3D-SYSTEK: Recording and exploiting the production workflow of 3D-models in Cultural Heritage

Anastasia Axaridou, Ioannis Chrysakis, Christos Georgis, Maria Theodoridou, Martin Doerr
Institute of Computer Science
Foundation for Research and Technology – Hellas
Heraklion, Greece
{axaridou, hrysakis, georgis, maria, martin}@ics.forth.gr

Antonios Konstantaras, Emmanuel Maravelakis
School of Applied Sciences
Technological Educational Institute of Crete
Chania, Greece
{akonstantaras, marvel}, @chania.teicrete.gr

***Abstract*—The diversity of contemporary technology on 3D-model digitizing and processing procedures necessitates the systematic documentation of all the involved activities. In this paper we present essential concepts and the infrastructure of 3D-SYSTEK (3DS), a system that supports the 3D-modelling provenance preservation in the Cultural Heritage (CH) domain. The proposed system provides an efficient repository and special tools for ingesting and browsing data, supporting the detailed and effective documentation. Specialists working on 3D-model production are able to record the production steps, keep track of their work and recall conditions and processing methods for reproduction. Additionally, CH scientists and researchers are able to browse, retrieve and annotate related CH data. Hence 3D-SYSTEK becomes a powerful tool in the area of 3D-model production, archiving and dissemination.**

Keywords—3D-model production; production workflow; 3D-modelling; data provenance; semantic network; cultural heritage preservation

I. INTRODUCTION

The 3D-modelling process involves several manual and automatic procedures. It starts with the digitization (photography, scanning) of physical objects either movable or immovable, and then continues with a variety of processing steps, to produce the desired 3D-model [1]. Most steps require software and hardware setup arrangements and produce bulks of data. We designed and implemented 3D-SYSTEK (3DS), a core data management system that supports the archiving and dissemination of data and metadata involved in the 3D-model production. The necessity of recording all the steps of production in a well-organized manner requires the selection of an appropriate documentation methodology. We propose the use of metadata schemas based on CIDOC-CRM[1], a well-established standard in the CH domain.

In the following sections we present the basic concepts, architecture and functionality of the 3DS repository. We also introduce ReposIt and BrowseIt two important tools used for controlling ingestion and powering browsing, respectively.

1. http://www.cidoc-crm.org

II. RELATED WORK

Managing of digital CH content is a multidimensional and complex problem that has been addressed by various approaches in specialized contexts and needs. Dspace[2] preserves and enables easy and open access to all types of digital content including text, images, moving images, and data sets. It is applied for accessing, managing and preserving scholarly works. The Fedora[3] digital repository provides a flexible digital content repository which can be adapted to a wide variety of scenarios and can store any kind of digital content including images, videos, datasets, together with a complex network of relationships linking the digital objects to each other. However systems like DSpace and Fedora do not support the workflow of processes and are bound to their own data management philosophy that cannot exploit provenance information. In addition, the generated metadata are produced in Dublin Core[4] which can capture only basic information with limited expressivity.

There are also systems focusing only on the metadata creation of digital objects [2] or on the metadata aggregation of cultural content [3]. In [2], the Metadata Generator tool is implemented for generating cultural heritage metadata following the CIDOC-CRM standard through generic dynamic input forms. The drawback of this approach is that it allows duplicates, violating this way the referential integrity. In [3] a web-based system is presented that provides content providers and users the ability to map, in an effective way, their own metadata schemas to common domain standards and models like Europeana[5]. Based on these mappings, semantic enrichment and query answering techniques are proposed as a means for providing effective access of users to digital CH.

Our proposed approach is a continuation of previous work done in the 3D-COFORM project [4] where an integrated repository was developed to store and manage 3D-models ensuring the semantic integrity of the content. Our present work includes major improvements and vital new features such as efficient uploads based on FTP capabilities (e.g. resume broken file transfers), restore connection during data transfer operation, support of asynchronous data ingests to enable parallel work, reliable transaction management,

2. http:// www.dspace.org
3. http://www.fedora-commons.org
4. http://www.dublincore.org
5. http://europeana.eu

garbage manipulation, including complete reimplementation of core components.

