conversion. As we have mentioned before, the page number is part of the description of the message.

Finally, the action tag identifies the action of the message (i.e. add document, remove document, update document). The only action that was implemented is the add action. Conceptually speaking, the remove action is a dangerous action to implement because if documents are removed, some messages could lose the context of the document. Another alternative is removing documents jointly with messages that are linked to these documents; however, if we remove messages from the conversation, we lose the temporal context of the conversation messages generating "holes" in the story. The situation regarding the update action is similar to the remove action.

There are different types of message references. The Fig. 8 shows a sample of a message extension to represent an image mark linked to a message. The message that points to the mark is implicitly defined because the 'x' tag is a nested tag of the message tag that contains the message information.

The type of the mark is defined by the xmlns attribute which is set to 'jabber:x:imageMark'. The reference tag defines the document in which the mark is embedded by defining the url attribute that points to the URL of the document being referred.

```
<x xmlns=’jabber:x:imageMark’>
 <reference
  url=’http://localhost:8080/ConverterWEB/files/1/1/1.html’
      xmlns=’jabber:x:imageMark’/>
 <anchor xmlns=’jabber:x:imageMark’>
  <color b=’0’ r=’255’ g=’0’ xmlns=’jabber:x:imageMark’/>
 </anchor>
 <selection xmlns=’jabber:x:imageMark’>
  <background b=’0’ r=’255’ g=’0’xmlns=’jabber:x:imageMark’/>
  <foreground b=’0’ r=’255’ g=’0’xmlns=’jabber:x:imageMark’/>
  <offset xmlns=’jabber:x:imageMark’>23207</offset>
  <rectangle xmlns=’jabber:x:imageMark’>
   <x xmlns=’jabber:x:imageMark’>25</x>
   <y xmlns=’jabber:x:imageMark’>159</y>
   <width xmlns=’jabber:x:imageMark’>287</width>
   <height xmlns=’jabber:x:imageMark’>34</height>
  </rectangle>
 </selection>
</x>
```

Fig. 8. XML content of an XMPP message with an image mark extension

The anchor tag defines the color of the link between the mark on the document and the message which is drawn on the PACA. In the example, the color tag defines the red color by defining the r, g and b attributes to 0, 255 and 0 respectively.

The selection tag defines the image mark on the document. It requires four nested tags to define the image mark characteristics. The background and foreground tags define the background and foreground colors of the image mark using the color nested tag in the same way as the anchor tag does.

As the application supports documents that have text interleaved with images, the offset tag contains the information about the relative image position within the text to identify the image.

Finally, the rectangle tag contains the information of the mark dimension and position using the x, y, width and height inner tags.

The Fig. 9 shows an example of the message extension to represent text marks. As in the image mark, the type is defined by the xmlns attribute, which is set to 'jabber:x:textMark'.

The reference and anchor tags are defined as in the image mark. However, the selection tag defines the start and end tags, instead of the rectangle and offset tags to represent the beginning and the end of the text fragment that represents the mark.

```
<x xmlns='jabber:x:textMark'>
<reference url='http://localhost:8080/ConverterWEB/files
    /1/1/1.html'
 xmlns='jabber:x:textMark'/>
<anchor xmlns='jabber:x:textMark'>
 <color b='0' r='255' g='0' xmlns='jabber:x:textMark'/>
</anchor>
<selection xmlns='jabber:x:textMark'>
 <background b='0' r='255' g='0' xmlns='jabber:x:textMark'/>
 <foreground b='0' r='255' g='0' xmlns='jabber:x:textMark'/>
 <start xmlns='jabber:x:textMark'>9932</start>
 <end xmlns='jabber:x:textMark'>9954</end>
</selection>
</x>
```

Fig. 9. XML content of an XMPP message with a text mark extension

4 Discussion

This section presents how the problems that motivated this work are tackled by this proposal. Besides, it highlights the benefits of employing the CTTT application instead of traditional systems/procedures in the development of eLearning activities.

The discussion is presented from two different perspectives. On the one hand, the user perspective describes the solution conceptually. On the other hand, the technological perspective presents the arguments that leaded the architectural and technological decisions.

4.1 The User Perspective

From the user perspective, the lack of information regarding the learning activity process (1) is a key issue to board when conversations are carried out using audio and video conference systems. To cope with this situation, the system is based on a chat conference system. The benefits of this approach are not only the capability of searching or analyzing specific information in a text-based form, CCCT is also capable of identifying the author of the information being shared. Besides, as observations can be linked, a semantic network could be easily derived from stored information to recreate the learning process.

The overhead of information (2) derived from the description of the artifact in the observation to put it in context is an important issue to deal with in traditional chat conferences because the extra information increases the message composition time, and it makes the observation more difficult to discern from the context information.

