International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.7 (2016), pp.103-118
http://dx.doi.org/10.14257/ijmue.2016.11.7.11

An Interactive Visual User Interface Model for Multimedia Open Annotation Tools

Zhuoming Xu [1] (✉), Wenping Guo [1,2], Ting Wu [1] and Lixian Ni [1]

[1] College of Computer and Information, Hohai University, Nanjing 210098, China;
[2] Institute of Image Processing & Pattern Recognition, Taizhou University, Taizhou 317000, China
E-mails: {zmxu, gwp, wuting, nilixian}@hhu.edu.cn

Abstract

The W3C Open Annotation Community Group's Open Annotation (OA) Data Model is emerging as a standardized Web Annotation Data Model. We argue that a friendly interactive visual user interface for OA Data Model based annotation tools, which is largely based on an effective user interface model, plays an essential role in the usability of the tools. In this paper we propose an Interactive Visual User Interface Model for Open Annotations (or simply IVUIM4OA), which allows the annotating user to dynamically transform OA data into a visual representation and to interactively manipulate the visual objects to modify the OA data or produce new one. IVUIM4OA is an adaptation of a generic information visualization model, the Data State Reference Model, for the specifics of the visualization of OA data; it is further extended with view operators that enable the user to interact with the visual objects. We implemented a prototype multimedia annotation tool on the basis of IVUIM4OA and conducted case study experiments with the prototype tool. The implementation and experimental results show that the proposed IVUIM4OA is feasible and works well with the OA Data Model, in such a way that the annotating user can obtain a better understanding of and more intuitive interaction with the annotation data.

***Keywords**: user interface design; information visualization model; multimedia annotation tools; Open Annotation Data Model; human-computer interaction; usage-centered design*

1. Introduction

Semantic annotation of Web resources facilitates the indexing, retrieval, access and sharing of the Web content, especially of digital multimedia content [1]. Web annotation approaches have been evolving due to the development of Web technologies. Early semantic image and video annotation tools [2] have not provided entirely satisfactory results, mainly because they usually use proprietary, incompatible annotation models and protocols, resulting in a lack of interoperability, exchangeability or shareability of the generated annotations. Web 2.0 portals, such as Flickr and YouTube, allow the community users to annotate multimedia resources by adding free-text descriptions of audio-visual content. But these portals are usually closed systems and user-generated annotations, often of lower quality or even unreliable, live within a single system and remain inaccessible to the Web of Data [3].

In contrast to multimedia content, massive structured data have been semantically interconnected, thus mutually described, by applying the Linked Data principles that are based on standard Web technologies such as URIs [4] and Resource Description Framework (RDF) [5], which forms a true Web of Data [6].

Inspired by the success of the Linked Data effort, scholars began to investigate how to apply the Linked Data principles to multimedia annotations [7]. As a result, proposals

ISSN: 1975-0080 IJMUE

represented by the Annotation Ontology (AO) [8] and the Open Annotation Collaborative (OAC) model [9] were presented to the W3C Open Annotation Community Group. The Group has reconciled the two proposals and has been working towards the standardization of the Open Annotation (OA) Data Model [10], which is the basis for the Web Annotation Data Model specification [11] under development by the newly created W3C Web Annotation Working Group (http://www.w3.org/annotation/).

Before widespread adoption of the OA data model, the community must develop easy-to-use tools for annotating users, especially for lay-users without enough semantic annotation techniques. We believe that an important step to take is the fine design of an interactive visual user interface [12] for the tools. Although various linked data visualization approaches [13] exist, they cannot be directly applied to the context of Open Annotations because these approaches just use RDF semantics, rather than the semantics for the OA data model, to interpret the linked annotation data. We argue that a friendly user interface for an annotation tool must be based upon an effective interactive visual user interface model. Unfortunately, recent seminal lines of research on OA (or OA-like) prototypes, such as [7-9], mainly focused on linked annotation data modeling and architectural issues of the annotation tools, and did not study thoroughly the most important issue—user interface modeling.

To fill the gap, in this paper we propose the *Interactive Visual User Interface Model for Open Annotations (IVUIM4OA)*, which allows the annotating user to dynamically transform Open Annotation data into a visual representation and to interactively manipulate the visual objects to produce or modify annotation data. We devise our IVUIM4OA by adapting and extending a generic information visualization model called the Data State Reference Model (DSRM) proposed by Chi [14]. IVUIM4OA has been implemented in our prototype multimedia annotation tool, OAMM, with which we successfully carried out several case-study experiments. The implementation and experimental results show that our proposed IVUIM4OA is feasible and works well with the W3C OA Data Model. Since IVUIM4OA is a general purpose, essential model for Open Annotations, it can be widely applied to the construction of any OA Data Model based annotation tools.

The rest of the paper is organized as follows. Section 2 discusses related work. In Section 3 we present our design ideas for IVUIM4OA, the overview of the proposed model as well as the data stages and transformations in the model. Our prototype implementation technologies and case study are given in Section 4. The final section is discussion and future work.

2. Related Work

With regard to *annotation data models*, Ciccarese, *et al* [3] briefly summarized the evolution of Web annotation. Annotea [15], an early W3C project, specified an open RDF infrastructure for shared Web annotations. The Annotea data model describes annotations with a dedicated RDF schema and views annotations as RDF statements or remarks made by the author or a third party about a Web document. Annotea laid out a basic path for subsequent annotation systems/tools, though it faced several challenges to widespread adoption [3]. Inspired by Annotea and especially by recent success of Linked Data, several Linked Data annotation projects started recently. The most representative are the Annotation Ontology (AO) [8] and Open Annotation Collaborative (OAC) model [9]. Both models adopted Annotea's fundamental framework, but they drew special attention to different functionalities according to their user communities' needs and use cases. The W3C Open Annotation (OA) Data Model [10] incorporates features from its predecessors, but focuses on interoperability for annotations and maximizing the benefit of annotations with unrestricted access. Based on common Web standards including URIs and RDF, the OA data model provides an extensible, interoperable framework for expressing annotations such that they can easily be shared across platforms/systems. In terms of OA, an *Annotation* is considered to be a set of connected accessible resources, typically including one *Body* or multiple Bodies and at least one *Target*, and conveys that the Body resource(s) is somehow related to the Target resource(s). Annotations, Bodies and Targets are identified by URIs/URNs and further described using ontology terms from the OA ontology and other common ontologies such as Dublin Core. Such linked data makes Web annotations a first-class object [3].