III. System Overview

3DS supports the archiving and dissemination of all the information involved in 3D-model production. An important advantage of the system is the recording of knowledge regarding the 3D-model production workflow, using RDF Semantics technology[6] and complying with the LOD rules [5]. The documentation of procedures in all phases of the workflow based on an *event-centric* approach offers significant flexibility and efficiency in the way that information is recorded and managed. Fundamental principles that rule the operation of 3DS satisfy the demand for consistency and reliability of 3D-modelling documentation.

IV. Entities And Principles

3DS entities are classified into 3 main conceptual types:

1) ***Data** objects:* They include the produced 3D-models as well as any 2D-image, text or other digital material involved in the 3D-model production process.

2) ***Area** objects:* One or more parts or items, representing a meaningful internal set of components of the Data object which further can be annotated and conceptually related to other resources. Specific parts of 3D-models with particular interest are expressed with this entity, in order to be classified or related to other areas, models, persons, places, etc.

3) ***Metadata** objects:* Sets of RDF triples that carry the information about all semantic entities related to 3D-models or involved in the 3D-model production. They are used in an event-centric logic for: (i) The step-by-step recording of the acquisition and process events, (ii) The description of other semantic entities such as persons, legal bodies, places, devices, terminology, typology, etc. participating in events' documentation and (iii) The creation of annotations for Data, Areas of Data or any other resource of the Semantic Network.

The entities described above comprise the essential types of resources managed by 3DS. Their representation in the system is attained via specific types:

- A Data object is a file of any file format.
- Metadata objects are files of any RDF format, describing the 3D-model production in accordance to data models (schemas) based on the CIDOC-CRM ontology and its CRMdig extension [6]. The basic unit of a Metadata object is a *triple* linking Data objects and semantic entities with properties. The triples are ingested into a triple-store constituting a dynamic Semantic Network that carries, unifies and interconnects the knowledge, providing a powerful knowledge repository on 3D-model provenance.
- An Area object is a file with area definition compliant to the *mets:area* element of the extensible METS[7] standard. We defined extensions to METS in order to cover any missing area properties (such as the *COLLADA*[8] *format* extension to define 3D-areas and the *HTML5 range* to define areas in Web-pages).

The 3DS operation regarding the complete and accurate recording of 3D-model creation adheres to specific **conceptual** and **operation principles**:

Principle-1: The ingested Data objects are always coupled with their respective Metadata objects in order to ensure the data provenance. Thus, ingestion of a Data object without its related Metadata object is forbidden.

Principle-2: The ingested Data objects are never updated or deleted from the repository. History preservation and referential and semantic integrity requirements have to be protected for complete and reliable provenance archiving. A modified Data object e.g. a modified image is considered as a new Data object derived from the original one with a process event. This new Data object will be ingested in the 3DS repository coupled with its production event. Update or Delete functions for Data objects are not provided by the 3DS. Special handling for removing wrong Data objects is carried out by the 3DS administrator.

Principle-3: Metadata objects are updateable and versioned but it is forbidden to be deleted from the 3DS repository. Information about events and other entities can be enriched producing new Semantic Metadata versions in the 3DS.

Principle-4: Areas of Data are of two types. The *primary* Area objects are physical areas originally defined on a Data object. The *propagated* Area objects are the translation of primary areas on a different Data object which is a derivative or another instance of the original one. Both *primary* and *propagated* Areas share the same identifier although geometrically and structurally are defined with different METS xml files.

Principle-5: Annotation of Area objects is semantically propagated to all instances of the Area object.

Principle-6: Area objects of any type can be modified or deleted unless they are annotated.

Principle-7: All three conceptual 3DS entities presented above are realized in the system with files. These files are represented and managed by specific application and database structures (file-structures). Additionally to the files, proper repository structures with special attributes are considered for Area and Metadata objects.

Principle-8: The ingestion of 3DS entities involves two distinct operations. First comes the creation of the appropriate Data or Metadata file and other entity structures in 3DS repository and secondly, the upload of the file stream to the file-store. This design implies that ingestion functionality can operate in synchronous or asynchronous mode regarding the file transfer completion. Very big files can be transferred asynchronously while new ingestions or other operations can take place in parallel.