The integration of the document sharing and marking capabilities into the chat conference tool enables participants to easily put messages in context. This approach goes further and provides different marking capabilities according to the media (text or image) to improve the expressiveness of the context. Thus, this system reduces the message composition time and increases the precision of the context description reducing the risk of misunderstandings.

Due the media nature in which the audio and video is manipulated, apart from the problems (1) and (2), these communication medias do not allow chat participants to focus their attention on 2 audios or videos at the same time (3). However, the situation becomes more manageable when participants are dealing with simultaneous chats.

There are some studies showing that the late-coming of participants occurs very often depending on the participants' culture. Therefore, it is important to support the possibility of following two threads of conversation at the same time as well as reviewing information that was already shared during the same or different conversation sessions.

Chats seem to be the most suitable communication medium when participants join into a conference that is already in progress because they can follow the actual thread of the conversation while reviewing previous messages and documents in real-time. Another consequence of this approach, is the reduction of information access time which makes the conversation smoother.

The use of different applications which are not coupled to manage observations and artifacts (4) leads to problems, such as the overhead of information (2) and analysis difficulties (3). This approach integrates and relates observations and artifacts into the same user interface to overcome this situation.

Modern document processors support comments on documents. Although most of these comments do not distinguish between comments linked to texts from those linked to graphics, they provide a good mechanism to attach observations to artifacts. However, there is a lack of temporal awareness among observations (5) because comments are linked to the document structure instead of the conversation structure. Consequently, the benefits of the improvement regarding the overhead of information (2) and the difficulty to analyze conversations (3) is diminished by an increment of the loss of information regarding the record of the learning activity process (1) because the importance of what is being said is as important as when it was said, particularly during a review process.

As this approach integrates both views, the temporal as well as the artifact context, they are synchronized to provide a complete view of the conversation. As word processors enable participants to "reply" comments with new comments, observations can be easily related (6) improving the decision making process analysis (1). This feature is also included in our proposal as links to other messages. Thus, message contexts are enriched with information from three different sources: time, observation and artifact context information.

4.2 The Technological Perspective

From a technological perspective, our approach presents some advantages over other alternatives. The first advantage is the low communication bandwidth that is required by the system to work, compared to the amount of resources required to process audio or video conference systems. This feature makes the system more suitable in extreme situations where this resource is poor (i.e. mountains, isles, forests, etc.)

The system deals with two types of processes: the document sharing and the message exchange. As the document sharing process does not support the document edition, the information to be provided does not change over the time; therefore, a simple and worldwide-known stateless communication protocol, such as HTTP, provides the most flexible solution to implement the DCS where Web services are in charge of converting and publishing documents in HTML format on the Internet.

The message exchange process requires extra information involving message and participant states. To implement this requirement, we have chosen the XMPP protocol to fulfill MDS requirements. As XMPP is an open standard, there are many implementations and public servers, and the most of them, are capable of playing the role of the MDS.

Finally, the XMPP requires a full duplex communication channel to keep clients up to date. Although HTML 5 [28] provides messaging capabilities, they are not are fully supported by some browsers. Therefore, we have employed Java to develop the client application to implement a multi-platform solution.

5 Conclusions and Future Work

This paper presents a chat-based application to support CSCL activities in order to improve the communication and the analysis of collaborative conversations when the focus of these conversations are artefacts such as reposts, images, etc.

The proposed solution is based on a chat conference system that integrates observations with artefacts. It enables chat participants to link observations to an artefact in a graphic way. As result of this action, users obtain: an improvement of message understandings, a reduction of the message composition time, the capability to focus on different threads of information at the same time, and the introduction of 3 different types of context information into a message: temporal, conceptual and observational. The temporal context information defined by the chat message sequence, the conceptual context information defined by links between messages and the observational context information defined by the links between messages, and the artefact which is the focus of the observation.

A usability evaluation based on heuristics was performed on this tool by usability professionals which concurred in the good performance of the application.

Regarding future works, we are actually working on a new version of the tool clients based on the HTML 5 [27] standard introducing the Web Messaging (or cross-document messaging) API [28] to support message delivering. Besides, we are working on supporting PDF documents which are becoming a standard on the field. Finally, we are also working on the implementation of a mobile version of the tool client in HTML 5 [27].

