A Collaborative Environment for E-training in Archaeology

Manuella Kadar and Maria Muntean

"1 Decembrie 1918" University of Alba Iulia, Romania
{mkadar,mmuntean}@uab.ro

Abstract. This paper describes CREST (Collaborative Environment for Students and Teachers) a novel integrated environment for collaborative content retrieve and annotation and e-training in the field of Archaeology that is used by teaching stuff and students, as well. CREST is used in a broad range of collaborative applications and enables multi-authoring, using information in educational interactions by indicating information source, maintaining information, structuring information, adding meta-information, and sharing information among participants. An ontology enabled annotation and knowledge management environment was developed and endowed with collaborative information searching agents. Two agents were implemented in order to search and download aggregated metadata and text documents from the Web and to store information into a documents corpus repository. The corpus repository is further accessed by users through the annotator interface. This original approach is based on open standards and integrates open source services that improve discovery and reasoning across domain specific collections. CREST allows users to create, edit, store, and retrieve objects and annotations, promotes development and re-use of meaningful content. Such environment has a great utility for the development of virtual learning and research spaces.

Keywords: Domain knowledge representation, social tagging, information retrieval, 2D/3D digital objects annotation, ontology.

1 Introduction

The wide adoption of technologies that enable users to connect to each other and to contribute to the online community has changed the way that content is organized and shared on the Web. Social tagging to annotate resources represents one of the innovative aspects introduced with Web 2.0 and the new challenges of the semantic Web 3.0. In many online applications, it is possible for users to upload their own content or links to existing content and to organize it by use of tags, i.e., free-form keywords. Such applications, examples of which are *delicious*, *flickr*, and *BibSonomy*, are commonly referred to as collaborative tagging systems and they use the Internet to harness collective intelligence.

This paper presents the design and implementation of a **C**ollabo**R**ative **E**nvironment for **S**tudents and **T**eachers, named **CREST**. CREST allows a broad range of

G. Jezic et al. (eds.), *Agent and Multi-Agent Systems: Technologies and Applications*,
Advances in Intelligent Systems and Computing 296,
DOI: 10.1007/978-3-319-07650-8_30,

collaborative applications and enables multi-authoring by: using information in educational interactions, indicating information source, maintaining information, structuring information, adding meta-information, and sharing information among participants. The participants are required to contribute through annotations that may include features such as comments on multimedia objects: text, 2D/3D objects, audio, video, virtual graphical spaces.

Current ways for Web indexing are not sufficient for learning resources. Indeed, automatic indexing, e.g. Google, can hardly rise above the syntax level of contents while indexing by human experts implies high costs. Recent approaches like semantic web and participative web (Web 2.0) offer promising solutions. This approach is based on semantic web technologies and ontology development used for building educational hypermedia systems. Another challenge addressed in this paper is about the study of functionalities on participative websites and the adding of content and metadata by visitors. In this respect, this paper presents a model of participative web interface adapted to communities of students and teachers.

Formal domain ontologies generally produced by experts are opposed to heterogeneous tags added by numerous users with various profiles. The presented model takes advantage of both semantic and participative approaches by populating formal domain ontologies with automatically extracted information from annotated multimedia objects. The goal of this model is to help the development of applications for sharing resources into communities of practice. It is based on a progressive indexing in which users progressively structure metadata, to finally allow semantic reasoning by computers and a shared vision of the domain by humans. This model integrates through the annotator interface several tools such as: social bookmarking tool, SemanticScuttle [1], that offers original features like tags structured by relations of inclusion and synonymy, or wiki spaces to describe tags. Another integrated facility is the tagging system of 2D objects achieved through Annotation Pilot [2] and a 3D ontology-enabled semantic annotator, ShapeAnnotator [3]. The environment was developed and tested with students and teachers in the field of Archaeology.