As for *user interfaces for annotation tools*, most of early annotation tools [2] provided the user with a graphical user interface. However, these tools did not use an open annotation model. Annotea's implementation [15] used a full-featured Web browser and editor, Amaya, as the client, where the user can create and browse annotations in popup windows. But the interface is intended for use in annotating HTML or XML-based documents based on the RDF data model. The Annotation Ontology (AO) [8] was designed to provide a shareable structure for semantic integration of biomedical domain ontologies with scientific documents. But there is no dedicated tool for the AO method at present. Recent OA (or OA-like) prototypes' focus was on linked annotation data modeling and architectural issues of the annotation tools. For instance, both Synote [7,16] and YUMA [9] annotation frameworks focused on how to used the OAC model to integrate multimedia annotations with the Linked Data principles. So far, these prototypes have not dealt with the most important issue—interactive visual user interface modeling, which we believe is the key to constructing easy-to-use annotation tools.

Regarding *information visualization models*, Chi [14] proposed a generic model called the *Data State Reference Model (DSRM)* and employed this model to categorize and analyze 36 visualization techniques. Recently, this generic model has been specially adapted for use as a Linked Data Visualization Model (LDVM) by Brunetti, *et al* [17]. However, the LDVM is neither intended for use in visualizing Open Annotation data, nor is it capable of supporting interactive annotating tasks. Since the DSRM model describes information visualization process in a generic way, in this paper we use DSRM as a conceptual framework and a starting point for our interactive visual user interface model.

Concerning *user interface design*, modeling user interfaces is a well-established discipline in its own right. As one of the most praised approaches to user interface engineering, *usage-centered design*, a model-driven approach first introduced in [18], has been used successfully on a variety of software projects. It is a systematic, model-driven approach to visual and interaction design for user interfaces in Web-based applications. The principles of usage-centered design have been applied to the design of our interactive visual user interface model.

3. Design Ideas and the Proposed Model

In this section we present the proposed *Interactive Visual User Interface Model for Open Annotations (IVUIM4OA)*. After briefly explicating IVUIM4OA's design ideas, we first give an overview of the proposed model and then explain its key elements.

3.1. Design Ideas

The fundamental goals of an interactive visual user interface for an Open Annotation (OA) tool are to allow the annotating user to get a good understanding of the OA data being produced and thus obtain intuitive interaction with the data during the task of annotation. Such a user interface is largely based upon an effective user interface model. Conceptually, a user interface model for an annotation tool is a representation of how the annotating user interacts with the tool software to produce or modify annotation data and how the software responds. So we believe that the user interface model must meet two basic requirements. First, it is a *visualization model* that is used to dynamically transform OA data in an RDF serialization format (such as Turtle [19]) into a visual representation (view). Second, it is an *interaction model* that enables the user to intuitively interact with the visual objects and then produce or modify OA data. Therefore, we devise IVUIM4OA by both adapting and extending the generic DSRM model [14]. More specifically, IVUIM4OA is an adaptation of DSRM for the specifics of the visualization of OA data, and is further extended with view operators (*i.e.*, event-triggered operations) that can trigger *interaction contexts* within which the user performs actions and the system responds accordingly, resulting in data generation/updating and then refreshing the visual representation. Furthermore, following the principles of usage-centered design [18], we focus on annotating users' intentions and usage patterns and employ abstract (essential) use cases—called *task cases*—to structure the work annotating users need to accomplish with the system and to represent the user interface's contents and how these are organized into users' interaction contexts, from which we derive IVUIM4OA's visual and interaction design.

3.2. Overview of the Model

Figure 1 depicts the overview of our IVUIM4OA. The names of the data stages, transformations and operators proposed by DSRM have been slightly adapted to the context of Open Annotations. Furthermore, the View In-Stage Operators have been extended with "event triggering" operators which can trigger performing well-organized *task cases-based user actions* (straightforward commands or interactive operations) in the *interaction contexts* of the Data Updating Transformation, where the user can produce new annotation data or modify the existing one by means of, for example, form-based operations in popup windows. Unlike DSRM, IVUIM4OA now resembles a *loopy* data pipeline in which Open Annotation data are transformed stage by stage into a visual representation of the raw data, and new/modified annotation data can be produced through the Data Updating Transformation.

3.3. Data Stages and Transformations in the Model

IVUIM4OA is broken into four distinct data stages that raw data needs to pass through:

- *Open Annotation (OA) Data:* Raw data that is an RDF dataset in a serialization format such as RDF/XML, N3, Turtle, N-Triples and JSNON-LD.
- *Analytical RDF Abstraction:* A set of RDF triples, each consisting of a subject, a predicate and an object. An RDF triple states that some relationship, indicated by the predicate resource, holds between the resources denoted by the subject and object. These resources are normally identified by unique HTTP URIs (the object can also be a literal).
- *Visualization OA Abstraction:* OA graphic objects data that is visualizable on the screen using a particular visualization technique. The semantics and graphic styles have been specified by the W3C Open Annotation Data Model specification [10].
- *View:* On-screen visual representation presented to the annotating user, where the user sees and interprets the representation and interacts with the visual objects.

