

# A Real-time Collaboration-enabled Mobile Augmented Reality System with Semantic Multimedia

Dejan Kovachev, Goekhan Aksakali and Ralf Klamma  
Advanced Community Information Systems (ACIS) Group  
RWTH Aachen University  
Ahornstr. 55, 52056 Aachen Germany  
{kovachev|goekahan|klamma}@dbis.rwth-aachen.de

**Abstract**—Mobile Augmented Reality (MAR) enables overlays of semantically-enriched multimedia on video streams of smart phone cameras. These new ways to interact with digital objects via position and 3D movement can be helpful for on-site professional communities. However, current MAR applications lack real-time collaborative features. Moreover, blending multimedia semantics in collaborative MAR is still challenging. In this paper, we present a mobile real-time semantic multimedia-based collaborative system with augmented reality features based on open standards like XMPP and MPEG-7. The prototype system is evaluated in the digital documentation of historical sites for cultural heritage management. The evaluation results indicate increased productivity and awareness within on-site professional communities.

**Index Terms**—real-time collaboration; augmented reality; mobile multimedia; multimedia metadata; MPEG-7; XMPP;

## I. INTRODUCTION

Mobile augmented reality becomes increasingly feasible on inexpensive hardware at mass-market effect. Augmented Reality (AR) is a natural complement to mobile computing, since smart phones can change their user interface so that the physical world becomes a part of the the user interfaces itself. Accessing and understanding information related to the real world becomes easier. This has led to a widespread commercial adoption of MAR in many domains like gaming, cultural heritage, assisted directions, marketing, shopping, education and instruction.

Furthermore, in order to support diverse digital content several popular MAR applications like Layers and Wikitude have shifted from special-purpose applications into MAR browsers which can display third party content. Third party content providers use predefined APIs which can be used to feed content to the MAR browser based on context parameters. Hence, they are relieved from the technical burden of managing AR applications.

However, such MAR browsers have limited support for mobile real-time collaboration (MRTC), i.e. to enable users to collaborate and communicate with each other in real-time. Most MAR browsers operate on common service oriented approaches like REST, SOAP and HTTP which are inadequate for real-time collaboration. Moreover, these approaches provide only full

functionality, i.e. the request for data transmission is actively initiated by the client without genuine two-way peer-to-peer communication. Moreover, the semantics of digital objects (i.e. multimedia) in MAR is barely considered beyond the location parameters. A MAR system for collaborative actions on common data and display that collaboratively edited multimedia and metadata with the assistance of augmented reality can increase the user experience in many professional domains enormously.

We developed a mobile real-time collaborative system for multimedia that performs the collaboration on an open, customizable XMPP-based lightweight framework. It provides basic components for building mobile collaborative applications. The system, called XMPP-based Mobile Multimedia Collaboration (XmmC), in a broader sense, provides a set of services so that mobile clients can perform multimedia acquisition, real-time collaborative annotation and at the same time blend these operations within a MAR browser. We also define XMPP protocol extensions (XEPs) for these services to fulfill the effective communication between client and server side. The shared data for augmented reality is stored and shared as custom XML data in the augmented reality markup language (ARML) [1]. Furthermore, we define new XEPs for exchanging the AR related data and managing multimedia and annotating the related multimedia among community members. Moreover, XmmC integrates with existing MPEG-7 multimedia services, thus achieving interoperability with the MPEG-7 [2] multimedia standard. Thus, it takes advantage of large existing multimedia metadata repositories.

We evaluated the proposed architecture as a collaborative annotation tool that can assist professionals in cultural heritage for digital documentation of historical sites based on semantic multimedia and annotation metadata. The evaluation considered the system performance, i.e. mobile collaborative annotation services, multimedia data management, energy impact and sensor accuracy. Additionally, user study on the experience of the proposed system was conducted too. The results show increased cultural heritage awareness via the proposed MAR tools using inexpensive hardware such as smart phones.

In the rest of the paper, we first review the related research

work (Section II). Then in Section III we describe our approach to MAR with real-time collaboration and multimedia semantics. We explain the prototype system and its design in Section IV. In Section V we introduce the evaluation setup and the corresponding results. Finally, we draw conclusions and refer to the future work.

## II. RELATED WORK

### A. Mobile Collaboration

Technology could assist a group of people to communicate, manipulate and modify shared digital objects in a coherent manner [3], [4]. Popular groupware software applications are based on simultaneous writing of a document by different authors, also known as shared file editing, instant chatting or cooperative design. Collaborative editing is the practice of participants within a group, work together to produce simultaneously a common output using a set of defined operations [5]. A lot of previous research deals with conflict resolution and avoidance in a concurrent collaborative editing session, i.e. when two or more users access the same document and perform simultaneous incompatible operations [5], [6].

Since XML is de-facto the standard interchange and data format, systems that support real-time collaborative editing of XML documents could lead to a general solution for many application areas especially real-time collaboration. Furthermore, XML is widely used for storing and exchanging multimedia metadata because its tree like hierarchical structure can describe the complexity of metadata descriptors which can be interchanged with different systems [7].

Currently, Operational Transformation (OT), first introduced by Ellis and Gibbs [8] then simplified by Jupiter System [9] is the most popular technique behind collaboration features. In OT, every client has a replicated copy of the document and sends the operations to the server side with an optimistic approach therefore the changes that are made on client side are reflected before sending them to the server. The main idea of the algorithm executing the locally generated operations without and delay and transforming remote editing operations into a new form according to the effects of previously executed concurrent operations and ensuring after every operation the consistency of the document at all participants [10]. Many research works improved the OT approach. dOPT [8], Jupiter [9], SOCT2 [11], GOT [12] focus on linear structured text documents, whereas OT for SGML [13] treeOpt [14] and P2P Editing on XML-like Trees [5] are some examples that performs OT on tree like structures. Furthermore, COT [15] and adOpted modified the algorithm by adding undo functionality. Google Operational Wave Transformation assures concurrency control of XML documents [16]. Consistency Maintenance Algorithm for XML (CMAX) [6] is a lightweight approach that focuses on ensuring collaborative editing on structural XML-based documents. The algorithm mainly inspired from operational transformation for structural documents. CMAX follows the same principles of OT for solving divergence and causal violation, however for solving intention violation problem a new approach is applied. The OT update operation is not used since Gerlicher [6] showed that the

probability of modification of the same tree node concurrently is relatively small when the XML tree is averaged size or big size. Therefore, CMAX delegates conflict resolution to the user instead of combining the attribute values by OT.