References

1. Adler, D.: The JACOB Project: A JAva-COM Bridge (2014). http://danadler.com/jacob/. Accessed 12 May 2014
2. Apache Software Foundation: Apache Tomcat Web site (2014). http://tomcat.apache.org/index.html. Accessed 12 May 2014
3. Biblioscape 9: Rich Text Format (RTF) Version 1.5 Specification (2014). http://www.biblioscape.com/rtf15_spec.htm. Accessed 12 May 2014
4. Churchill, E.F., Trevor, J., Bly, S., Nelson, L., Cubranic, D.: Anchored conversations: chatting in the context of a document. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2000, NY, USA, pp. 454–461 (2000). http://doi.acm.org/10.1145/332040.332475
5. CompuServe Incorporated, World Wide Web Consortium (W3C): Graphics Interchange Format (2014). http://www.w3.org/Graphics/GIF/. Accessed 12 May 2014
6. Ellis, R.K.: Field Guide to Learning Management Systems (2009)
7. Ignite Realtime: Openfire. Jive Software (2014). http://www.igniterealtime.org/projects/openfire/. Accessed 12 May 2014
8. Independent JPEG Group (IJG): JPEG Homepage ISO/IEC IS 10918–1. ITU-T Recommendation T.81. (2014). http://www.jpeg.org/jpeg/. Accessed 12 May 2014
9. Java Community Process: SR-000340 Java TM Servlet 3.1. Java Community Process Web site (2014). https://jcp.org/aboutJava/communityprocess/final/jsr340/. Accessed 12 May 2014
10. Johansen, R.: Groupware: Computer Support for Business Teams. The Free Press, Macmillan Inc., New York (1988)
11. Microsoft Corporation: Microsoft Excel. Microsoft Web site (2014). http://office.microsoft.com/es-es/microsoft-excel-software-de-hoja-de-calculo-FX010048762.aspx. Accessed 12 May 2014

12. Microsoft Corporation: Microsoft Power Point. Microsoft Web site (2014). http://office.microsoft.com/es-es/microsoft-powerpoint-software-de-presentacion-de-diapositivas-FX010048776.aspx. Accessed 12 May 2014
13. Microsoft Corporation: Microsoft Word. Microsoft Web site (2014). http://office.microsoft.com/es-es/microsoft-word-software-de-procesamiento-de-texto-y-documentos-FX010048798.aspx. Accessed 12 May 2014
14. Moodle: Moodle web site (2017). https://moodle.org/. Accessed 12 Jan 2017
15. Network Working Group: Hypertext Transfer Protocol - HTTP/1.1. RFC 2616 (1999). http://www.w3.org/Protocols/rfc2616/rfc2616.html. Accessed 12 May 2014
16. Oracle: Java Native Interface (2014). http://docs.oracle.com/javase/7/docs/technotes/guides/jni/. Accessed 12 May 2014
17. Penichet, V.M.R., Marin, I., Gallud, J.A., Lozano, M.D., Tesoriero, R.: A classification method for CSCW systems. Electron. Notes Theoret. Comput. Sci. **168**, 237–247 (2007). http://dx.doi.org/10.1016/j.entcs.2006.12.007
18. Shigeoka, I.: Instant Messaging in Java: The Jabber Protocols. Manning Publications, San Diego (2002)
19. The Unicode Consortium: The Unicode Consortium Web site (2014). http://www.unicode.org/. Accessed 12 May 2014
20. Whittaker, S., Brennan, S.E., Clark, H.H.: Co-ordinating activity: an analysis of interaction in computer-supported co-operative work. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1991, NY, USA, pp. 361–367 (1991). http://doi.acm.org/10.1145/108844.108944
21. Wikipedia: Microsoft Windows Server 2003 (2003). http://es.wikipedia.org/wiki/Windows_Server_2003. Accessed 12 May 2014
22. Wikipedia: Component Object Model (2014). http://es.wikipedia.org/wiki/Component_Object_Model. Accessed 12 May 2014
23. Wikipedia: Microsoft Docs (2014). http://en.wikipedia.org/wiki/Google_Docs. Accessed 11 May 2014
24. Wikipedia: Microsoft Net Meeting (2014). http://en.wikipedia.org/wiki/Microsoft_NetMeeting. Accessed 11 May 2014
25. World Wide Web Consortium (W3C): HTML 4.01 Specification (1999). http://www.w3.org/TR/html401/. Accessed 12 May 2014
26. World Wide Web Consortium (W3C): Portable Network Graphics (PNG): Functional specification. ISO/IEC 15948: 2003 (E) (2003). http://www.w3.org/TR/PNG/. Accessed 12 May 2014
27. World Wide Web Consortium (W3C): HTML 5.1 Nightly. a vocabulary and associated APIs for HTML and XHTML (2014). http://www.w3.org/html/wg/drafts/html/master/. Accessed 12 May 2014
28. World Wide Web Consortium (W3C): HTML5 web messaging (2014). http://www.w3.org/TR/webmessaging/. Accessed 12 May 2014