Design, Implementation and Evaluation of a User Generated Content Service for Europeana

Nicola Aloia[1], Cesare Concordia[1], Anne Marie van Gerwen[2], Preben Hansen[3],
Micke Kuwahara[4], Anh Tuan Ly[5], Carlo Meghini[1], Nicolas Spyratos[5],
Tsuyoshi Sugibuchi[5], Yuzuru Tanaka[4], Jitao Yang[5], and Nicola Zeni[1]

[1] Istituto della Scienza e delle Tecnologie della Informazione,
National Research Council, Pisa, Italy
[2] EDL Office, The Hague, The Netherlands
[3] Swedish Institute of Computer Science, Sweden
[4] MEME Media Lab., University of Hokkaido, Sapporo, Japan
[5] Laboratoire de Recherche in Informatique, University of Paris South, Orsay, France

Abstract. The paper presents an overview of the user generated content service that the ASSETS Best Practice Network is designing, implementing and evaluating with the user for Europeana, the European digital library. The service will allow Europeana users to contribute to the contents of the digital library in several different ways, such as uploading simple media objects along with their descriptions, annotating existing objects, or enriching existing descriptions. The user and the system requirements are outlined first, and used to derive the basic principles underlying the service. A conceptual model of the entities required for the realization of the service and a general sketch of the system architecture are also given, and used to illustrate the basic workflow of some important operations. The planning of the user evaluation is finally presented, aimed at validating the service before making it available to the final users.

Keywords*: User Generated Content.

1 Introduction

In the 2011-2015 Strategic Plan, Europeana [1] announces User Engagement to be one of the strategic tracks by which the organization will deliver value. By the term 'Engage' Europeana refers to cultivating new ways for end user to participate in their cultural heritage. The Europeana network comprises communities of archivists, curators and librarians who show a growing interest in exploring new methods of access and dialogue. Europeana intends to enhance the user experience and offer services that allow users to interact and participate.

User-generated-Content (UGC) is one aspect of this renewed way of participating. Information about cultural heritage exists outside the heritage institutions; artifacts, written sources and memories of individuals complement collections held in institutions. UGC services are designed to provide users with means to support and interpret content. They will be involved in storytelling, curating of virtual exhibitions, reviews and even the creation of new collections. Greater participation will increase

users' interest and loyalty. Europeana is therefore devoting increasing resources to initiatives that bring out the value of the contribution those users can make. In response to these needs, the ASSETS [2]. Consortium has included the support of user-generated content amongst the services it is going to develop for Europeana. ASSETS is a two-year Best Practice Network co-funded by the CIP PSP Programme to improve the accessibility and usability of Europeana. Rather than focusing on a specific set of UGC applications, ASSETS is developing a general purpose, back end component that aims at supporting any UGC service Europeana will want to offer to its users. To this end, the ASSETS back end component implements an Application Programming Interface (API) for creating, storing and manipulating UGC Units of Work, and for submitting these Units of Work to Europeana, in the form of Europeana Submission Information Packages (SIPs). Final users will interact with their Units of Work through client interfaces, which will hide the unnecessary technical details and complexities of the back end to them, providing them with the level of representation that is most suitable for the specific UGC task at hand. Indeed, it is expected that every UGC task will be supported by a different final user interface. But this will have no impact on Europeana, since every different front end will talk to Europeana through the same API. The API will relieve future UGC applications from implementing any server side functionality and will move away from Europeana the technical interoperability problems that would arise upon integrating into its database the possibly different objects coming from future UGC applications. The service will rely on the Europeana Data Model (being developed by the Europeana version 1.0 project [3]) in order to tackle the more serious semantic interoperability problems.

The definition of the conceptual model underlying the UGC API is the most difficult challenge that the ASSETS UGC team is facing. The model has to strike the optimal balance between simplicity, so to be quickly learned and easily coded against by the future UGC service developers, and generality, so to satisfy the needs of any possible future UGC service. This conceptual model has been defined during the first year of the ASSETS project, based on an analysis of the different types of requirements that are in place. The model has been subsequently used to define the UGC API.

2   Requirements

User requirements can take different forms: (a) Submission of objects to a repository. A minimal set of metadata will have to be provided in order to support the interpretation, discovery and management of the object. The user requires to be free of choosing which metadata format to use, but the system must propose a default one. (b) Metadata enrichment: Users contribute factual metadata to an object, such as location, date, names, or tags. The object can be created by the user, but also by another user; the object may also be existing content in Europeana. (c) Annotations: users are contributing their views, comments. opinions to an object. (d) Contextualization: through storytelling or creating virtual exhibitions and galleries and possibly adding narratives to them, users are combining existing objects into a new context (without changing the objects and metadata itself). Before publishing user generated content, moderation may be added as

an intermediate step in the process. Authorized users review the UGC and decide to accept and publish it. In most cases, this includes a feedback loop to the user who originally contributed the data.