### B. Communication for Collaborative Applications

The right set of underlying communication protocol is crucial in the mobile real-time collaboration. The XMPP protocol provides a pure XML foundation for real-time messaging, opening up tremendous possibilities for more advanced real-time applications. XMPP together with its extensions is a powerful protocol for collaborative services. Together they demonstrate several advantages beyond traditional HTTP-based Web services ( e.g. SOAP and REST), such as decentralized, open and flexible (extensible) communication protocol, federation of servers, support for real-time data streaming in two directions, event notifications, remote procedure calls, and multimedia session management. Asynchronous invocation eliminates the need for ad-hoc solutions like polling.

Google Wave protocol is an excellent example of XMPP-based communication and collaboration platform for concurrently editable structured documents and real-time sharing between multiple participants. Novell Vibe Cloud [17] is a web-based social collaboration platform for the enterprise providing social messaging and the ability of online document co-editing along with file management, groups, profiles, blogs and wikis, and security and management controls. Both Google Wave Protocol and Novel Vibe are very sophisticated collaborative editing software, but they rely on heavy-weight client JavaScript libraries which limits the usefulness for custom mobile applications.

On the other hand, the Collaborative Editing Framework for XML (CEFX) enables lightweight concurrent real-time editing of XML files using operational transformation algorithms [6]. Since the nature of XML is generic and extendable different kind of information can be stored such as graphic files (SVG), AR contents (ARML) etc. Voigt [18] further extended the framework by changing the communication protocol from JAVA RMI to XMPP. Moreover, communication data volume reduced significantly.

Similar to the Mobilis framework [19], our work also uses the CEFX+ for providing collaborative editing services and providing the communication between the nodes that attends a collaboration session on XMPP. However, the Mobilis Platform is heavy-weight at both client and server sides and also the platform lack of mobile AR browsing features. Junction [20] is another XMPP-based communication framework for multi-device applications, but with limited collaborative features.

### C. Mobile Augmented Reality Browsers

Höller and Feiner [21] give a detailed introduction to MAR systems and review some important MAR system considerations. Layar [22] is a mobile outdoor AR platform for discovering surrounding of the user. The platform displays the physical world by augmenting it with the digital retrieved information called “layers” via a mobile device. The platform

not only provides a location based augmented reality solution, but also an API for third party developers that them to create, maintain and publish “layers” that extend the application for their own purposes [22]. However, the major drawback of Layar is the flow of information which is only one way. The client can not push other data beside the location and some parameters which conflicts with collaborative applications. You et al. [23] tried to overcome such limitations of pull-based MAR browsers in their Mixed Reality Web Service platform. Their platform provides a RESTful interface for 3rd party geospatially oriented content. However, their workaround solution of a publish/subscribe system with periodical polling suffers from the same limitations. In contrast, our XmmC framework is build around XMPP protocol extensions, thus ensuring simple and effective bi-directional XML-based communication.

### III. THE XMMC COLLABORATIVE MAR APPROACH

Augmented reality and basic multimedia metadata, semantic annotations are stored as XML document for collaborative editing. Our collaborative AR metadata editing service uses the consistency maintenance algorithm for XML (CMAX) [6] algorithm. The synchronization is done by keeping a copy of the XML at every client in the session and then in case of edit operations the service ensures timely updates on the copies. The updates are performed by sending the changes with a message “propagateLocalOperation” to the server which will apply the changes to the server copy. If some conflicts exists, resolves them and broadcasts the changes with a message to be made to have consistent versions of the document on all clients.

Figure 1 demonstrates the data model which is persisted by XmmC.

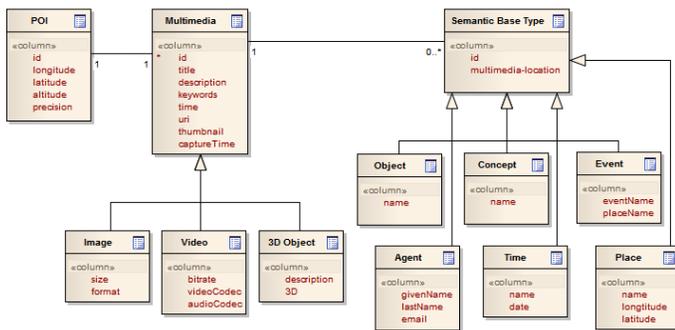


Fig. 1. XmmC ER diagram of multimedia metadata. The multimedia semantics is expressed as MPEG-7 Semantic Descriptors

The system uses data type POI for AR purposes. Therefore, users can query the POIs according to the geographical location. The altitude field is used for determining where POIs are displaying on the augmented reality. Every POI has a reference to a multimedia.

A multimedia artifact can be any kind of multimedia data, which can be rendered by the mobile client, e.g. video, images and 3D objects. Title, description and keywords form the basic

metadata about the multimedia whereas URI specifies where the multimedia is available and thumbnail keeps the reference to the thumbnail of the multimedia.

Every multimedia artifact can have multiple semantic base types which are used for annotation. Integration with our existing MPEG-7 services [24] is one of the goals of this work. The annotation model is defined according to the semantic base type definition used at MPEG-7. This integration provides access to large repository of already semantically-enriched multimedia.

Figure 2 shows screenshots of the XmmC mobile AR browser. On the left is the camera preview augmented with nearby POI and their related multimedia artifacts. On the right is the interface with MPEG-7 semantic annotations which can be created, edited and deleted with other mobile clients collaboratively.

### IV. XMPP-BASED MOBILE MULTIMEDIA COLLABORATION IN AR

In this section, the design and realization the XmmC system, i.e. the system architecture, system interaction, server and client prototypes, are discussed.

#### A. System Architecture

XmmC is designed as a services oriented architecture. By applying a multilayer and modular design approach, we achieve a flexible easy-maintainable platform for developing XMPP based mobile multimedia collaborative applications.