All the above principles ensure the reliable and consistent operation of the 3D-model provenance repository.

6. http://www.w3.org/RDF
7. http://www.loc.gov/standards/mets/
8. https://collada.org

V. System Design And Implementation

A. System Architecture

3DS is based on a client-server architecture. An overview of 3DS architecture is presented in Fig 1.

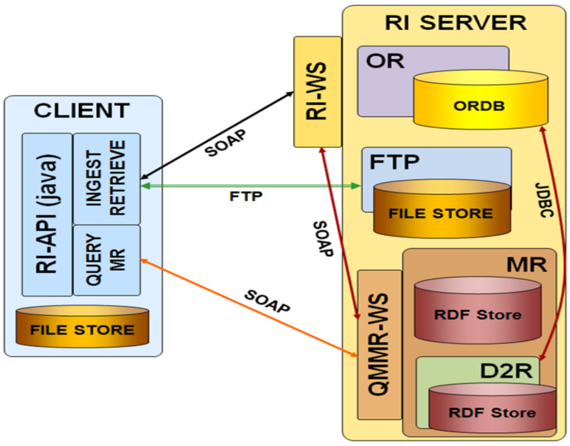

Fig. 1. 3DS system architecture

The **server-side** consists of four essential components comprising the 3DS Repository Infrastructure Server (RI Server):

1) The Object Repository (OR). This component is responsible for holding the appropriate records about the ingested files. It provides database management over file structures of Data, Metadata, and Area objects. It also implements the application logic (concepts and principals) to database structures to describe the involved 3DS objects. For example OR is responsible to correlate Data with their Metadata objects and Data with their Area objects as well as to manage the Metadata object versions.

2) The Metadata Repository (MR). This component is responsible for managing the semantic information that is derived from the ingested Metadata objects. Metadata objects are RDF files adhering to a specific data model. They are stored in the OR being treated as common file structures and at the same time, they are ingested in the MR contributing in the construction of the 3DS Semantic Network. As a result a powerful repository of knowledge is developed regarding activities, people, places, organizations, and data, all of them involved directly or not in the 3D-model production. An important component, integrated into the MR, is the Query Manager (QM), used to enable complex querying on different data sources: the OR database and the MR triple-store. QM accepts SPARQL queries, splits them in appropriate components and forwards them to the OR and MR accordingly. QM collects the results from each repository and returns the intersected list to the user. MR and QM are published under the QMMR Web Service (QMMR WS): the front-end for communicating with the semantic repository infrastructure. To enable SPARQL querying for the OR relational database another component, called D2R9 server, is used by QM. D2R server is a RDF wrapper to relational databases applying specific mapping of the relational database to the D2RQ schema.

3) The RI central Web Service (RI-WS). A central web service is the front-end of the RI server and serves the client-server communication. It is a SOAP end-point that provides the interface for ingesting, updating, retrieving and querying repository data. All 'write' operations are managed by the RI-WS which is responsible to implement the RI business logic before proceeding to OR-MR structure changes. This tier has the overview of the repository activity synchronizing the requested operations in the context of user-defined transactions. All the activity performed in OR-MR structures is recorded for enabling rollback and restoration of the structures, as well as the establishment of the new changes on transaction commit.

4) The FTP server and file-store. On ingest of a Data and Metadata object, apart from OR and MR structures changes, the file streams of the related (data and metadata) files must be uploaded on the server in order to be preserved and always accessible to the users. Thus the files are directly transferred from the client to the configured FTP server used as an online file-store. The files, stored there, can be directly accessed via FTP URLs in read-only mode.

The involved server-side applications and services are *Java-based*, hosted on the *Apache Tomcat* web application server. The OR relational database used is *mySQL server* and the MR triple-store backend is the OpenRDF Sesame[10].

At **client-side** a 3DS-client library provides the Java API for enabling the available operation of the RI. The library is responsible to support the communication of the client with the online RI-WS via SOAP messages. Login using the appropriate RI-WS URL is mandatory in order to be able to work with 3DS.

B. Functionality

3DS provides a Java-API which consists of several methods that support all 3DS functionality allowing the tool developers to implement their own integrated applications. This functionality comprises:

1) Login/logout and session management. A new session is created when an authorized user logs in to RI, being the passport for every requested action.