Europeana audiences include academic researchers with a high level of language and computer skills but also people who are hardly familiar with foreign languages or using the internet. While possessing intermediate to good knowledge of foreign languages and online search, these groups generally expect services to be easy and intuitive. At the same time, they want to understand what happens with their contribution and who keeps control over their content. User Interfaces should therefore preferably be simple, straightforward and visual. Additionally, clear information must be provided about rights regarding the content.

The back end component has to comply with the Europeana architecture, which is based on an Open Source policy. Europeana has also defined a set of guidelines [4] regarding the coding, the testing and the deployment of the components that make up its architecture.

3   The Conceptual Model of the UGC Service

In order to meet the user and system requirements, the ASSETS team designed a UGC service based on the concepts outlined below and presented in Fig. 1 as a UML class diagram.

From the UGC server point of view, at any point in time there exists a set of users of the UGC service. Each user is in fact a role, identified by an id and a password, behind which a whole community may actually operate. Each user has its own Workspace (WS) on the UGC server. A WS is simply a container of the objects that the associated user needs to perform UGC tasks.

The creation of a single UGC object may take a long time and span several sessions of work. In between one of these sessions and the next, the partial results achieved so far have to be persisted, in order not to be lost and to be resumed at the beginning of the next session. The concept of "partial" UGC is captured by the notion of Unit of Work (UoW).   The UoWs of a user are maintained in the user's WS.

A single UoW contains objects, identified by URIs, and their accompanying descriptions. The objects in a UoW can be of two kinds:

- Existing Europeana objects, that the user has included in the UoW in order to link them to new objects (see below) as values of some property, or in order to enrich them with new descriptions. Existing Europeana objects can be retrieved for inclusion in a UoW via a query issued to Europeana.
- Newly created objects, which are called UGC objects. These objects are original contributions to Europeana, and can be of three kinds:
    - o   digital objects having an associated media file with the content;
    - o   digital objects for which no media file is available;
    - o   non-digital objects.

Every object in a UoW has an associated description. A description represents a metadata record of the object and is modelled as a set of attributions, each attribution consisting of a property and a value. Different attributions can have the same property with a different

value. A value can be itself an object, or a literal or another resource, external to the digital library.When a UoW is ready to be submitted to Europeana, the user can do so by using an operation that transforms the UoW into a well-formed Submission Information Package (SIP) and places a message signalling the existence of the SIP into the Outbox. Each user WS is endowed with an Outbox. Europeana retrieves messages from Outboxes in order to harvest the corresponding SIPs. As already mentioned, users can issue queries to Europeana in order to retrieve objects. Each query returns a result, in the form of a message stored in a special area of the user WS called the Inbox. Each user WS is endowed with an Inbox. Messages in the Inbox are of two kinds: query results and notifications that communicate the result of submissions. In case of a negative notification the rejected SIP can be retrieved and re-transformed into a UoW so to allow the user to perform the necessary repairing actions. It is important to notice that these concepts define a general-purpose schema, whose machinery need not be used by every UGC application. For instance, a simple UGC task that takes place in a single session, such as an image upload, may be implemented by directly building the corresponding SIP, so by-passing the UoW stage. On the other hand, another UGC application may decide to publish a finished UoW to a community of users in order to perform a socially oriented form of mediation before submitting the UoW to Europeana. These decisions will be taken by the client side of the applications, relying on appropriate shortcuts offered by the UGC API.
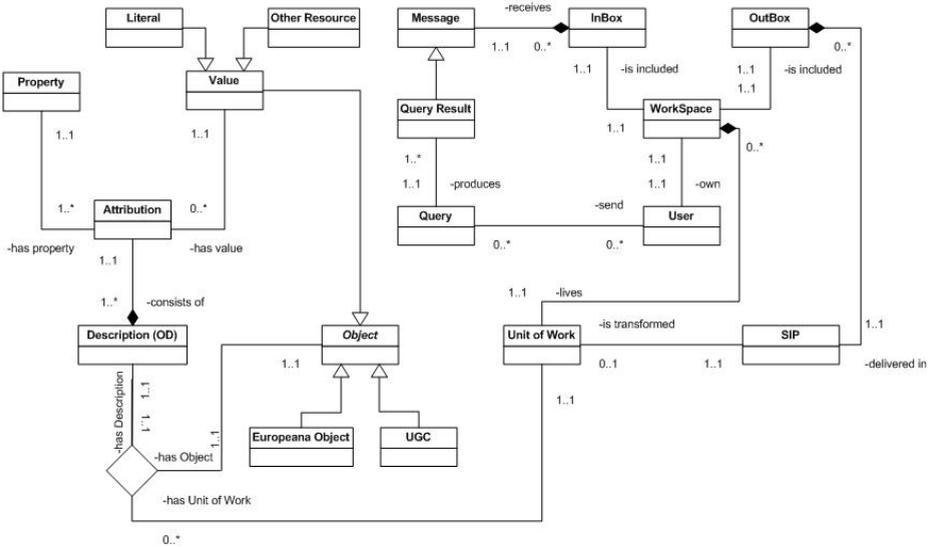