The Collaborative Environment for Students and Teachers – CREST aims to identify, implement and evaluate semantic approaches and enable academic institutions to exploit the full potential of community annotation/tagging systems. CREST was developed in order to enable annotation and knowledge management environment and to provide semantic web services. Personalized annotation is used to equip the collaborators with Web based authoring tools for commenting, knowledge articulation and exertion. The environment has enhanced conventional annotation system by extracting metadata from both the annotated content and the annotation itself, and establishing ontological relation between them. The aim was to develop an efficient environment based on open standards and comprising a set of open source services that improves discovery and reasoning services across domain specific collections by meeting the following main objectives:

- to identify a common model for representing tags and annotations on text, 2D and 3D digital objects, virtual graphical spaces and to enhance the interoperability of tags/annotations from distributed sources e.g., different communities using different systems;

- to develop a set of easily deployable tools and services for attaching annotations to multimedia including text, 2D/3D objects, virtual graphical spaces for harvesting annotations, and aggregating distributed annotations with metadata;
- to search and download aggregated metadata and documents from the web with information searching agents and to collect searched results into a corpus repository.

The remainder of this paper is organized as follows: section 2 discusses previous related work and the technical issues that influenced system design and implementation, section 3 describes the common, extensible model employed for representing annotations/tags and CREST architecture, section 4 provides examples of operation within CREST, section 5 presents results and discussions, and finally future work plans and conclusions are pointed out in sections 6.

2 Background and Related Work

2.1 Web-Based Annotation Services for Documents

Annotation of documents can be achieved by specialists according to a model produced by experts such as Learning Object Metadata [4]. In such cases, experts create ontologies, which are models of domains on which computers can perform reasoning. According to [5] and [6], this approach includes at least two weaknesses. Firstly, the creation of a common model is a difficult task — even for experts — requiring an important negotiation phase. Secondly, the annotation process is costly because it must be operated by specialists who understand and are able to apply the pre-defined model. Consequently, this centralized approach can be difficult to manage as difficulties encountered in the application of LOM. Nevertheless, tagging reaches its limits in the fact that it does not allow advanced structuring. First of all, tags leads to flat structures, letting few ways for users to organize their own tags. Moreover, tags' efficiency decreases because of problems of typography or synonymy. On the other hand, the rigidity of approaches based on controlled vocabularies (thesaurus, ontologies) realized by experts seems incompatible with the flexibility of tags built in a very distributive way. One of the big efforts to integrate annotations with ontologies is the Semantic MediaWiki (SMW) open-source project to which many people and organisations have contributed. Semantic MediaWiki (SMW) is an extension of MediaWiki – the wiki application best known for powering Wikipedia – that helps to search, organise, tag, browse, evaluate, and share the wiki's content. While traditional wikis contain only text which computers can neither understand nor evaluate, SMW adds semantic annotations that allow a wiki to function as a collaborative database. Semantic MediaWiki introduces some additional markup into the wiki-text which allows users to add "semantic annotations" to the wiki [7].

2.2 Web-Based Annotation Services for 2D/3D Objects

Learning systems should be provided with sufficient visualization of multidisciplinary contents for educational activities, especially by 3D models, 3D animations and simulations, which facilitate the learner's immersion in a hidden world [8]. The coding of additional knowledge in the form of structured metadata is important for the development of learning environments that take into account not only the geometry of shapes but also their semantics or meaning. At the same time, structural decompositions allow to consider a shape not only as a whole, but also as the collection of its parts. An approach to structural decomposition is augmented reality that can be used as system for annotating real-world objects. In this approach a virtual form (model) of a real-world object is matched to the real object, allowing one to visually annotate the real components with information from the corresponding model. Augmented reality provides a natural method for presenting the "enhancing" computer-based information by merging graphics with a view of the real object. User queries on the real object can be translated into queries on the model, producing feedback that can augment the user's view of the real world [9].