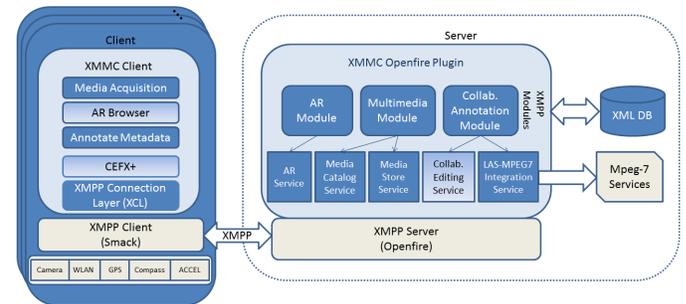


Fig. 3. XmmC system architecture

Figure 3 shows that multiple XmmC clients can communicate with a XmmC server over the XMPP protocol. The client application requires mobile devices with GPS, compass, accelerometer, WLAN and camera. Android was selected as mobile software platform for this prototype. It allows many already developed Java libraries or Java projects to be imported into the Android applications.

As mentioned before, XMPP is the main communication protocol used in or system. Since, Smack [25] is most mature XMPP client library written in Java, it is used for XMPP communication with the XMPP server. The XMPP connection layer (XCL) is responsible for receiving and sending XML stanzas by using the Smack library. CEFX component is used for synchronization of the XML metadata. The Android



Fig. 2. XmmC MAR browser

application operates both as a MAR browser and provides features to users for media acquisition and metadata annotation.

On the server side the XMPP communication with clients is enabled by an XMPP Server [26]. The XmmC server actually acts as an XMPP client considered from the XMPP Server. This simplifies the software development and maintenance since both the mobile client and XmmC server can use the same communication components built atop Smack. XMPP Modules are responsible for XMPP communication with the clients. There are three main XMPP modules:

**AR Module** provides AR related services such as retrieving the POIs according to the given restrictions. AR module uses AR services and Media Catalog Services in background to provide its functionalities. The AR module is based on the Mixare Augmented Reality Engine [27], a free open-source mobile augmented reality browser.

**Multimedia Module** is responsible for transfer services like uploading, retrieving and deleting multimedia. Multimedia module utilizes media catalog service that persists the basic multimedia related data on a relational database and enables the client applications to maintain this type of data. Furthermore, the module notifies all the clients that subscribed to XmmC PubSub node by publishing multimedia events.

**Collaborative Annotating Module** handles metadata related services, i.e. metadata management and synchronizing the annotation metadata in real-time among the client applications by using Collaborative Editing Service. Collaborative Editing Service is based on CEFX+ and is used for synchronization of metadata. During the synchronization process, a copy is also situated at MPEG-7 Integration Service so that the service would act as a client that synchronizes the metadata XML document and calls the related MPEG-7 multimedia content and Semantic Base Type Services whenever a XML document is updated.

### B. System Interaction

We show here the XMPP communication and workflow between the XMPP modules and underneath services. We classified interactions under three main groups based on the purpose namely multimedia, AR and collaborative annotation.

1) *Multimedia Interaction*: XmmC offers functionalities to acquire, share and view multimedia. Figure 4 illustrates the sequence diagram how mobile client application insert

a multimedia to the media catalog and by uploading media content to the media store using the XMPP protocol. First, basic media description like title, description, location etc. is sent via `<mmedia-insert>` custom set IQ, but the actual content resides at the client side. Once the multimedia XMPP module receives a stanza, it forwards the request to media catalog service. If no error occurs responds back with a `<mmedia-transfer>` get-IQ which demands actual file transfer. Immediately the client initiates the transfer of the file via XEP-096 SI File transfer extension [28]. When the actual transfer finishes, the client sends back a `<mmedia-transfer>` result-IQ that informs the server that transfer completed successfully. Finally, the server side sends back result-IQ `<mmedia-insert>` that verifies that inserting of multimedia is completed successfully. The XMPP module also sends a PubSub publish set-IQ that notifies other mobile clients when a new media is inserted.

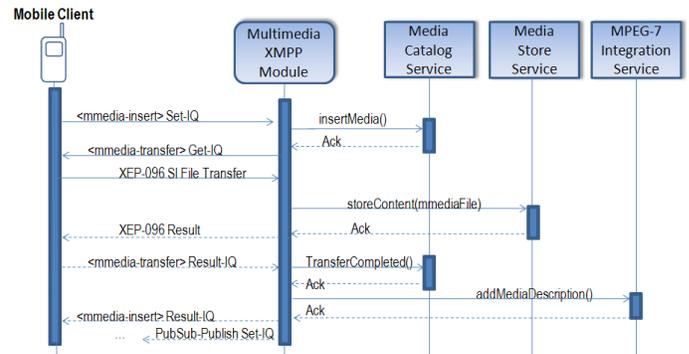


Fig. 4. Sequence diagram - Multimedia insertion

After media is inserted to the multimedia catalog, retrieving it is also possible. The workflow for retrieving multimedia is similar to inserting multimedia.

XmmC also allows modification of basic media description. The workflow is initiated by sending a `<mmedia-update>` set-IQ by mobile client with payload of updated basic media description. After the XMPP module receives the request, it asks media catalog service for modifying the catalog entries. After getting the acknowledgement from media catalog service, the module also sends a request to MPEG-7 Integration Service to update corresponding media description at the metadata

```

Listing 1. Examples of set and result IQ stanzas for Mmedia-insert operation
<iq id="jOE9e-33"
to="xmmc@merian.informatik.rwth-aachen.de/Smack"
from="alice@merian.informatik.rwth-aachen.de/Smack"
type="set">
<mmedia-insert
xmlns="http://dbis.rwth-aachen.de/xmmc/mmedia-insert">
<id>b48ec4b4-e5ea-493f-acce-16a52a60ffa0.jpg</id>
<name>Trojan Horse</name>
<keywords>horse , trojan</keywords>
<description />
<latitude>50.77816836</latitude>
<longitude>6.061606090000001</longitude>
<altitude>2</altitude>
<accuracy>23</accuracy>
<captureTime>131912599814</captureTime>
<mediaType>Image</mediaType>
</mmedia-insert>
</iq>

<iq id="jOE9e-33"
to="alice@merian.informatik.rwth-aachen.de/Smack"
from="xmmc@merian.informatik.rwth-aachen.de/Smack"
type="result">
<mmedia-insert
xmlns="http://dbis.rwth-aachen.de/xmmc/mmedia-insert">
<id>b48ec4b4-e5ea-493f-acce-16a52a60ffa0.jpg</id>
</mmedia-insert>
</iq>

```

repository. Finally, the XMPP module sends back `<mmedia-update>` result-IQ and also sends a PubSub notification to other participants that terminates the workflow.