2) Ingestion of Data coupled with Metadata. A Data file is always ingested coupled with a Metadata file that describes its provenance. Appropriate file structure entries are created and stored into the OR for each file. Moreover the Metadata file is ingested to the MR. Finally, the file stream of each file is uploaded to the FTP file-store.

3) Ingestion of single Metadata. A single Metadata file describing any other resource (but not a Data object) can be ingested in the same way to the 'coupled' case.

4) Update of any Metadata file. Metadata files can be overwritten with a newer version. The update procedure is regarded as the regular ingestion of a new Metadata file along

9. http://d2rq.org/d2r-server

10. http;//www.openrdf.org

with the removal of the previous version in the MR.

5) Ingestion/deletion of Areas of Data. Area files can be ingested and deleted from the OR. On Area ingest both appropriate file and area structures are created in the OR; Metadata about the type of Area-Data relation (primary/propagated) are ingested as well to the MR. On deleting an Area all area structures are removed from the OR/MR.

6) Querying Data and Metadata. Queries are enabled on the entire repository or separately in either the OR or the MR. Special SPARQL query functions support particular queries in the MR, such as: describe/construct and ask queries.

7) Retrieving Data, Metadata and Areas as application structures. Direct retrieval of specific entity structures such as file structures for files, metadata version structures and area structures can be done with appropriate functions.

*8) Download any repository file by id. D*ownload a file to the local disk. A query may be used to find the desired identifier.

9) Support of user-defined transactions. Ingestion and update actions cause permanent changes to the repository. Hence it is recommended that such actions are performed in the context of user-defined transactions. In addition to the low-level transactions that are used to prevent illegal database situations, the user-defined transactions preserve the semantic integrity. The last is important for preventing situations such as Data ingested without Metadata, Areas of missing Data objects, annotation Metadata about unknown entities, etc.

10) Thumbnails can be uploaded by the user for the ingested Data object in order to enable previewing with browser tools. Thumbnail creation is automatically activated on ingest of a new image file producing a thumbnail. The automatically created thumbnail can be replaced anytime.

11) Re-upload of file streams. There are cases, usually of big files, where the file transfer is not completed. Hence, ingested Data objects may exist in the repository without their actual content. The system supports the re-upload of a stream and the proper update of its internal structures. The 'resume' option can be enabled to avoid upload from the beginning.

12) Garbage collection of left-overs from unterminated operations is supported as well. Open inactive sessions and transactions due to unexpected conditions i.e. a network connection failure, are cleaned with batch procedures.

13) Transaction management. Both database and user-defined transactions ensure the consistency of the repository data. The user-defined transactions are related to a user session and are considered as valid only during session life. After session release, the related open transactions are treated as garbage. The transactions keep track of the OR database and the MR internal activity. Changes on distinct rows of individual database tables are recorded during transaction to be handled later on commit or rollback.

14) Error handling mechanism. Error codes and related messages are classified according to the system components that catch the unexpected or unacceptable situations. Thus errors are distinguished to OR errors, database errors, MR errors, QM errors, file transfer errors, etc.

15) Finally, several other *utility functions* are provided such as for 'touching' active sessions to extend their lives, checking repository identifier existence, checking for left open transactions to be closed, etc.

VI. TOOLS

A. The ReposIt – Tool

Data objects in 3D-model production are typically created in individual procedures forming workflows with complex structure. Usually these procedures involve iterative operations and many steps (without predefined order) by one or more users. Additionally, during the procedures the generation of metadata files is needed to ensure the integrity of data and capture useful provenance information. However, the manual creation of metadata seems hard and inefficient while the import of files from an external generator tool sets clearly integration issues. Moreover, sometimes users need to update information for entities (e.g. edit responsible organization for a person etc.) or whole events (i.e. for annotation reasons, change input/output of events etc.). In other cases it is meaningful to batch-ingest files with the same event setup.