At present, several research projects are focused on the issue of the content-based information attached to 3D data generated by 3D scanning (achieved by scanner devices) photogrammetry (reconstruction of 3D data from 2D images) and procedural/parametric shape design (creation of new shapes from existing similar, parameterized shapes). Due to the nature of the data type and complexities involved in acquisition, production and processing of 3D data, a number of serious issues exist regarding 3D data acquisition, representation, encoding, content mark-up, and data history management. To date, these problems have not been sufficiently solved, and represent a major obstacle to a full integration of the 3D data type into digital archives. Among the 3D annotation tools that have been developed so far, are to be mentioned: SpacePen that allows a team to collaboratively work on a building design by annotating Java3D models using a web browser and a pen-based interface for drawing suggested modifications on the model [10], AnnoCryst [11] that enables users to annotate 3D crystallography models through a JMOL viewer and store the annotations on a shared web server and 3DSEAM designed to enable 3D scenes represented using the Extensible 3D (X3D) standard to be annotated using MPEG-7 [12]. All of these systems enable users to attach annotations to 3D models and to browse annotations added by others, asynchronously.

3 Architectural Model

CREST stores the annotations on a server that is separate to the server hosting the multimedia data. It has been assumed that there may be multiple communities, who may be annotating the same set of objects.

The Web annotation system comprises an annotation creation/authoring and attachment interface, an annotation web browser, information searching agents with retrieval interface, annotation storage and indexing component that stores the information into a documents corpus repository. CREST architecture is presented in Fig. 1.

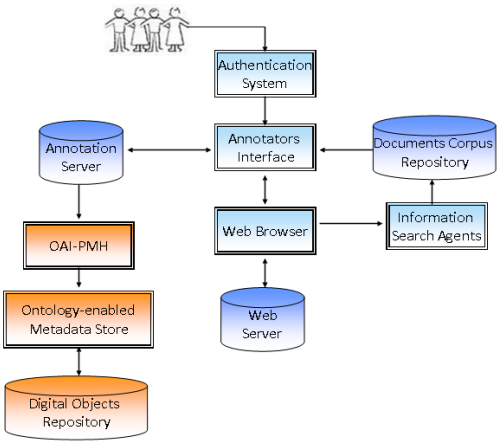

Fig. 1. CREST architecture

In the proposed model, decoupling of the annotations from the content allows more control and flexibility over how the annotations are accessed, processed, presented and re-used. The annotations are stored on a separate server that facilitates easy access, authoring and posting of responses that are controlled and restricted to a particular community of users, in this case, students and teachers from the Archaeology specialization. The separation also allows a single resource to be annotated in many different ways by users on the same annotation server or on different annotation servers, by using different community-specific terminologies or ontologies. By separating the annotations from the resources, the copyright issues that arise when having to store a copy of the digital resources on the social tagging site are also avoided.

The importance of being able to aggregate metadata from a range of sources has been recognized by a number of projects. Annotea [13] has recognized the need for standardized ways of defining annotations and tags so they can be shared between communities. However, the problem of annotation aggregation is still largely unresolved.

CREST proposes retrieving stored annotation data through the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). This approach involves mapping the annotations stored on Annotea server to the collections' metadata schema and periodically harvesting the annotations/tags by having the central agency send OAI-PMH (HTTP) requests to the server (Fig.1).

CREST adopts the "ontology-enabled folksonomy" approach in which users are provided with suggested Dublin Core metatags [14] and popular tags of an ontology (specified at system configuration). Users have also the option to define their own unique tags (Fig. 3). When an information searching agent browses on a parent tag, all items with the parent tag, synonym tags or children tags are retrieved. The class hierarchies are also incorporated within the tag cloud to embed multi-level structuring. In addition, CREST restricts access to the annotation server through an identity management system. This novel approach provides annotation services for closed communities with specific knowledge or expertise – students and teachers in a specific field that reduces the proportion of incorrect, inappropriate or misleading tags.

Within CREST community annotations are stored on an annotation server that is separate from the data collections or the web sources that the users annotate. An OAI-PMH interface has been built on top of the annotation server. This enables the periodic harvesting of new annotations (since the last harvest) by sending OAI-PMH (HTTP) requests to the server. The harvested annotations are then aggregated with the institutional metadata, to enrich the metadata store with community knowledge [15].

The information searching agents download documents from the Internet by using a list of words as thesaurus. The downloaded files are placed in the documents corpus repository and are processed in order to extract information by using specific annotation tools.