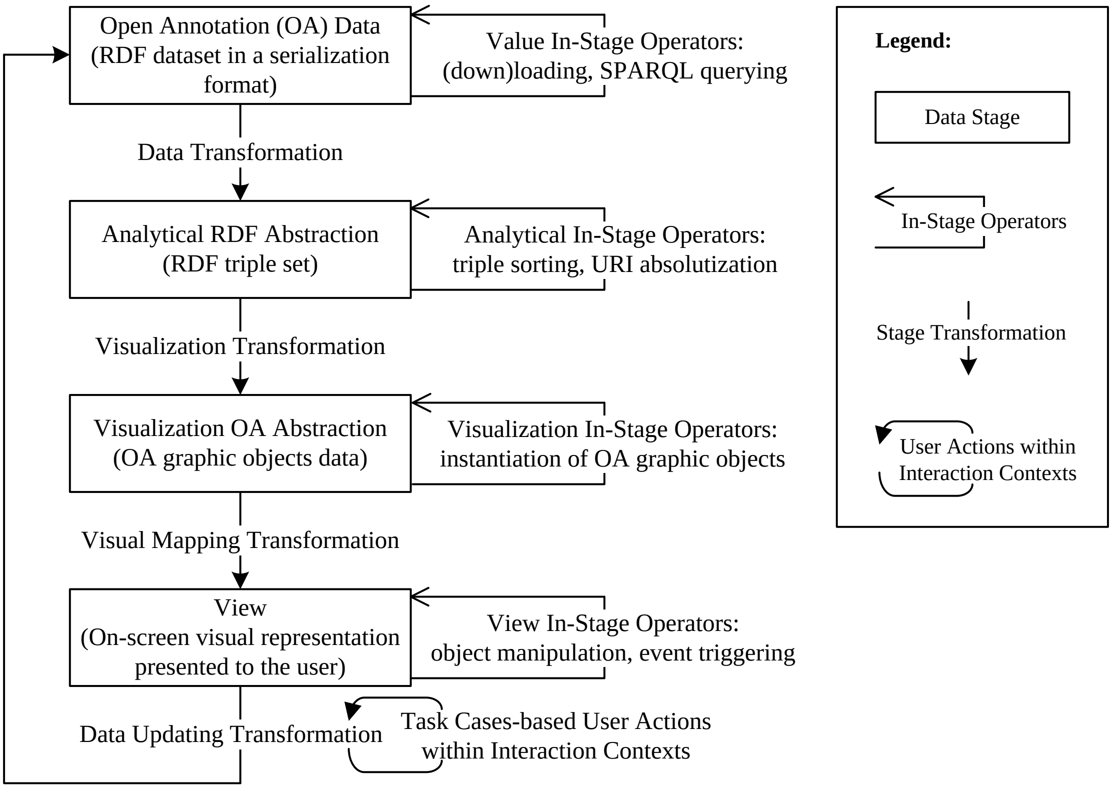

Open Annotation (OA) Data
(RDF dataset in a serialization format)
Value In-Stage Operators:
(down)loading, SPARQL querying
Data Transformation
Analytical RDF Abstraction
(RDF triple set)
Analytical In-Stage Operators:
triple sorting, URI absolutization
Visualization Transformation
Visualization OA Abstraction
(OA graphic objects data)
Visualization In-Stage Operators:
instantiation of OA graphic objects
Visual Mapping Transformation
View
(On-screen visual representation presented to the user)
View In-Stage Operators:
object manipulation, event triggering
Data Updating Transformation
Task Cases-based User Actions within Interaction Contexts
Legend:
Data Stage
In-Stage Operators
Stage Transformation
User Actions within Interaction Contexts

Figure 1. Overview of the Proposed Model (IVUIM4OA)

Data is propagated cyclically through the IVUIM4OA pipeline from one stage to another by applying four types of transformations:

- *Data Transformation:* Transforming the OA Data (*i.e.*, an RDF dataset in a serialization format) into the Analytical RDF Abstraction (*i.e.*, RDF triple set) by applying RDF parsing.
- *Visualization Transformation:* Transforming the Analytical RDF Abstraction into the Visualization OA Abstraction (*i.e.*, OA graphic objects data) by semantically interpreting the set of RDF triples, identifying OA elements and then determining the corresponding graphic styles.
- *Visual Mapping Transformation:* Mapping the Visualization OA Abstraction to a concrete View (*i.e.*, on-screen visual representation) by using a particular visualization technique.
- *Data Updating Transformation:* It is triggered by user-interactions (*e.g.*, mouse clicks on the view's objects) and enables the user to produce annotation data or to modify the existing one by performing user actions. The updating of the data will result in refreshing the View.

There are operators within the stages that allow for in-stage data transformations:

- *Value In-Stage Operators:* Loading the raw data into the tool from local storage, or downloading the raw data from the Web, or sometimes obtaining the data via a SPARQL query against the data source.
- *Analytical In-Stage Operators:* Sorting the RDF triples and converting all relative URLs into absolute URIs by merging the relative URLs with the corresponding base URIs.
- *Visualization In-Stage Operators:* Creating data instances of various classes for OA graphic objects.

- *View In-Stage Operators:* Two types of operators: (i) mouse manipulations of the visual graph or its objects (scaling, zooming, moving, *etc.*), (ii) mouse clicks on the graph's nodes (Annotation, Body, Target, *etc.*), which trigger the corresponding user actions (interactive operations) in the Data Updating Transformation.

User's click on an Annotation node will trigger the following task cases-based straightforward commands or interactive operations:

- *Modify the Annotation's provenance information:* Allowing the user to change the existing provenance information (*e.g.*, <Annotation> oa: annotatedBy <Person> and <Annotation> oa:serializedAt <Datatime>) attached to the Annotation.
- *Remove the entire Annotation:* Allowing the user to delete all annotation data related to the chosen Annotation.
- *Add a Body to the Annotation:* Allowing the user to add a new Body and its descriptive information to the chosen Annotation.
- *Add a Target to the Annotation:* Allowing the user to add a new Target and its descriptive information to the chosen Annotation.
- *Add another Annotation:* Allowing the user to add a new Annotation to the existing annotation dataset.
- *View the Annotation's OA data in RDF format:* Allowing the user to look at the Open Annotation data (in an RDF serialization format such as Turtle) relevant to the chosen Annotation.

User's click on a Body or Target node will trigger the following task cases-based straightforward commands or interactive operations:

- *Remove the Body/Target from the Annotation:* Allowing the user to delete the Body's/Target's provenance and descriptive information from the Annotation.
- *Modify the Body's/Target's descriptive information:* Allowing the user to change the existing descriptive data (*e.g.*, <Body> rdf:type oa:SemanticTag and <Target> ma:hasFormat "image/jpeg") attached to the Body/Target.
- *Modify the Body's/Target's provenance information:* Allowing the user to change the existing provenance information attached to the Body/Target.
- *Render the Body/Target resource:* Allowing the user to render the Body/Target resource, *e.g.*, playing a video.
- *View the Body's/Target's OA data in RDF format:* Allowing the user to look at the Open Annotation data (in the RDF format) relevant to the Body/Target.

As mentioned above, task cases-based user actions include two types of actions: *straightforward commands* or *interactive operations*. For a straightforward command (*e.g.*, a command for the "Remove the entire Annotation" task case), its interaction context is simply a confirmation prompt. In contrast, an interaction context for interactive operations contains more contents with a relatively complicated structure. To save space, here we only give an example to illustrate the design of this type of interaction contexts, within which the user performs desired actions and the system responds accordingly. The interaction context for the "Add a Body to the Annotation" task case should abstractly represent the user interface's contents and organization as follows:

- *Showing identifier of the Annotation:* The system should indicate the Annotation to which the Body is attached by showing the identifier of the Annotation.
- *Conveying motivation of annotation:* The user can convey the motivation of the annotation (*i.e.*, the reason why the Annotation is created using this Body) by simply choosing a motivation (*e.g.*, oa:describing) from a list in which twelve kinds of motivations specified in the W3C specification [10] have been placed. When the user chooses the oa:tagging motivation, he/she is required to further specify the type of the tag, such as oa:Tag, oa:SemanticTag, or foaf:page.
- *Specifying identifier for the Body:* The user can enter an identifier (*e.g.*, a resolvable HTTP URI or a non-resolvable URN) for the Body resource. If the user

does not have any Body resource used to describe the Target, he/she can search for an appropriate resource on the Web.