Listing 1 shows the content of `<mmedia-insert>` set and result IQ stanzas. The payload of the set stanza consists of basic multimedia description. *Id* field is used for uniquely identifying a multimedia artifact. *Name*, *description* and *keywords* are part of the semantic description. *Latitude*, *longitude* and *altitude* (in terms of meters) describes the geographical location where the multimedia is taken where as *accuracy* refers to the accuracy of the geographical location. Another important element is *captureTime* which refers to the time point that the multimedia is created. The last field *mediaType* distinguishes different media types like image, video, etc. Result IQ has only *id* that represents the multimedia.

**Multimedia PubSub Notifications.** Since, XmmC prototype promised to get real-time updates, it is significant to notify the other users about multimedia events like inserting a new, deleting an existing or updating one. If our system would have no push functionality for notifications then the prototype either had to do polling or the data won't be consistent after those changes. Polling is a undesirable solution since it will lead to bad performance and extra communication traffic. On the other hand, the changes that are generated on client should be propagated in order to ensure consistency of the overall system. As aforementioned, PubSub ensures data freshness with less communication traffic with publish and subscribe mechanisms.

First mobile clients have to subscribe to "xmmc" PubSub node for retrieving notifications. Whenever data is modified on the server side after `<mmedia-insert>`, `<mmedia-update>` or `<mmedia-delete>` set-IQ stanza, multimedia XMPP module sends a PubSub set-IQ message with the payload of multimedia

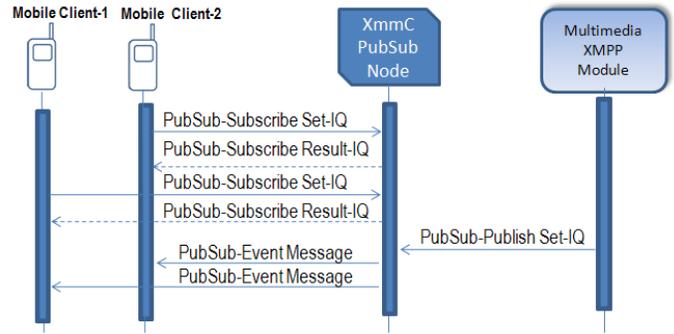


Fig. 5. PubSub notifications

id and the event that is associated with the multimedia. Afterwards, the published information is broadcasted to all subscribers by the PubSub node. The sent stanza is a PubSub notification message with a similar payload to the published stanza.

2) *AR Interaction:* Browsing surroundings augmented with the multimedia is among the main use cases. Therefore, XMPP-based AR services were developed for this prototype which exchange content represented in the Augmented Reality Markup Language (ARML). ARML is a specification that enables content developers to create content that can be displayed on various mobile AR browsers. ARML is originated from KML (Keyhole Markup Language) by reducing some tags that are not relevant for AR and adding some new features. An ARML file consists of two sections. The first section provides information about the content provider that the POIs are related to. At this section, provider's name, description, logo, URL, icon and related tags are specified.

Figure 6 demonstrates the fundamental service calls which are requested by the mobile client application and responded by the AR XMPP Module. Since XMPP is the core protocol used in this work, the communication between client and server side accomplishes via custom XMPP IQs starting with `<ar-...>`. The child elements at `<ar-...>` are declared in the namespace `http://dbis.rwth-aachen.de/xmmc#services/ARService`.

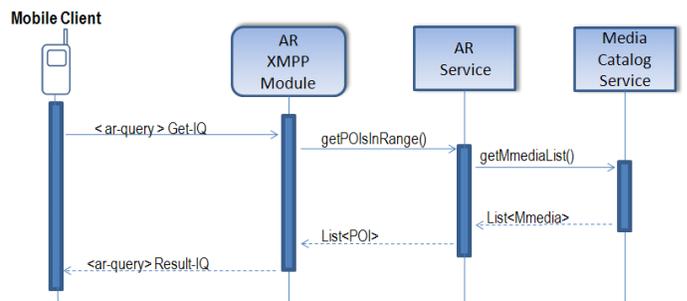


Fig. 6. AR Service sequence diagram

Listing 2 illustrates a get `ar-query` custom IQ stanza with

Listing 2. Ar-query Get IQ Stanza Example

```
<iq type='get' id="XcupS-1"
  from='alice@merian.informatik.rwth-aachen.de/Smack'
  to='xmmc@merian.informatik.rwth-aachen.de/Smack'>
  <ar-query
    xmlns='http://dbis.rwth-aachen.de/xmmc#services/AR'>
    <latitude>50.77816836</latitude>
    <longitude>6.0616060</longitude>
    <radius>800</radius>
  </ar-query>
</iq>
```

an example. The client side issues a get IQ *ar-query* stanza with the location restriction parameters *latitude*, *longitude* and *radius* that conforms to a circular geographical area with a center point of the circle is denoted by *latitude*, *longitude* and radius of the circle denoted by the *radius*.

Listing 3. Ar-query Result IQ Stanza Example

```
<iq type='result' id="XcupS-1"
  from='xmmc@merian.informatik.rwth-aachen.de/Smack'
  to='alice@merian.informatik.rwth-aachen.de/Smack'>
  <ar-query
    xmlns='http://dbis.rwth-aachen.de/xmmc#services/AR'>
    <ar-xml xmlns:ar="http://www.openarml.org/arml/1.0"
      xmlns:wiktude="http://www.openarml.org/wiktude/1.0">
    <ar:provider id="dbis.rwth-aachen.de/XMMC">
    <ar:name>XMMC</ar:name>
    <ar:description>
      XMPP-based Multimedia Collaboration
    </ar:description>
    <wiktude:providerUrl>
      http://dbis.rwth-aachen.de/~goekhan/xmmc/
    </wiktude:providerUrl>
    <wiktude:tags>
      media, real-time, collaboration
    </wiktude:tags>
    </ar:provider>
    <ar-placemark id="16a52a60ffa0.jpg">
    <ar:provider>dbis.rwth-aachen.de/XMMC</ar:provider>
    <name>Battleship Averof</name>
    <description>
      with 3D animation of battleship in aciton
    </description>
    <wiktude:info>
    </wiktude:thumbnail>
    http://.../MediaRepository/files/16a52a60ffa0-thmb.jpg
    </wiktude:thumbnail>
    <wiktude:url>
    http://.../MediaRepository/files/16a52a60ffa0-thmb.jpg
    </wiktude:url>
    </wiktude:info>
    <Point>
    <coordinates>6.1334,50.5036,12</coordinates>
    </Point>
    </ar-placemark>
    <ar-placemark id="3b90aa319d44.jpg">
    ...
    ...
    </ar-placemark>
  </ar-xml>
</ar-query>
</iq>
```

After the AR Module receives the request it retrieves the POIs which are inside the restriction area and sends back a result IQ stanza with a ARML element composes of POIs that provides basic information. Listing 3 shows result *ar-query* IQ stanza.