The information searching agents consist of the following classes: *ReceiverAgent*, *ReceiverBehaviorReceivePing*, *SenderAgent*, *SenderBehaviorReceiveAnswer*, *SenderBehaviorSendPing*, *download*, *yahoo*, *MessageManager*, *RunJade* and *RunMe*. The main classes are presented in the class diagrams in Fig. 2.

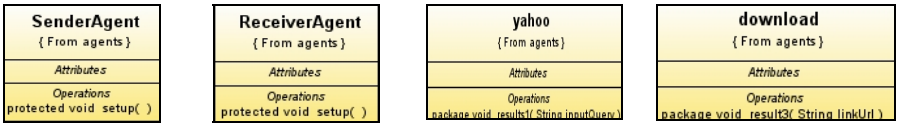

Fig. 2. Class diagrams

SenderAgent class aims to create intelligent agent "a1" which extends the *Agent* class. Method *setup()* initializes the agent, and by calling *addBehavior* method, behavior is added. Thus, agent "a1" has two behaviors: it receives messages from the agent "a2" through *SenderBehaviorReceiveAnswer* and it sends messages through *SenderBehaviorSendPing*.

```
public class SenderAgent extends Agent {
      protected void setup() {
               //Add SenderBehaviourReceiveAnswer behavior
               addBehaviour(new SenderBehaviourReceiveAnswer(this) );
               // Send messages to "a2" agent
               addBehaviour(new SenderBehaviourSendPing(this)); } }
```

ReceiverAgent class aims to create "a2"agent, which also extends the *Agent* class. The *setup()* method initializes the agent, and by calling *addBehavior* methos, it receives a behavior. Thus, "a2" receives messages from the agent "a1" and sends its reply through *ReceiverBehaviorReceivePing*.

```
public class ReceiverAgent extends Agent {
      protected void setup() {
               addBehaviour(new ReceiverBehaviourReceivePing(this)); } }
```

This behavior is used by the agent "a1" to send to the "a2" keywords for searching and downloading *.pdf* or *.html* documents from the www. *SenderBehaviorSendPing* class extends *OneShotBehavior* class, having a one-shot type behavior, i.e. sends all the keywords, and then it stops working.

SenderBehaviorReceiveAnswer class is the second "a1" agent behavior and aims receiving messages from "a2" through conversation with id "message". This behavior stops working when *done()* method is called and then it returns the value "true".

ReceiverBehaviorReceivePing class represents the "a2" agent behavior, which has the following functions that are repeated for each keyword received from the agent "a1":

1. Gets the word through conversation with id "message";
2. Calls the method *results1* from *yahoo* class sending as parameter the keyword;
3. Takes the search results from *yahooresults.html* file and extracts through regular expressions the target links to which documents to be downloaded are related;
4. Sends response to "a1" agent with the links found, one by one;
5. Sends the links found, one by one, to the *download* class by calling *results3* method that will download documents from the www to a local folder.

All this take place within the *action()* method, and when the *done()* method returns true, the agent "a2" stops its activity.

Yahoo class is called inside *ReceiverBehaviorReceivePing* class and through r*esults1()* method, it performs the following operations:

6. Add keyword received as a parameter to a URL through is perform a search with the www.yahoo.com popular search engine;
```
URL url1 = new URL("http://search.yahoo.com/search?p="+inputQuery);
```
7. Create a *yahooresults.html* document that will store bit by bit all page content resulting from the search by given keyword:
```
FileOutputStream fs = new FileOutputStream("D:\\DownloadPDF\\yahooresults.html");
```