- *Render the Body resource:* The user can render the Body resource using a simple command.
- *Specifying content type for the Body:* The user can specify a content type for the Body resource by conveniently selecting one content type (*e.g.*, dctypes:MovingImage) from a list, where five kinds of general content types specified in the W3C specification [10] have been placed into.
- *Specifying media type for the Body:* The user can specify a media type for the Body resource by choosing a media type (*e.g.*, "video/x-ms-wmv") from a list.
- *Adding more descriptive properties of the Body:* If the user wants to further describe the Body resource using ontological vocabularies, he/she can give a command to activate a secondary interaction context, within which, for example, a multimedia resource can be further described using W3C's Media Resources Ontology [20] or other common ontologies.
- *View generated OA data:* If the user wants to view the automatically generated OA data in the RDF format, he/she can give a command to look at the data.
- *Cancelling annotation data and exiting:* The user should be given the option of cancelling the annotation data as well as exiting from the interaction context of the annotation tool.
- *Saving annotation data and exiting:* The user should be given the option of saving the annotation data and exiting from the interaction context. After the user gives the command by choosing this option, the system should save the newly generated annotation data and then update the annotation dataset, which results in refreshing the visual representation.

4. Prototype Implementation and Case Study

The proposed IVUIM4OA has been implemented as a proof-of-concept in our prototype multimedia annotation tool called OAMM (Open Annotator for MultiMedia resources). To validate the effectiveness of IVUIM4OA, we carried out several case studies with the prototype tool. In the following we first describe the implementation technologies including the visualization technique we employed in our prototype and then give our experimental results using a small-scale case study.

4.1. Implementation Technologies

OAMM is a Java Web application created using Java EE 5.0 and Apache Jena framework and deployed on the Apache Tomcat 7.0 server. OAMM's front-end is implemented using HTML, CSS and JavaScript and runs in a browser; its back-end is implemented in Java using JavaServer Pages (JSP) Model 2 architecture in order to effectively separate presentation from content. In OAMM, the main implementation technologies for IVUIM4OA are given in the following:

1) Data Transformation: Using the Apache Jena parser to transform an RDF dataset into RDF triples.

2) Visualization Transformation: Interpreting the RDF triples using the semantics for the OA data model, identifying OA elements and then forming OA graphic objects by instantiating the Java classes as listed in Table 1. The definitions of these Java classes are shown in Figure 2. These data structures are independent of particular visualization techniques. OA graphic objects are created by instantiating these Java classes.

Table 1. A List of Java Classes for OA Graphic Objects

Class	Description
ColoredEllipse	Ontology instances are depicted as colored ellipses
WhiteRectangle	Ontology classes are depicted as white rectangles
WhiteRoundRectangle	Literals are depicted as white round rectangles
StraightBlackArrowLine	Ontology properties are depicted as straight, black arrow lines
ClassInstantiationArrow	Class instantiation (rdf:type) is depicted as a straight, black line with a white arrow head

Figure 2. Definitions of the Java classes for OA Graphic Objects

```
Class ColoredEllipse {
    String id="";
    String nodeType=""; /* Including Body, Target, Annotation, and other */
    String shape="ellipse";
    String style="filled";
    String fillColor=""; /* Different node types use different colors */
    String fontName="";
    Integer fontSize=12;
    String nodeLabel=""; /* Including resolvable URIs, non- resolvable URNs, and blank nodes */
    ... // Getter and Setter Methods
}
Class WhiteRectangle {
    String id="";
    String shape="box";
    String style="none";
    String fillColor="white";
    String fontName="";
    Integer fontSize=12;
    String nodeLabel=""; /* Ontology class identifiers */
    ... // Getter and Setter Methods
}
Class WhiteRoundRectangle {
    String id="";
    String shape="roundRectangle";
    String style="none";
    String fillColor="white";
    String fontName="";
    Integer fontSize= 12;
    String nodeLabel=""; /* RDF literals */
    ... // Getter and Setter Methods
}
Class StraightBlackArrowLine {
    String id="";
    String sourceNodeId="";
    String targetNodeId="";
    String direction="forward";
    String arrowHead=""; /* Default indicates black arrow head */
    Integer lineWeight=8;
    String fontName="";
    Integer fontSize= 12;
    String edgeLabel=""; /* RDF predicates */
    ... // Getter and Setter Methods
}
Class ClassInstantiationArrow {
    String id="";
    String sourceNodeId="";
    String targetNodeId="";
    String direction="forward";
    String arrowHead="empty";
    Integer lineWeight=8;
    String fontName="";
    Integer fontSize=12;
    String edgeLabel="rdf:type";
    ... // Getter and Setter Methods
}
```

3) Visual Mapping Transformation: Mapping the graphic objects data to on-screen visual representation by using a particular visualization technique. We chose the JavaScript plugin Viz.js (https://github.com/mdaines/viz.js/) that can put the open source graph visualization software GraphViz (http://www.graphviz.org/) on the Web page. Figure 3 shows a segment of OA data in Turtle format [19] and the mapped OA graphic objects data to be loaded by Viz.js.

```
oamm:anno1  a  oa:Annotation;
   oa:hasTarget  <http://boonage.pjss2.net/images/content/olympic-lh-sb.jpg>.
<http://boonage.pjss2.net/images/content/olympic-lh-sb.jpg>
   dcterms:subject  "Performers of Olympic theme song".
```