The ReposIt tool addresses these requirements that arise from real time recording of 3D-model production workflows. The tool provides user-friendly forms that can be filled, in any order, to support back and forth steps in the whole procedure. Specifically, it limits the possible complexity of a particular process to at most one level of sub-processes. The metadata are generated automatically for one entity or an event. The last case includes more than one metadata file to maintain the processing chain that led from the physical object to its 3D-model. It also ensures, the correlation between the objects involved in the processing chain and the information kept in each stage keeping this way the data provenance.

The functional characteristics of the ReposIt tool allow the ingestion or update of any acquisition or process event by exploiting the basic principles of the 3DS infrastructure. Each ingest can be either *synchronous or asynchronous* and accompanied with the *automatic generation of metadata files* that resulted from validated form fields. The tool ensures *referential integrity of data* and metadata by using *a unified URI policy* and *rollback mechanisms* for corrupted events or files. Moreover it provides *resuming on pending files* (files being uploading), *re-upload of incomplete files* and *auto-switching to update mode* if part-of event has been ingested before a failure happens (i.e. network failure). Thus, users are safe to use or refer to valid (meta) data that are actually hosted in the repository and their related information can be also retrieved or updated anytime through the tool. The ReposIt-Tool supports *type ahead search* at all fields for easy access to the ingested entities/digital objects of the repository. The mechanism is enabled after typing at-least three characters and uses the Lucene[11] engine for optimized queries. Finally, the embedded functionality *of zip file generation* before upload can be useful for very large digital objects. In order to ensure *the referential integrity of Data objects,* the tool does not allow data files that are not already ingested in the repository,

11. http://lucene.apache.org/

to be specified as input files of process events. As a result the user should firstly describe acquisition events, and then the subsequent process events. Fig. 2 presents the two main user interface layouts used for acquisition and process events.

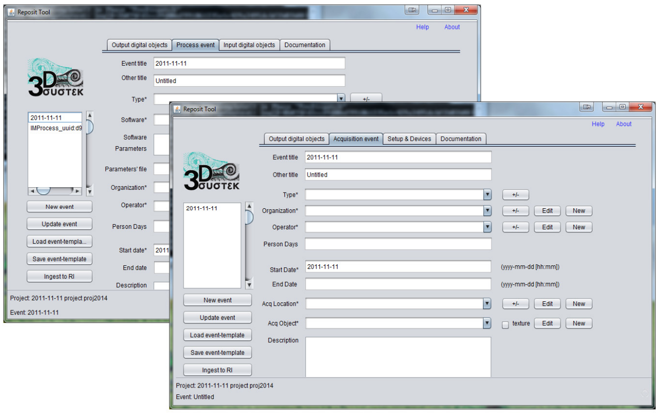

Fig. 2. ReposIt Tool*: Layout of acquisition (front) and process event (back)*

The interface, based on tabbed input forms, assists the user to answer the following questions: (a) **which data** were produced, (b) by **which procedure** (c) using **which input** or **setup**, (d) **what** is the **documentation** information asserted. Thus, in one form the user specifies output data objects located on his/her local storage. In another form the user describes main information about acquisition or process events. For each event the user should specify attributes concerning the event itself: its title, the time and place where the event took place, the organization and the operators/actors involved in the event etc. Specifically, for the description of an acquisition event the user should also specify the devices or software used for the acquisition, some information on the acquisition setup and if necessary device calibration information. For the description of a process event the user should specify as well the software used for processing the input data, the type of the software and the parameters used for the processing. In addition, the system aprovides an optional form to record documentation information including a setup description and attached documentation files.

During the workflow of 3D-model processing users take many steps. In several cases some of these steps and their resulted data-files are reproducible. Users tend to clean these *intermediate data files*, to save space. The system supports this practice by enabling the users to describe these intermediate processes without having to ingest their resulted data files. When describing a process event, the input, except from being a set of already ingested data objects, it may also be the missing output of another process (the user has **erased** output as useless) (see Fig.3, Process B output). In this case the ReposIt tool assists the user in describing the metadata (software and used parameters) for this process event (process B) and its input. The input could be either already ingested data objects or erased output of similar process events. This way the tool persists the *referential integrity* when describing a chain of processes with erased intermediate output data. The metadata generation in the case of intermediate files between multiple processes includes the description of all the processes up to the input files of the first process.