3) *Collaborative Annotation Interaction*: Xmmc is also capable of annotating multimedia collaboratively in real-time

based on XMPP. Figure 7 demonstrates the sequence diagram that specifies the workflow of the annotation of the multimedia collaboratively. First, mobile client-1 sends a joinSession *<jabber:iq:rpc>* set-IQ with parameters of sessionId of the multimedia and clientId. Then, the Collaborative Annotation Module decrypts these parameters and calls joinSession of Collaborative Editing Service. The service creates a collaboration session and also a multi user chat room for the session and returns generated SessionData. SessionData carries session information like MUC room name, participant client ids etc. Afterwards, the module decrypts the SessionData and sends back RPC result IQ with the return value. As soon as the mobile client receives the SessionData, the client joins the session room that is generated at the server side and requests XML document that carries semantic multimedia annotations by sending loadDocument *<jabber:iq:rpc>* set-IQ with a parameter of corresponding documentId. Similar to the previous RPC request, the XMPP module decrypts the parameter and calls Collaborative Editing Service to retrieve DocumentData which comprises the DOM structure of shared XML, state vector and history buffer. Once, the client receives DocumentData from the XMPP module, it can start to annotate multimedia in the collaborative editing session.

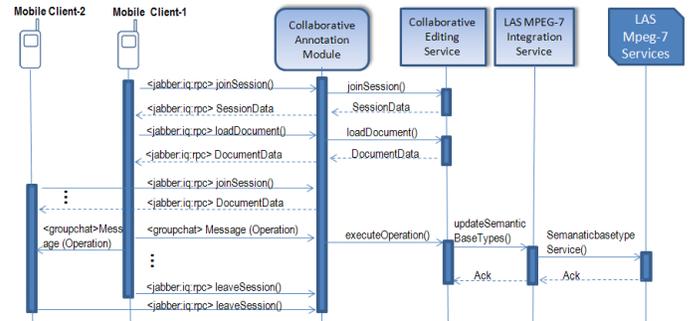


Fig. 7. Sequence Diagram - Collaborative annotation

Mobile client-2 also wants to participate to the collaborative annotation session and it goes through the same process as mobile client-1. Currently, both clients joined the collaborative annotation session. Operations made by each client are sent to all participants of the session via a *<groupchat>* message of the XEP-0045: Multi-User chat extension [29]. The XML document at server side is also updated via calling *executeOperation()* method of the collaborative editing service. Clients can send various update operations during the collaborative annotation. The workflow ends when clients leave the collaboration session by sending a leaveSession *<jabber:iq:rpc>* set-IQ stanza.

**Annotation Operations.** Operation message stanzas represent the information that is needed to be provided for the CMAX algorithm. In the following the abbreviations that are used at operation messages and their corresponding semantics are explained.

- *p*: position identifier of the parent node at XML Document which is defined at *cefx:uid* attribute

Listing 4. Insert operation message stanza

```
<message id="8Dpbh-26"
to="xmme@merian.informatik.rwth-aachen.de/Smack"
from="3b90aa319d44@conference.merian...de/test_1"
type="groupchat">
<body>
<basetype type="time" cefx:uid="101">
<name cefx:uid="201">
Second World War</name>
<date cefx:uid="301">
1997-09-24T00:00:00F1000+01:00
</date>
</basetype>
</body>
<x xmlns="jabber:x:cefx#ins"
p="100" ci="1" sv="0,1"
ba="1" fn="null"/>
</message>
```

Listing 5. Update operation message stanza

```
<message id="8Dpbh-27"
to="xmme@merian.informatik.rwth-aachen.de/Smack"
from="3b90aa319d44@conference.merian...de/test_1"
type="groupchat">
<body/>
<x xmlns="jabber:x:cefx#us" p="201" ci="1" sv="0,2"
ba="1" nt="txt" text="Second_World_War" fn="null"/>
</message>
```

- *ci*: client ID of the operation sender
- *sv*: state vector at the client which is originated before operation is locally executed
- *ba*: before or after, 0 for before and 1 for after
- *fn*: position identifier of the fix node at XML Document which is defined at *cefx:uid* attribute
- *nt*: node type possible values *txt* for text node or *attr* for attribute
- *ins*: insert operation
- *del*: delete operation
- *us*: update state operation

There are two main XML nodes *body* and *x*. Body element as seen at Listing 4 carries string representation of a new XML node with sub attributes and nodes. On the other hand, *x* node consists of operation specific attributes. It is crucial to have minimum payload for various operations in order to fulfill minimum battery consuming behavior. Therefore, only required information is transmitted with the stanzas. For instance as shown at Listing 5, the payload of the stanza consists of only atomic modified node instead of sending modification events with the whole corresponding semantic annotation.

## V. EVALUATION

We evaluated the system in terms of performance and user experience in a mobile environment. The evaluation process was separated in two main parts. i.e. technical evaluation and user experience evaluation.

Contemporary real-time collaboration systems run well and consistent with Web-based or desktop clients. However, ensuring real-time responsiveness within mobile network settings can be problematic due to the unstable, low-bandwidth, high-latency mobile network connections. In this part, we evaluate the XMPP-based CEFX framework for collaborative annotation

context in mobile settings. We can divide evaluation process of this section into two parts. First, performance test for sending and executing remote updates were conducted and the results were analyzed. Second, we examined the framework for conflict resolution and consistency maintenance in simulated mobile network settings.

In the context of performance test, the test suit used for the previous section is used except that we used at 2 mobile phones. We measured the time passed during a mobile client's generation of an operation with adding, modifying or deleting a semantic annotation and sending the generated operation to the other mobile client until corresponding operation is executed at the client. In the test scenario, the client prototype has generated 10 insert, 10 update and 10 delete operations.

The average time was 412 ms with standard deviation of 209 ms based on on the collected performance values. Both average and deviation values are acceptable in a mobile real-time collaboration scenario. However, these values depend on network characteristics too. We have done our tests with a WLAN connection, the process times can be longer with different mobile networks like GPRS, EDGE, UMTS, or LTE.