(a)

```
digraph VisualGraph {
   // The mapped data corresponding to an object of ColoredEllipse.
   "oamm:anno1" [ id="anno1", shape=ellipse, style=filled, fillcolor="khaki1", fontname="Arial
   Narrow", fontsize=12 ];
   // The mapped data corresponding to an object of WhiteRectangle.
   "oa:Annotation" [ shape=box, style=none, fillcolor="white", fontname="Arial Narrow",
   fontsize=12 ];
   // The mapped data corresponding to an object of ColoredEllipse.
   "http://boonage.pjss2.net/images/content/olympic-lh-sb.jpg" [ shape=ellipse, style=filled,
   fillcolor="pink", fontname="Arial Narrow", fontsize=12 ];
   // The mapped data corresponding to an object of WhiteRoundRectangle.
   "Performers of Olympic theme song" [ shape=Mrecord, fillcolor="white", fontname="Arial
   Narrow", fontsize=12];
   // The mapped data corresponding to an object of ClassInstantiationArrow.
   "oamm:anno1" -> "oa:Annotation" [ dir="forward", arrowhead="empty", weight=8,
   fontname="Arial Narrow", fontsize=12, label="rdf:type" ]
   // The mapped data corresponding to an object of StraightBlackArrowLine.
   "oamm:anno1" -> "http://boonage.pjss2.net/images/content/olympic-lh-sb.jpg [ dir="forward",
   arrowhead="normal", weight=8, fontname="Arial Narrow", fontsize=12, label="oa:hasTarget" ];
   // The mapped data corresponding to an object of StraightBlackArrowLine.
   "http://boonage.pjss2.net/images/content/olympic-lh-sb.jpg" -> "Performers of Olympic theme
   song" [ dir="forward", arrowhead="normal", weight=8, fontname="Arial Narrow", fontsize=12,
   label="dcterms:subject" ];
}
```

(b)

Figure 3. (a) A Segment of OA Data in the Turtle RDF Format; (b) the Mapped OA Graphic Objects Data to be Loaded by Viz.js

4) View In-Stage Operators: Adding event listeners to the objects in the JSP page. The events that the listeners are triggered including:

- *Mouse manipulations of the visual graph or its objects:* We chose to use JavaScript's plugin jQuery UI v1.10.4 (http://jqueryui.com/), which is built on top of the jQuery JavaScript Library, to achieve object manipulations such as scaling, zooming, moving, and rendering.
- *Mouse clicks on the graph's nodes (Annotation, Body, Target, etc.):* We chose to use JavaScript's jQuery alert plugin v1.0.3 (https://code.google.com/p/jquery-alert/) to trigger the interactive operations in the Data Updating Transformation.

5) Data Updating Transformation: The triggered interactive operations rely on HTML form-based technology implemented in the JSP pages. User interacts with the HTML forms in popup windows, and Java codes automatically produce new annotation data or modify the existing one. Data updating will result in refreshing the View.

Note that IVUIM4OA can be realized by using other implementation technologies and visualization techniques, not necessarily the above technologies used in this paper.

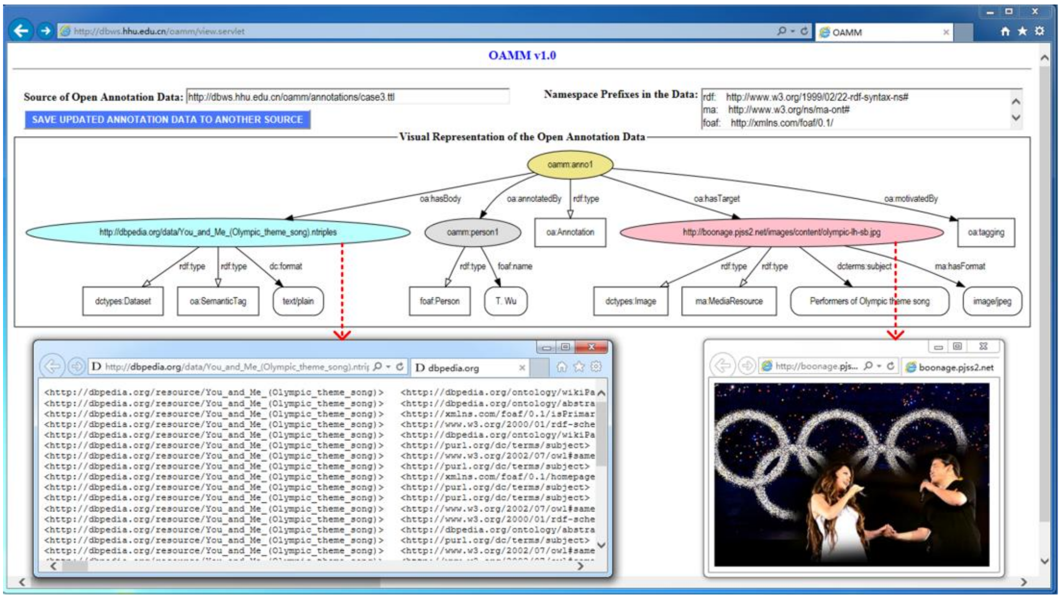
4.2. Case Study

We have used the OAMM tool to conduct several case studies with both the examples created by our group members and some examples from the W3C's Annotation Cookbook (http://www.w3.org/community/openannotation/wiki/Cookbook). All case studies have obtained a satisfactory outcome. Saving space, here we give a small example created by ourselves. Figure 4 shows an example of Open Annotation dataset in the Turtle RDF format [19], where the Annotation (oamm:anno1) has one Body and one Target, and the Body resource (http://dbpedia.org/data/You_and_Me_(Olympic_theme_song).ntriples) is used to semantically tag, by means of oa:SemanticTag, the Target resource (http://boonage.pjss2.net/images/content/olympic-lh-sb.jpg). Taking this dataset as input, the OAMM tool produced an on-screen visual representation of the Open Annotation data, as shown in Figure 5.

```
@prefix dc:  <http://purl.org/dc/elements/1.1/>.
@prefix dctypes:  <http://purl.org/dc/dcmitype/>.
@prefix dcterms:  <http://purl.org/dc/terms/>.
@prefix foaf:  <http://xmlns.com/foaf/0.1/>.
@prefix ma:  <http://www.w3.org/ns/ma-ont#>.
@prefix oa:  <http://www.w3.org/ns/oa#>.
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix oamm: <http://dbws.hhu.edu.cn/oamm/annotations/case3#>.
oamm:anno1  a  oa:Annotation;
   oa:hasTarget  <http://boonage.pjss2.net/images/content/olympic-lh-sb.jpg>;
   oa:hasBody
      <http://dbpedia.org/data/You_and_Me_(Olympic_theme_song).ntriples>;
   oa:motivatedBy  oa:tagging;
   oa:annotatedBy  oamm:person1.
<http://boonage.pjss2.net/images/content/olympic-lh-sb.jpg>
   a  dctypes:Image;
   a  ma:MediaResource;
   dcterms:subject  "Performers of Olympic theme song";
   ma:hasFormat  "image/jpeg".
<http://dbpedia.org/data/You_and_Me_(Olympic_theme_song).ntriples>
   a  dctypes:Dataset;
   a  oa:SemanticTag;
   dc:format  "text/plain".
oamm:person1
   a  foaf:Person;
   foaf:name  "T. Wu".
```