Fig. 1. UGC model description

4   Architecture

For the purposes of developing, testing and evaluating with users the UGC functionality, the UGC server will be deployed on the ASSETS Server. After successful evaluation, the Server will be moved into the Europeana production server.
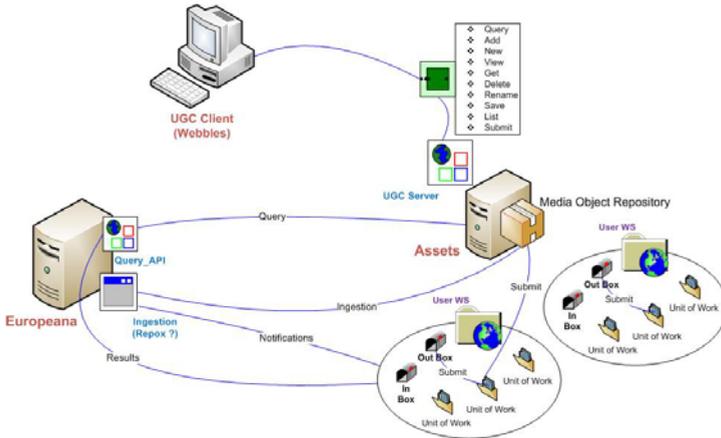
Fig. 2. Overall architecture

The three main components of this architecture (see Fig. 2) are:

Europeana Server. This is the server implementing the Europeana search functionality. It provides an API to search for content in the digital library, it implements the Open Search directives [5]. The Europeana server also provides an application for harvesting of data.

ASSETS UGC Server. In the context of UGC, the ASSETS UGC server provides functionality to communicate with Europeana and with the UGC client. The communication with the Europeana server is by invoking the Query API module (namely OpenSearch API). The ASSETS UGC server manages the workspaces of the users, and provides an API to manipulate the UoWs in the workspace. The UGC server provides API as REST Web services and is independent from any specific UGC client. The User WS contains an Inbox, an Outbox and the set of Units of Work that the user is currently playing with. In addition, the UGC Server maintains the ASSETS Media Object Repository (AMOR), implementing OAI-PMH functionality to allow Europeana to harvest SIPs.

UGC Client. Is a browser-based GUI supporting the user in a specific content generation tasks. The UGC Client interacts with the ASSETS Server via REST web services provided by the UGC Server module.

5   Evaluation

Digital Libraries, Archives and Museums represent the main target for Europeana and through them it will be possible to gain a better understanding of user needs and requirements that will contribute to the design of future services for the users in general as well as for specific groups of users and within specific environments such as the mobile usage and social media applications.

Within the Europeana project, there will be two types of evaluations carried out: a technical and a user-centred evaluation. The technical evaluation will be focused on verifying that the functionalities delivered by ASSETS fulfil efficient and scalable requirements. The user-centred evaluations aim at verifying that the services delivered by ASSETS fulfil the expectations of the end-users. The EDL Foundation together with ASSETS will conduct the evaluation. More specifically, the goal for the evaluation will be a) to utilize a user-oriented approach, and b) to focus on usability aspects of the services proposed and their end-users.

6   Conclusions and Outlook

The main concepts and architectural features of the user-generated content service have been illustrated. The service is being implemented by the ASSETS Best Practice Network and will be evaluated within the lifetime of the project. The basic principles of the evaluation methodology have been described.

The UGC service developed by ASSETS is based on a general-purpose back end, which is meant to relieve Europeana from dealing with the specificities of the possibly very many UGC tasks that may be offered to users. At the same time, the back end relieves developers of UGC tasks from implementing the server side of their applications.

References

1. The Europeana portal, `http://www.europeana.eu`
2. The ASSETS project, `http://www.assets4europeana.eu`
3. The Europeana version 1.0 project, `http://version1.europeana.eu`
4. The Europeana Labs, `http://europeanalabs.eu/wiki/`
5. The Open Search, `http://www.opensearch.org/Home`