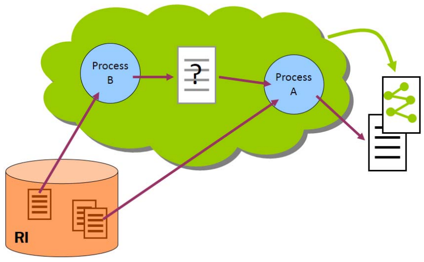

Fig. 3. Workflow of process event and intermediate (missing) files

Furthermore, the ReposIt tool supports the update of events; events can be selected and their description appears in the user interface forms for editing. Thus the user may correct or continue a description that was not completed, or that was interrupted (e.g. due to a connection failure). Sometimes is useful to pause the editing of an event. For this reason, the tool provides the functionality to save and load event to local storage. Saved events may not only be loaded at a later point in time, but may also be used as *templates* when the user has to batch load events with similar characteristics or descriptions. Moreover, the event templates may be used in cases where the process has been assigned to different users (i.e. one user to collect the output, other user to fill in event setup information etc.).

The tool has been implemented in *Java Swing Framework* and can be used as standalone application by simultaneous users. The only requirement for running is an active network in order to establish the connection with the RI Server. The results of recording are presented by the BrowseIt tool.

B. The BrowseIt - Tool

In order to support the dissemination processes and usual searches on CH objects it is necessary to have a tool which can retrieve and display all the related recorded information that resulted from correlated 3D-model production procedures. Moreover annotation between objects enhances the ability to study the provenance and the correlations of each CH object. The BrowseIt tool fills in this gap and can be used in combination with ReposIt tool as clients that communicate with the RI Server.

The BrowseIt tool enables searching, browsing, annotating and downloading of 3D collections within the whole 3D documentation workflow. The searching and browsing functionality is motivated by using fundamental relationships [7] between objects with the combination of free text keywords, as parameters to the final query. In fact by determining general relationships (fundamental) between main classes the user can browse provenance information about objects or related events without having specialized knowledge of the whole Semantic Network. This network is built with 250 kinds of links and is too complicated for a non expert user to express a reasonable query. Therefore, the BrowseIt tool overlays the real network with a simple, intuitive model deduced from the actual model and provides an easy to use interface to support query composition. Queries are forwarded to the QM module of 3DS and results are displayed in the right area side of the tool by enabling the

Browse action (**Browse Query).** For each field of result if there is more information to be retrieved; an active link leads to another tab with a new assigned query (**Entity Query**) that returns the corresponding results. By following the hyperlinks for each entity we can navigate through a *provenance chain of correlated entities* that take part in the production workflow.

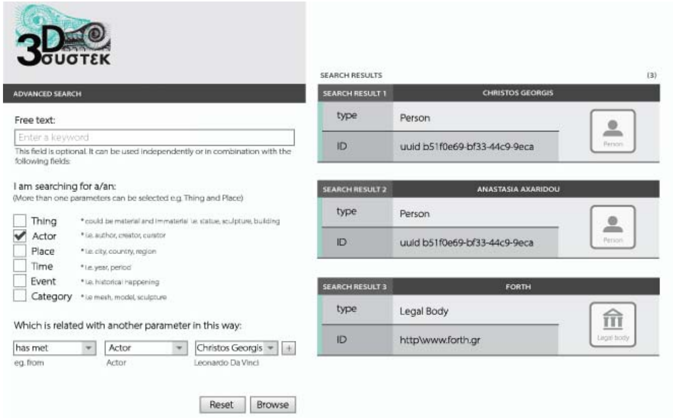

Fig. 4. The main user interface of BrowseIt application

Fig. 4 displays the main user interface of the BrowseIt Tool. On the left we define the search criteria consisting of an optional free-text keyword field and basic classes of concepts that we are searching for: *Thing* (material and immaterial i.e. statue, sculpture, building), *Actor* (i.e. author, creator, curator), *Place* (i.e. city, country, region), *Time* (i.e. year, period), *Event* (i.e. historical happening), *Type* (i.e concept) (mesh, model, sculpture). Also, we can define one or more relationships between concepts to narrow our query (via the "Add" Button). Browse action is started by the "Browse" button. The "reset button" removes all relationships to start from the beginning a different query. The refinement of query is allowed by changing search criteria and click to "Browse" button again.