Figure 4. An Example of OA Data in the Turtle RDF Format

On Figure 5's screen, the source URI and the namespace prefixes used in the Open Annotation RDF dataset have been listed in the top. The visual graph has been displayed in the middle of the screen, where all OA graphic objects corresponding to the OA elements in the annotation dataset have been accurately visualized. The Body resource, a DBpedia's RDF dataset in the N-Triples format, has been opened in a browser by mouse clicking the Body node. And the Target resource, a picture of the performers (Liu Huan and Sarah Brightman) of the 2008 Beijing Summer Olympics, has also been displayed by mouse clicking the Target node. This results show that our IVUIM4OA-based annotation tool (*i.e.*, OAMM) can effectively support the visualization of Open Annotation data. Note that when the user employs the tool to start a new annotation task, that is to say, no Open Annotation data currently exists, the tool can initialize a visual graph for the user to interact with by adding a Annotation (*e.g.*, oamm:anno2 a oa:Annotation) to a new dataset.


http://dbws.hhu.edu.cn/oamm/view.servlet
OAMM
OAMM v1.0
SAVE UPDATED ANNOTATION DATA TO ANOTHER SOURCE
Visual Representation of the Open Annotation Data
oamm:anno1
oa:hasBody
oa:annotatedBy
rdf:type
oa:hasTarget
oa:motivatedBy
http://dbpedia.org/data/You_and_Me_(Olympic_theme_song).ntriples
oamm:person1
oa:Annotation
http://boonage.pjss2.net/images/content/olympic-lh-sb.jpg
oa:tagging
rdf:type
rdf:type
dc:format
rdf:type
foaf:name
rdf:type
rdf:type
dcterms:subject
ma:hasFormat
dctypes:Dataset
oa:SemanticTag
text/plain
foaf:Person
T. Wu
dctypes:Image
ma:MediaResource
Performers of Olympic theme song
image/jpeg
http://dbpedia.org/data/You_and_Me_(Olympic_theme_song).ntrip
dbpedia.org
http://boonage.pjs...
boonage.pjss2.net

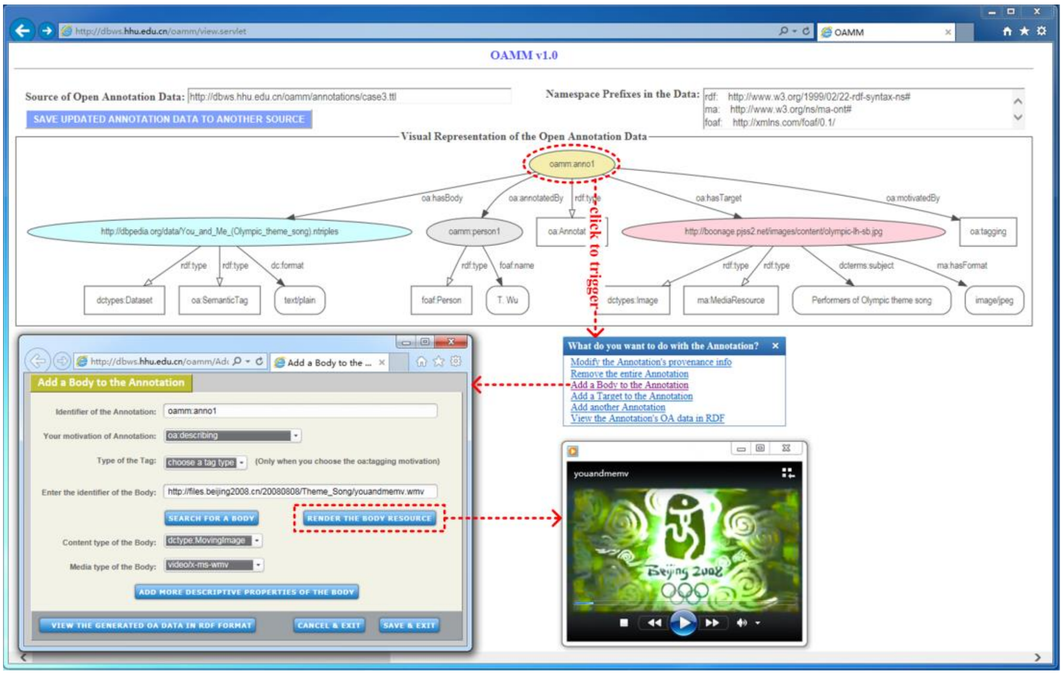
Source of Open Annotation Data: http://dbws.hhu.edu.cn/oamm/annotations/case3.ttl

Namespace Prefixes in the Data:
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
ma: http://www.w3.org/ns/ma-ont#
foaf: http://xmlns.com/foaf/0.1/

```
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://dbpedia.org/ontology/wikiPa
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://dbpedia.org/ontology/abstra
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://xmlns.com/foaf/0.1/isPrimar
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://www.w3.org/2000/01/rdf-sche
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://dbpedia.org/ontology/wikiPa
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://purl.org/dc/terms/subject>
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://www.w3.org/2002/07/owl#same
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://purl.org/dc/terms/subject>
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://xmlns.com/foaf/0.1/homepage
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://purl.org/dc/terms/subject>
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://purl.org/dc/terms/subject>
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://www.w3.org/2002/07/owl#same
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://www.w3.org/2000/01/rdf-sche
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://dbpedia.org/ontology/abstra
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://purl.org/dc/terms/subject>
<http://dbpedia.org/resource/You_and_Me_(Olympic_theme_song)>  <http://www.w3.org/2002/07/owl#same
```

Figure 5. Screenshot of the Visual Representation of the OA Data Shown in Figure 4, where the Body and Target Resources have also been Displayed