4 Examples of operation of CREST

This section presents an example of how users annotate historical objects, based on the information previously searched by agents and automatically stored into the documents corpus repository. The information searching agent starts to perform search by keywords such as: *axe.pdf*, *socket.pdf* and *sharpaxe.pdf*. A piece of the output generated by running application is shown below:

```
   Sender: I am a1 and I am sending input queries
   Receiver: I am Agent a2 and I have received: axe.pdf from a1
   Sender: I am a1 and I received: Search in the browser performed from a2
   Sender: I am a1 and I received: I extracted the link: http://www.flickr.com/ from a2
   Sender: I am a1 and I received: I downloaded a .pdf or .html file from the address:
http://www.flickr.com/   from a2
   Sender: I am a1 and I received: I extracted the link:
http://www.grandforest.us/TheAxeBook.pdf from a2
   Sender:I am a1 and I received: I downloaded a .pdf or .html file from the address:
http://www.grandforest.us/TheAxeBook.pdf  from a2
```

After the information searching agents populate the corpus repository users may access this repository and start annotation through the annotator's interface.

The annotator's interface also enables users to access SemanticScuttle, Annotation Pilot and ShapeAnnotator that are open source software integrated into CREST collaborative learning environment. An example is given in Fig. 3. Users can create and attach annotations to resources retrieved via a web search interface.

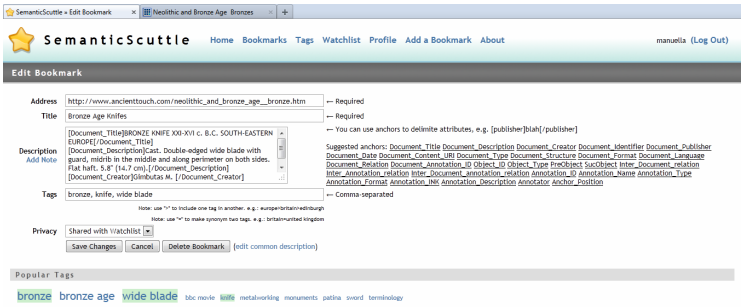

Fig. 3. Example of annotation for Bronze Age Knifes

Further on, a 2D object can be loaded into the Annotation Pilot, the image being captured and annotated with tags describing the most important parts of the object. This is an important tool for learning classifications of ancient artefacts (Fig. 4).

Fig. 4. Example of Annotator Pilot for images

The environment supports the annotation of 3D objects, as well as provides a user interface for browsing and searching annotations. Users can search across annotation attributes that include: creator, date, keywords or free-text searching over the description. In the ontology-driven annotation, the tags are defined by the ontology. The coupling of segmentation and knowledge formalization fosters the development of totally new approaches to shape retrieval. An example is provided for the bronze axe presented in Fig. 5. For this artefact it is possible to address queries such as: "*find a shape containing a loop and socketed body*", or more specifically to refer directly to subparts, e.g. "*find a socketed axe with rectangular mouth*", or "*rope moulding around the mouth*" obtaining as results, proper subparts of shapes. Semantics can be associated to the content itself, thus providing an enriched representation of the content.

Fig. 5. Bronze axe

The bronze axe presented in Fig.5 has an associated ontology developed in Protégé 3.4, which is an open source ontology editor and knowledge-base framework [16]. The ontology is loaded into the ShapeAnnotator tool [17]. After loading of the model and ontology, the first step of the annotation pipeline is execution of the segmentation algorithms to build the multi-segmented mesh. Once done, from the resulting multi-segmented mesh interesting features can be easily selected by simple mouse-clicks. Each interesting feature can then be annotated by creating an instance of a concept described in the ontology.

5 Results and Discussions

The result of the annotation process is a set of instances that, together with the domain ontology form a knowledge base (Fig. 6).

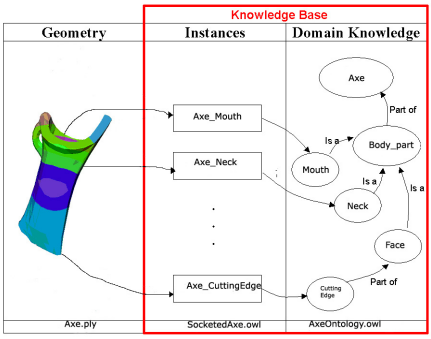

Fig. 6. Instances with specific relations representing a bridge between the geometry and semantics

All instances produced during the annotation pipeline are automatically assigned values for the above properties, so that the link between semantics and geometry is maintained within the resulting knowledge base. The ontology is stored in the *AxeOntology.owl* file (Fig.6). The knowledge base can be further exploited in educational activities. CREST was evaluated by more than thirty students and two professors in the field of Archaeology. Users have assembled data storage and retrieved structure for the archaeological field that allowed long-term data storage, (re)annotation, free-text searching, and dynamic record assembly. The system proved to be effective because students became active and interactive learners.