We also examined the framework for conflict resolution and consistency maintenance at annotation scenario in low bandwidth high-latency networks. In order to provide such network settings, we used the Android emulator. The emulator was set network settings with 900 ms fixed latency and 1.2 KB/s download and upload speed. The test set ups consisted of a client prototype on the emulator and another one on mobile device used at previous tests. First, we tried to delete the same annotation concurrently from both clients. Both operations were sent to the server however, the operation from emulator arrives to the server after the operation of mobile device was executed. The framework resolves the issue by discarding the message from the emulator. Another possible conflict, is to modify the annotation, which is also concurrently deleted. Emulator client tried to modify an annotation, concurrently device client tried to delete an annotation. First delete operation was reached to the server then the modify operation. Server executed first the modify operation then the delete operation. We also tried to modify the same semantic base type from both the emulator and the device concurrently. In this case the latest operation effect is seen on the server side.

### A. Energy Consumption

Energy efficiency is a fundamental consideration for mobile devices due to the short battery live issues. As you may recall, we have chosen XMPP as a general communication protocol. However, XMPP was mainly originated for instant messaging and mobile platform considerations were not taken into account. Due to the nature of the protocol, XMPP generates verbose XML streams, which is acceptable in general, however it shortens battery life in mobile environment by over-generating network traffic. Moreover, XMPP requires a constant TCP connection. Open network sockets are very energy consumptive [30]. Moreover, network outages or connection type changes in the underlying mobile network connection, e.g. WLAN to

GPRS, cause to repeat all network socket and XMPP session establishment process.

In order to analyze the reason of verbose XML streams, we had logged the XMPP stanzas at the XMPP server. We observed half of the traffic as functional stanzas and the other half consists of available and unavailable presence messages. High percentage of unavailable messages can be explained by the unstable characteristic of the mobile connection.

### B. Accuracy of AR Browser Location Perception

In cultural heritage scenarios, it is expected to view the POIs that represents acquired and annotated multimedia at right geographical locations. We tested the accuracy of perceived POIs' location. First, ten artifacts were acquired and their geographical location was edited so that they have different distances and directions to the current location. Afterwards, AR browser was tested outdoors to observe the perception of where POIs are appeared. We conducted the test in outdoor intentionally in order to retrieve more accurate current location information by making use of GPS signal. We observed that the location perception of POIs were accurate enough for long distance POIs however, it was not accurate for the nearby POIs.

We can explain the results by examining how AR browser calculates the location of POIs. AR browser highly relies on the GPS to get current location, compass and accelerometer to indicate pitch and roll of the device. However, these sensors doesn't provide 100% percent accurate data. Especially, compass can generate incorrect data due to the presence of nearby metal objects.

### C. User Experience Evaluation

This section mainly focuses and evaluates the prototype based on the user point of view. The role of the prototype was expressed as a technical tool for digitally documenting the historical sites with semantic annotations and increasing the cultural heritage awareness. Since, it is significant to get actual feedback from users that tried achieve these goals, we conducted a evaluation session with seven participants.

In the scope of the evaluation session, we came across with some challenges. First, conducting the evaluation session at a real historical site would be ideal, however most of the participants do not have UMTS/3G connection as seen at Table I. Therefore, we tried to simulate the historical site by mapping historical artifacts at Aachen area as shown at figure 8. Another challenge is the number of participants. The number of evaluators is limited to seven, since we have restricted number of mobile devices. It would be better to have more users to demonstrate scalability of the system, on the other hand but seven participants is enough to visualize the concepts at the cultural heritage scenario.

**Evaluation Process.** Similar to the motivating scenario, in the evaluation session participants were asked to digitally document the historical site together and explore it with AR browser. Instruction guidelines were handed out to the evaluators before the start of the evaluation session. Once all evaluators logged in to the application, they received a short

briefing about the basic features of the application. After that, the participants were requested to the perform the following tasks.

- **Multimedia Acquisition** Users were asked to capture corresponding image and upload it to the system.
- **Annotating the Multimedia** After acquisition, the users were asked to annotate the multimedia together with the help of these documents. They were also expected to make use of chat conference feature in order to organize annotation process.
- **Augmented Reality** The evaluators were divided into 2 groups and they situated themselves at different locations. They were told to imagine their surroundings as a historical site and acquire interesting things and annotate them together as the previous parts. Furthermore, the evaluators used AR Camera and Map views to browse multimedia. They were also asked to compare geographical location consistency of the multimedia at Map and Camera View.

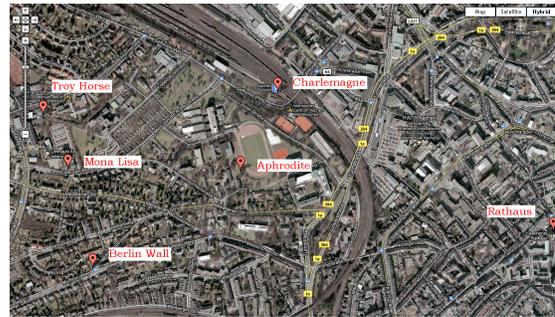


Fig. 8. Map shows location mapping of historical artifacts at evaluation session

At the end of the evaluation session participants were requested to fill a questionnaire in order to get a feedback.

### D. Results and Discussion

Participants at the evaluation session had different devices with different Android OS versions.

Device Type	Model	Android Version	Network	Processor	Screen size
Tablet	Samsung Galaxy Tab 10.1	3.1	WLAN	1 GHz Dual Core	10.1"
Smartphone	Motorola Defy	2.2	UMTS/3G	800 MHz	3.7"
Smartphone	HTC Desire	2.2	UMTS/3G	1 GHz	3.7"
Smartphone	Samsung Galaxy S2	2.3.3	WLAN	1,2 GHz Dual Core	4,3"
Smartphone	Motorola Milestone	2.2	WLAN	550 MHz	3,7"
Smartphone	Samsung Galaxy S	2.3.4	WLAN	1 GHz	4.0"
Smartphone	HTC Desire HD	2.3.3	UMTS/3G	1 GHz	4,3"

TABLE I  
MOBILE DEVICES USED AT EVALUATION SESSION

We can demonstrate the fact that even the participants who had previous knowledge about the mobile multimedia annotation can still learn new things and increase their cultural heritage awareness. This goal was achieved as seen at Figure 9.