Now, in the following we demonstrate how IVUIM4OA can also be used as an *interaction model* to perform interactive annotation operations. Suppose we want to add another Body to the oamm:anno1 Annotation, say, a video resource of the Olympic theme song on the Web, in order to further describe the Target resource. As shown in Figure 6, the user can employ the OAMM tool to click the Annotation node (oamm:anno1) to trigger popup of the "What do you want to do with the Annotation?" prompt. After the user chooses "Add a Body to the Annotation" item there, the related window appears. Within the interaction context of the window, the user can convey the motivation of this annotation by simply choosing a motivation (*e.g.*, oa:describing) from the related list, where twelve kinds of motivations specified in the W3C specification [10] have been placed in for the user to select. When the user chooses the oa:tagging motivation, he/she is required to further specify the type of the tag, for instance, oa:Tag, oa:SemanticTag, or foaf:page. In addition, the user is required to enter an identifier (*e.g.*, a resolvable HTTP URI or non-resolvable URN) for the Body resource (here is http://files.beijing2008.cn/20080808/Theme_Song/youandmemv.wmv). If the user does not have any Body resource that can be used to describe the Target, he/she can press the "SEARCH FOR A BODY" button to search for an appropriate resource on the Web. The chosen Body resource can then be rendered by pressing the "RENDER THE BODY RESOURCE" button. Finally, it requires the user to specify the content type, media type, and other descriptive properties of the Body resource. Similar to annotation motivation, a content type (*e.g.*, dctypes:MovingImage) can easily be selected from the related list in which five kinds of general content types specified in the W3C specification [10] have been placed. Similarly, a media type (*e.g.*, "video/x-ms-wmv") can be selected from the related list. If the annotating user wants to further describe the Body resource using some ontological vocabularies, he/she can press the "ADD MORE DESCRIPTIVE PROPERTIES OF THE BODY" button to activate another popup window (interaction context), where, for example, the multimedia resource can be further described using W3C's Media Resources Ontology [20].



http://dbws.hhu.edu.cn/oamm/view.servlet

OAMM v1.0

Source of Open Annotation Data: http://dbws.hhu.edu.cn/oamm/annotations/case3.ttl

Namespace Prefixes in the Data:
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
ma: http://www.w3.org/ns/ma-ont#
foaf: http://xmlns.com/foaf/0.1/

What do you want to do with the Annotation?
Modify the Annotation's provenance info
Remove the entire Annotation
Add a Body to the Annotation
Add a Target to the Annotation
Add another Annotation
View the Annotation's OA data in RDF

Add a Body to the Annotation

Identifier of the Annotation: oamm:anno1
Your motivation of Annotation: oa:describing
Type of the Tag: choose a tag type (Only when you choose the oa:tagging motivation)
Enter the identifier of the Body: http://files.beijing2008.cn/20080808/Theme_Song/youandmemv.wmv
Content type of the Body: dctype:MovingImage
Media type of the Body: video/x-ms-wmv

Figure 6. Screenshot of the "Add a Body to the Annotation" Interactive Interface Triggered by User's Click on Node "Oamm: Anno1" in the Visual Graph

Finally the user can press "VIEW THE GENERATED OA DATA IN RDF FORMAT" to look at the automatically generated OA data. After the user presses "SAVE & EXIT", the tool saves the newly generated Open Annotation data and then updates the annotation dataset, which results in refreshing the visual representation on the screen. In our example, the generated OA data in the Turtle RDF format, through the "Add a Body to the Annotation" interface (Figure 6), is given in Figure 7.

The above case-study experimental results indicate that the interaction mechanisms and contexts in our IVUIM4OA model are achievable and our IVUIM4OA-based annotation tool can help the user to perform interactive annotating operations expediently.

```
oamm:anno1  a  oa:Annotation;
   oa:hasTarget  <http://boonage.pjss2.net/images/content/olympic-lh-sb.jpg> ;
   oa:hasBody
<http://files.beijing2008.cn/20080808/Theme_Song/youandmemv.wmv>;
   oa:motivatedBy  oa:describing.
<http://files.beijing2008.cn/20080808/Theme_Song/youandmemv.wmv>
   a  dctypes:MovingImage;
   ma:hasFormat  "video/x-ms-wmv".
```

Figure 7. The Generated OA Data in the Turtle RDF Format

5. Discussion and Future Work

The W3C Open Annotation (OA) Data Model is emerging as a standardized Web annotation data model. Before widespread adoption of the OA data model, the community should make an effort to develop easy-to-use tools for wider annotating users, especially for lay-users without sufficient techniques and experience. Undoubtedly, a friendly interactive visual user interface, which is largely based upon an effective user interface model, plays an essential role in the usability of an Open Annotation tool. In this paper we have proposed an Interactive Visual User Interface Model for Open Annotations

(IVUIM4OA), which allows the annotating user to dynamically transform OA data into a visual representation and to interactively manipulate the visual objects to modify the OA data or produce new one. Our implementation and case-study experimental results indicate that the proposed IVUIM4OA model is feasible and works well with the W3C OA data model. Through the IVUIM4OA-based interactive visual user interface, annotating users can obtain a better understanding of and more intuitive interaction with Open Annotation data, particularly in the context of multimedia annotations.

Conceptually, IVUIM4OA is a general purpose, essential (*i.e*., abstract, simplified, and technology-free) model for user interface design of Open Annotation tools. It is independent of the particular implementation technologies and visualization techniques used in this paper. Also, because it focuses on abstract user intentions and usage patterns, software developers can exploit this essential model to systematically design and agilely construct the smallest, simplest (*i.e*., lightweight and easy-to-use) Open Annotation tool that fully and directly supports all the tasks the annotating user need to accomplish. In a word, IVUIM4OA can be widely applied to any Web Annotation tools that employ the W3C Open Annotation Data Model—the basis for the Web Annotation Data Model specification under development by the newly created W3C Web Annotation Working Group.

Our future work focuses on enhancing the IVUIM4OA-based annotation tool (*i.e*., OAMM) by integrating OAMM's interactive interfaces with search/query interfaces for annotating user's access to more URI-identified resources in the Web of Linked Open Data, enabling the user to conveniently choose the most common Web resources (*e.g*., semantically rich structured data in DBpedia and well-described media resources in Wikimedia Commons), as a Body, to annotate the Target resources. The further work has been producing some encouraging results.

Acknowledgements

This work was partially supported by the following research grants: (1) No. BK20141420 and No. BK20140857 from the Natural Science Foundation of Jiangsu Province, China; (2) No. 61272261 from the Natural Science Foundation of China; (3) No. LY14F020036 from the Natural Science Foundation of Zhejiang Province, China, as well as by a project funded by the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions, China.

References

[1] G. Stamou, J. V. Ossenbruggen, J. Z. Pan, and G. Schreiber, "Multimedia annotations on the semantic web", IEEE MultiMedia, vol. 13, no.1, **(2006)**, pp. 86-90.