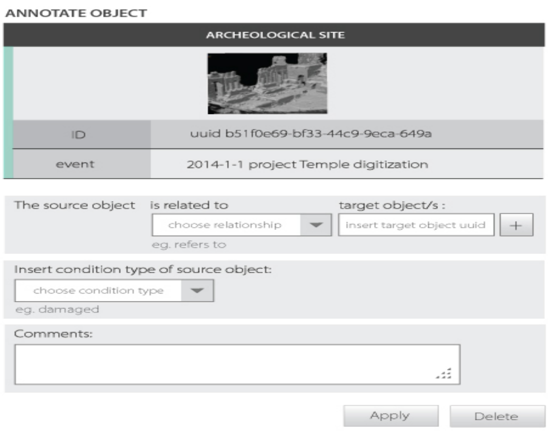

Fig. 5. The annotation interface of BrowseIt application

Digital objects can be at first, previewed in thumbnails. For each digital object a zoom in action is performed on click. Two more actions are supported for digital objects: **download** and **annotate** functionality. The preview functionality "give as a taste" of how the object looks like, while the download functionality give us the ability to further load or process the digital object on another software. In case where the object is start-up or part of other objects the whole set of files are downloaded to the specified folder. With the annotate functionality (see fig. 5) the user can ingest extended information about a digital object such as relation to other objects, physical condition or extra comments. The BrowseIt is a web-based application that runs in the RI Server and it has no other installation dependencies since it uses *HTML5* and *Java Servlets technology* for building the user interface.

VII. CONCLUSIONS

The 3DS and the presented collaborating tools provide an integrated, powerful framework for effective recording of 3D-modelling workflows in Cultural Heritage. Our proposed solution enhances contemporary approaches for CH preservation implementing modern data management techniques. The infrastructure offers extensibility via the provided API, allowing third parties to implement custom solutions for particular CH applications. Moreover, 3DS is based on well established metadata schemes and incorporates Semantics technology to interpret data into knowledge, offering a different perspective of the recorded information for exploitation by all members of the CH scientific community. Last but not least, the 3DS infrastructure complies with the LOD rules and thus can contribute to the richer experience and understanding of CH around the world.

VIII. ACKNOWLEDGMENT

The authors wish to thank the General Secretariat for Research and Technology of Ministry of Education and Religious Affairs, Culture and Sports in Greece for their financial support for the project: "3D-SYSTEK - Development of a novel system for 3D Documentation, Promotion, and Exploitation of Cultural Heritage Monuments via 3D data acquisition, 3D modelling and metadata recording".

REFERENCES

[1] E. Maravelakis, A. Konstantaras, A. Kritsotaki, D. Angelakis and M. Xinogalos, "Analysing User Needs for a Unified 3D Metadata Recording and Exploitation of Cultural Heritage Monuments System" ISVC 2013 vol. 8034, pp 138-147.

[2] M. Schröttner, S. Havemann, M. Theodoridou, M. Doerr, D. W. Fellner, "A Generic Approach for Generating Cultural Heritage Metadata". EuroMed 2012: 231-240.

[3] I. Kollia, V. Tzouvaras, N. Drosopoulos, G. B. Stamou, "A systemic approach for effective semantic access to cultural content". Semantic Web 3(1): 65-83, 2012.

[4] M. Doerr, K. Tzompanaki, M. Theodoridou, C. Georgis, A. Axaridou and S Havemann: "A Repository for 3D Model Production and Interpretation in Culture and Beyond". In Proc. of VAST 2010.

[5] C. Bizer, T. Heath, T. Tom and Berners-Lee, "Linked Data - the story so far". IJSWIS, 5, (3), 1-22, 2009.

[6] M. Doerr, M. Theodoridou, "CRM dig : A generic digital provenance model for scientific observation". In Proc of TAPP 2011.

[7] K. Tzompanaki, M. Doerr, "Fundamental Categories and Relationships for Intuitive querying CIDOC-CRM based repositories" ICS-FORTH Technical Report 429, 2012.