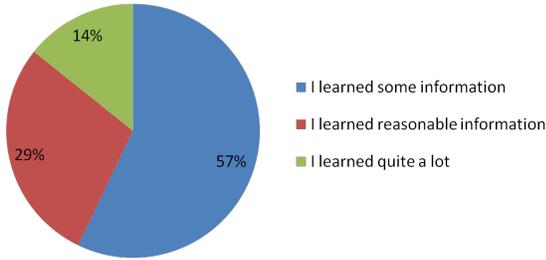


Fig. 9. How much participants increase their cultural heritage during the session

Evaluators were mostly satisfied with the user interface. They also found it responsive and user friendly as seen at Table II. They sometimes used chat functionality to get better organized for annotation and all evaluators find chat functionality useful. During the collaborative annotation the majority of the participants (5) perceived the real-time updates on the annotations however some of them (2) couldn't. This can be understandable because in order to see real-time updates, at least one other user should also annotate the same multimedia at the same time.

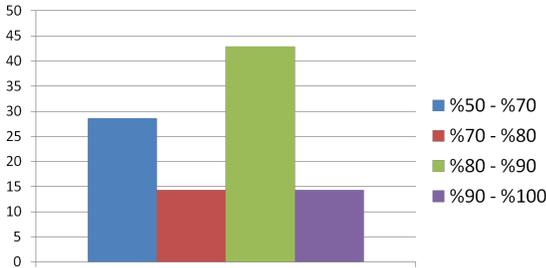


Fig. 10. How consistent perceived geographical locations of POIs at camera AR View to the actual locations at map view

The outdoor AR experience is significant to evaluate accuracy of the AR browsing at camera and map View. As expected Map View shows the locations at right positions. However, the questionnaire feedbacks pointed out that the the AR camera browser was not so successful to provide right geographical location perception as seen Figure 10. This result also supports technical evaluation test. During outdoor experience, majority of the evaluators (5) perceived real-time updates on the point of interests.

XmmC is also capable of integrating with LAS MPEG-7 multimedia services. We also compared the basic metadata information and semantic annotations on the XmmC client and MPEG-7 services. The results showed that, XmmC propagated all annotation requests to LAS MPEG-7 services successfully.

	Absolutely No	No	Neutral	Yes	Absolutely Yes
Do you find the user interface of XmmC is user friendly?	0	0	0	5	2
Do you think XmmC is a responsive application?	0	0	0	5	2
Do you think the feature to change the geographical location and accuracy of the acquired media is useful?	0	0	1	4	2
Do you think chat conference feature is useful for collaboration?	0	0	0	1	6
Have you received real-time updates on the annotations?	0	0	1	2	4
Do you find suggesting Street, City and Country names from map annotation useful?	0	0	0	2	5
Filtering the point of interests by changing the radius is useful?	0	1	1	2	5
Have you noticed real-time updates on the point of interests?	0	1	1	1	4
Do you think the prototype in general is useful?	0	0	0	2	5
Do you find the prototype in general successful?	0	0	0	4	3

TABLE II  
MULTIPLE CHOICE QUESTIONS RESPONSES

In the context of the evaluation session, 7 participants acquired 13 multimedia artifacts and they created and modified 47 semantic annotations collaboratively. All of the evaluators found the application successful and useful in general.

#### E. Evaluation Summary

We evaluated the implemented system based on technical and user point of view. In the technical part, we tested the prototype for storing and transferring for various media types. Then, the feasibility of XMPP-based CEFX+ framework in a mobile collaborative annotation scenario was evaluated for real-time performance and consistency maintenance. We also compared battery consumption while performing various tasks and examined XMPP protocol for energy efficiency. Technical evaluation finalized with examining the AR browser's accuracy to show the POIs at the right position. In the user experience based evaluation part, we conducted an evaluation session to examine XmmC while evaluators used the prototype for collaboratively documenting a historical site and increasing their cultural heritage scenario. After the evaluation we compared the acquired multimedia artifacts at XmmC and LAS MPEG-7 multimedia services.

In the light of these results, we can conjecture that XmmC is serving its purpose. However, a larger scale evaluation on more historical sites by professionals will allow us to improve the prototype.

#### VI. CONCLUSIONS AND FUTURE WORK

There are a lot of collaborative tools available on the Web to support professional communities with both their work practices and a lot of new emerging practices [31]. Such trends have been integrated in this paper too. For example, the fusion of Web and mobile technologies allow professional communities also to transform their collaborative work practices in the field into Web-based practices following the already happened transformations. Since a lot of scenarios for mobile

collaboration demand real-time functionality, more and more professional communities adopt such services, e.g. for disaster management or excavation management. Cultural heritage management is a practice we accompanied since many years [32]. Here, we presented a collaborative semantic-enhanced multimedia annotation system. We developed services like collaborative annotation and editing services and integrated them with existing mobile augmented-reality browsers and existing multimedia services based on the MPEG-7 standard. Real-time support was realized by deploying the Extensible Messaging and Presence Protocol (XMPP). All these services have been used in a cultural heritage management scenario with exposed many of the addressed requirements. Evaluation indicates that such solutions increase the awareness of community members for activities of co-workers and the productivity in the field in general.

This is a research prototype developed in the context of the German excellence cluster UMIC (Ultra Mobile High-Speed Information and Communication) funded by the national science foundation of Germany. In the cluster, future scenarios for high-demanding bandwidth usage are created to challenge current constraints on mobile phones like battery capacities, screen sizes, input handling and so on. Therefore, the evaluation was considering many of these constraints also. In future work, we will extend the scenarios in two directions. First, we will deploy more demanding mobile multimedia services like video editing. Therefore, we offload most of these services dynamically to cloud services because of the already mentioned constraints on mobile devices. Second, we enlarge the user communities and include trust and security management. This will allow the cooperation of different communities with different level of security requirements to collaborate with their mobiles.