[2] S. Dasiopoulou, E. Giannakidou, G. Litos, P. Malasioti, and Y. Kompatsiaris, "A survey of semantic image and video annotation tools", In Knowledge-Driven Multimedia Information Extraction and Ontology Evolution, G. Paliouras, C. D. Spyropoulos, G. Tsatsaronis, Eds.; Lecture Notes in Computer Science, vol. 6050, Springer, Heidelberg, Germany, **(2011)**, pp. 196-239.

[3] P. Ciccarese, S. Soiland-Reyes, and T. Clark, "Web annotation as a first-class object", IEEE Internet Computing, vol. 17, no. 6, **(2013)**, pp. 71-75.

[4] T. B. Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", IETF RFC 3986, the Internet Society, Available at: http://tools.ietf.org/html/rfc3986. (Accessed 8 March 2015), **(2005)**.

[5] R. Cyganiak, D. Wood and M. Lanthaler, "RDF 1.1 Concepts and Abstract Syntax", W3C Recommendation, Available at: http://www.w3.org/TR/rdf11-concepts/. (Accessed 8 March 2015), **(2014)**.

[6] T. Heath, "Linked data - welcome to the data network", IEEE Internet Computing, vol. 15, no. 6, **(2011)**, pp. 70-73.

[7] Y. Li, M. Wald, and G. Will, "Applying linked data in multimedia annotations", International Journal of Semantic Computing, vol. 6, no. 3, **(2012)**, pp. 289-313.

[8] P. Ciccarese, M. Ocana, L. Castro, S. Das, and T. Clark, "An open annotation ontology for science on web 3.0", Journal of Biomedical Semantics, 2 (Suppl 2): S4, doi:10.1186/2041-1480-2-S2-S4, **(2011)**.

[9] B. Haslhofer, R. Simon, R. Sanderson and H. Sompel, "The open annotation collaboration (OAC) model", In Proceedings of 2011 Workshop on Multimedia on the Web (MMWeb2011), Graz, Austria, September 8, 2011; IEEE Computer Society, Piscataway, NJ, USA, **(2011)**, pp. 5-9.

[10] R. Sanderson, P. Ciccarese, and H. Sompel, "Open Annotation Data Model. W3C Community Draft, Available at: http://www.openannotation.org/spec/core/. (Accessed 8 March 2015), **(2013)**.

[11] R. Sanderson and P. Ciccarese, "Web Annotation Data Model. W3C First Public Working Draft, Available at: http://www.w3.org/TR/annotation-model/. (Accessed 8 March 2015), **(2014)**.

[12] F. Murtagh, T. Taskaya, P. Contreras, J. Mothe, and K. Englmeier, "Interactive visual user interfaces: a survey", Artificial Intelligence Review, vol. 19, no. 4, **(2003)**, pp. 263-283.

[13] A. S. Dadzie and M. Rowe, "Approaches to visualizing linked data: a survey", Semantic Web, vol. 2, no. 2, **(2011)**, pp. 89-124.

[14] E. H. Chi, "A taxonomy of visualization techniques using the data state reference model", In Proceedings of the IEEE Symposium on Information Visualization (InfoVis2000), Salt Lake, UT, USA, October 9, 2000; IEEE Press, Piscataway, NJ, USA, **(2000)**, pp. 69-75.

[15] J. Kahan, M. R. Koivunen, E. P. Hommeaux, and R. R. Swick, "Annotea: An open RDF infrastructure for shared Web annotations", Computer Networks, vol. 39, no. 5, **(2002)**, pp. 589-608.

[16] Y. Li, M. Wald, T. Omitola, N. Shadbolt and G. Wills, "Synote: weaving media fragments and linked data", In Proceedings of WWW 2012 Workshop on Linked Data on the Web (LDOW 2012), Lyon, France, April 16, 2012; C. Bizer, T. Heath, T. Berners-Lee, and M. Hausenblas, Eds; CEUR Workshop Proceedings (CEUR-WS.org), Available at: http://ceur-ws.org/Vol-937/ldow2012-paper-01.pdf. (Accessed 8 March 2015), vol. 937, **(2012)**.

[17] J. M. Brunetti, S. Auer, R. García, J. Klímek, and M. Nečaský, "Formal linked data visualization model", In Proceedings of 15th International Conference on Information Integration and Web-Based Applications & Services, IIWAS '13, Vienna, Austria, December 2-4, 2013; E. R. Weippl, M. Indrawan-Santiago, M. Steinbauer, G. Kotsis, and I. Khalil, Eds.; ACM Press, New York, NY, USA, **(2013)**, pp. 309-318.

[18] L. L. Constantine and L. A. D. Lockwood, "Usage-centered engineering for Web applications", IEEE Software, vol. 19, no. 2, **(2002)**, pp. 42-50.

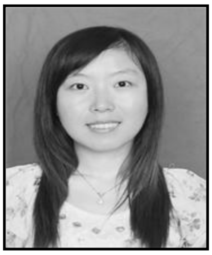
[19] E. Prud'hommeaux and G. Carothers, "RDF 1.1 Turtle: Terse RDF Triple Language", W3C Recommendation, 25 February 2014, Available at: http://www.w3.org/TR/turtle/. (Accessed 8 March 2015), **(2014)**.

[20] W. Lee, W. Bailer and T. Bürger, "Ontology for Media Resources 1.0. W3C Recommendation", 09 February 2012, Available at: http://www.w3.org/TR/mediaont-10/. (Accessed 8 March 2015), **(2012)**.

Authors

Zhuoming Xu, He received his BSc and MSc in computer science from Hohai University, China, in 1986 and 1994 respectively, and his PhD in computer science from Southeast University, China, in 2005. He was a visiting professor at Indiana University Bloomington, USA, from May 2011 to May 2012. He has been a full professor in College of Computer and Information at Hohai University for twelve years since 2004. His current research interests include Web data management and multimedia annotation.

Wenping Guo, He received his BSc in mechatronic engineering and MSc in computer science from Southwest Jiaotong University, China, in 1998 and 2005 respectively. He was a visiting scholar at Jacksonville State University, USA, from February 2012 to June 2012. He is currently an associate professor at Taizhou University, China and a PhD candidate in computer science at Hohai University, China, with research interests including multimedia annotation and Web annotation models.



Ting Wu, She received her BSc in computer science from Hohai University, China, in 2012. She is currently a graduate student in College of Computer and Information at Hohai University. Her current research interests focus on multimedia annotation and media fragment retrieval.

Lixian Ni, He received his BSc and MSc in computer science from Hohai University, China, in 2011 and 2014 respectively. His current research interests focus on multimedia annotation, visualization technology and graphical user interfaces.