6 Conclusions

The presented collaborative annotation environment enables extensible and flexible storage of a specific domain data through collaborative tagging and use of specific domain ontologies. Focusing on the needs of virtual learning environments, this paper

has presented how new approaches for data representation address changes due to the inevitable growth (both in diversity and volume) of data stores. The main components of the model and implementation architecture have been discussed with examples. Further developments will focus on models for melting different indexing solutions: automatic or by humans, including experts or simple users, based on structured models (e.g. ontologies) or on flexible metadata (e.g. tags). Structurable tags prove the technical possibility to make inferences on tags while keeping their spontaneous and flexible aspect. The contribution of domain experts and students in the design of such learning materials will increase the usability of structured tags created within CREST by the community of students and teachers and will extend the test bed to other fields of interest.

References

1. SemanticScuttle, http://semanticscuttle.sourceforge.net/
2. Annotation Pilot, http://www.colorpilot.com/annotation.html
3. Aim&Shape repository, http://shapes.aimatshape.net/
4. WG 12: Learning Object Metadata, http://ltsc.ieee.org/wg12/index.html
5. Brooks, C., Bateman, S., Liu, W., McCalla, G., Greer, J., Gasevic, D., Eap, T., Richards, G., Hammouda, K., Shehata, S., Kamel, M., Karray, F., Jovanovic, J.: Issues and directions with educational metadata. In: Third Annual International Scientific Conference of the Learning Object Repository Research Network, Montreal (2006)
6. Downes, S.: Resource profiles. Journal of Interactive Media in Education 5 (2004)
7. SMW, http://semantic-mediawiki.org/wiki/Semantic_MediaWiki
8. Klett, F.: A Design Framework for Interaction in 3D Real-Time Learning Environments. In: Second IEEE International Conference on Advanced Learning Technologies (2001)
9. Aracena-Pizzaro, D., Mamani-Castro, J.: Museum Guide Through Annotations Using Augmented Reality. In: Poster Papers Proceedings of the 18th International Conference on Computer Graphics, Visualization and Computer Vision 2010, pp. 35–38 (2010)
10. Jung, T., Gross, M.D., Do, E.I.Y.: Annotating and sketching on 3D Web models. In: Proceedings of the 7th International Conference on Intelligent User Interfaces, San Francisco, California, USA, pp. 95–102 (2002)
11. Hunter, J., Henderson, M., Khan, I.: Collaborative annotation of 3D crystallographic models. J. Chem. Inf. Model. 47(5), 2475–2484 (2007)
12. Bilasco, I.M., Gensel, J., Villanova-Oliver, M., Martin, H.: 3DSEAM: a model for annotating 3D scenes using MPEG-7. In: Proceedings of the 7th IEEE International Symposium on Multimedia (ISM 2005), pp. 310–319 (2005)
13. Koivunen, M.R., Kahan, J.: Annotea: an open RDF infrastructure for shared Web annotations. In: Proceedings of the 10th International Conference on World Wide Web, Hong Kong. ACM Press (2001)
14. Dublin Core Metadata Initiative, http://dublincore.org
15. Hunter, J., Gerber, A.: Harvesting community annotations on 3D models of museum artefacts to enhance knowledge, discovery and re-use. Journal of Cultural Heritage 11, 81–90 (2010)
16. Protégé, http://protege.stanford.edu/
17. Attene, M., Robbiano, F., Spagnuolo, M., Facidieno, B.: Part-based Annotation of Virtual 3D Shapes. In: Proceedings of the International Conference of Cyberworlds, pp. 427–436 (2007)