## REFERENCES

- [1] M. Lechner, "ARML - Augmented Reality Markup Language," [Online], Mobilizy GmbH., Tech. Rep., October 2010, [http://www.percey.com/ARStandards/Mobilizy\\_ARML.pdf](http://www.percey.com/ARStandards/Mobilizy_ARML.pdf).
- [2] H. Kosch, *Distributed Multimedia Database Technologies Supported by MPEG-7 and MPEG-21*. Boca Raton et al.: CRC Press, 2003.
- [3] C. A. Ellis, S. J. Gibbs, and G. Rein, "Groupware: Some Issues and Experiences," *Communications of ACM*, vol. 34, pp. 39–58, January 1991.
- [4] J. Grudin, "CSCW: History and Focus," *IEEE Computer*, vol. 27, no. 5, pp. 19–26, 1994.
- [5] D. Lugiez and S. Martin, "Peer to Peer Optimistic Collaborative Editing on XML-like Trees," 2009, <http://arxiv.org/abs/0901.4201>.
- [6] A. R. S. Gerlicher, "Developing Collaborative XML Editing Systems," Ph.D. dissertation, University of the Arts London, October 2007.
- [7] W. Bailer, L. Brunie, M. Döllner, M. Granitzer, R. Klamma, H. Kosch, M. Lux, and M. Spaniol, *Multimedia Metadata Standards*. Springer, Berlin, Germany, 2008.
- [8] C. A. Ellis and S. J. Gibbs, "Concurrency Control in Groupware Systems," in *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data*. ACM Press, 1989, pp. 399–407.
- [9] D. A. Nichols, P. Curtis, M. Dixon, and J. Lamping, "High-latency, Low-bandwidth Windowing in the Jupiter Collaboration System," in *Proceedings of the 8th annual ACM symposium on User Interface and Software Technology*, ser. UIST '95. ACM, 1995, pp. 111–120.
- [10] S. Xia, D. Sun, C. Sun, D. Chen, and H. Shen, "Leveraging Single-user Applications for Multi-user Collaboration: The Coword Approach," in *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW'04)*. ACM Press, 2004, pp. 162–171.
- [11] M. Suleiman, M. Cart, and J. Ferrié, "Serialization of Concurrent Operations in a Distributed Collaborative Environment," in *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (GROUP'97)*. ACM, 1997, pp. 435–445.
- [12] C. Sun, X. Jia, Y. Zhang, Y. Yang, and D. Chen, "Achieving Convergence, Causality Preservation, and Intention Preservation in Real-time Cooperative Editing Systems," *ACM Transactions on Computer-Human Interaction*, vol. 5, pp. 63–108, March 1998.
- [13] A. H. Davis, C. Sun, and J. Lu, "Generalizing Operational Transformation to the Standard General Markup Language," in *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW'02)*. ACM Press, 2002, pp. 58–67.
- [14] C.-L. Ignat and M. C. Norrie, "Customizable Collaborative Editor Relying on TreeOPT Algorithm," in *Proceedings of the Eighth Conference on Computer Supported Cooperative Work*. Norwell, MA, USA: Kluwer Academic Publishers, 2003, pp. 315–334.
- [15] D. Sun and C. Sun, "Operation Context and Context-based Operational Transformation," in *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work (CSCW'06)*. ACM Press, 2006, pp. 279–288.
- [16] D. Wang, A. Mah, and S. Lassen, "Google Wave Operational Transformation," Whitepaper, Google, Inc, Tech. Rep., July 2010, <http://www.waveprotocol.org/whitepapers/operational-transform>.
- [17] Novell, Inc, "Novell vibe cloud," [Online] last accessed at April, 2011, <https://vibe.novell.com/>.
- [18] M. Voigt, "Erweiterung und Anpassung des Collaborative Editing Framework for XML (CEFX)," Master's thesis, University of Applied Sciences Erfurt, 2009.
- [19] D. Schuster, T. Springer, and A. Schill, "Service-based Development of Mobile Real-time Collaboration Applications for Social Networks," in *IEEE PerCom Workshops held in conjunction with the IEEE Pervasive Computing and Communications Conference (PerCom 2010)*, 2010, pp. 232–237.
- [20] Stanford Mobile and Social Computing Research Group, "Junction Documentation for Application Developers," [Online] last accessed on April, 2011, <http://mobisocial.stanford.edu/index.php?page=junction>.
- [21] T. H. Höllerer and S. K. Feiner, *Telegeoinformatics: Location-Based Computing and Services*. Taylor and Francis Books Ltd., 2004, ch. Mobile Augmented Reality.
- [22] X. Wang, "Layar Augmented Reality Browser," [online] last accessed April, 2011, <http://layar.pbworks.com/>.
- [23] Y. You, P. Belimpasakis, and P. Selonen, "A Hybrid Content Delivery Approach for a Mixed Reality Web Service Platform," in *Ubiquitous Intelligence and Computing*, ser. LNCS, Z. Yu, R. Liscano, G. Chen, D. Zhang, and X. Zhou, Eds. Springer, 2010, vol. 6406, pp. 563–576.
- [24] M. Spaniol, R. Klamma, H. JanSSen, and D. Renzel, "Las: A lightweight application server for mpeg-7 services in community engines," in *Proceedings of 6th International Conference on Knowledge Management, I-KNOW '06*. Springer-Verlag, 2006, pp. 592–599.
- [25] Ignite Realtime, "Smack XMPP Library," [Online] last accessed at April, 2011, <http://www.igniterealtime.org/projects/smack/>.
- [26] —, "Openfire XMPP Server," [Online] last accessed at April, 2011, <http://www.igniterealtime.org/projects/openfire/>.
- [27] Mixare, "Mixare Augmented Reality Engine," [Online] last accessed April, 2011, <http://www.mixare.org/>.
- [28] T. Muldowney, M. Miller, and R. Eatmon, "XEP-0096: SI File Transfer," XEP-0096(Standards Track), xmpp.org, April 2004. [Online]. Available: <http://xmpp.org/extensions/xep-0096.html>
- [29] P. Saint-Andre, "XEP-0045: Multi-User Chat," XMPP XEP-0045 (Standards Track), xmpp.org, July 2008. [Online]. Available: <http://xmpp.org/extensions/xep-0045.htm>
- [30] I. Koren, "Conceptual Design of a mobile collaborative Platform based on Android and XMPP," Master's thesis, Technische Universität Dresden, 2008.
- [31] R. Klamma, M. Jarke, E. Hollender, and D. Börner-Klein, "Enabling communities by constructed media: The case of a web-based study environment for a talmudic tractate," in *Advances in Web-Based Learning, Proceedings of the First International Conference, ICWL 2002, Hong Kong*, ser. LNCS, vol. 2436. Springer, 2002, pp. 275 – 285.
- [32] R. Klamma, M. Spaniol, M. Jarke, Y. Cao, M. Jansen, and G. Toubekis, "ACIS: Intergenerational community learning supported by a hypermedia sites and monuments database," in *Proceedings of the 5th International Conference on Advanced Learning Technologies (ICALT 2005), Kaohsiung, Taiwan*. IEEE Computer Society, 2005, pp. 108–112.