Acquiring Digital Asset Management System for an International Project Consortium

Introduction

In the era of multimedia, digital asset management becomes a necessity for institutions with rich content repositories, if the content is to be used, not merely kept. Sometimes, it is involvement in a single project that makes institutions, which were free of this problem, to accumulate enough multimedia resources to start looking for a digital asset management solution. We address such situation in this paper.

Digital Asset Management Systems (DAMS) are a software solution to this problem. Although instances of DAMS are not as stunningly numerous as that of Web Content Management Systems, there is still enough of them to consider adopting an existing solution instead of developing a new one. It is certainly reasonable to choose the best of what is available. An assessment and ranking procedure is therefore inevitable. Moreover, even the best available solution may lack some of the features considered vital in the specific usage context. In that case, some degree of system adaptation is necessary.

In this paper we would like to show how the procedure of acquiring DAMS – in this case, for an international project consortium – could be
organized. The solution we propose is the one verified in practice in the Baltic Museums 2.0 Plus project.

The paper was written with the financial support from the European Union.

**1. Goals of the BalticMuseums 2.0 Plus project**

BalticMuseums 2.0 Plus project titled “Implementation of eGuides with cross-border shared content for South Baltic Oceanographic Museums” is realized within the South Baltic Cross-border Co-operation Programme 2007-2013 and is part-financed from the European Regional Development Fund under the European Territorial Co-operation objective of the cohesion policy [Operational Programme..., 2007]. Its realization is scheduled for three years from December 2010 till November 2013. The total budget of the project exceeds the sum of one million euro, whereat refinancing from the European Regional Development Fund amounts to 85% [BalticMuseums 2.0 Plus Fact Sheet, 2010].

BalticMuseums 2.0 Plus project is realized by an international consortium consisting of two scientific institutions – the University of Applied Sciences in Stralsund and the University of Szczecin, and four oceanographic museums – The German Oceanographic Museum in Stralsund, Gdynia Aquarium, Lithuanian Sea Museum in Klaipeda and the Museum of the World Ocean in Kaliningrad. This project is an upgrade to the BalticMuseums 2.0 project titled “Joint development of cross-border information products for South Baltic Oceanographic Museums”, which was realized in the years 2009-2012. The general objective of both projects is the promotion and effective use of the natural heritage stored in the oceanographic museums by means of cross-border tourism information tools, in order to increase their attractiveness and competitiveness, especially for international tourists.

The main objective of the BalticMuseums 2.0 Plus project is to develop multilingual content describing exhibits of the museums – partners of the project, and make it available for the tourists via multimedia e-guides. An equally important issue is the assumption of content sharing among museum partners to achieve durable cooperation with a long term perspective [BalticMuseums Website, 2013]. Therefore one of the primary goals of the BalticMuseums 2.0 Plus project is the
implementation of an effective content sharing system for storing and sharing content of various formats.

2. Digital Asset Management Systems

Digital Asset Management Systems are a kind of Content Management Systems (CMS). One of the available definitions presents digital asset management system as “a set of processes that when working together give a system, repository, and enabling workflow process for managing publishable media content such as images, illustrations, documents, audio, video and physical (non-digital) elements” [Sawarkar, 2001, p. 5]. From this point of view, it is important to understand what the content actually is and how we can use it.

Bob Boiko in his seminal work [Boiko, 2005, p. 8-11], defines “content” with the following statements:

- content is not [only] data (or file),
- content is information put to use,
- content is information plus data.

Note that the same author defines information products as “results of the transformation of knowledge into information and (...) an integral blend of content and container”. From the user’s point of view, content is every element that we seek and want to transform into knowledge (see [Orna, 2004, p. 7]).

Thus, we can define DAMS as a tool supporting users in the ingestion, annotation, cataloguing, storage, retrieval and distribution of digital content. The usual functionalities of DAMS include integration with authoring tools, image manipulation and transformation, on-the-fly image and video transcoding, metadata editing and search. A list of the most desirable DAMS features has been proposed by Zielinski and Lorenz; it consists of [Zielinski, Lorenz, 2012]:

- metadata definition,
- file types supporting,
- streaming video,
- structured authoring,
- personalized dashboard,
- customized permissioning,
- workflow management,
- microsites/portals,
- RSS feed generation,
– mobile accessing,
– modular/scalable,
– Web 2.0,
– multilanguage support.

As much as Document Management Systems (DMS) are focused on textual documents, DAMS are focused on multimedia content. As such they can be viewed as a subset of Enterprise Content Management (ECM) systems (cf. [Sawarkar, 2001, p. 7]).

There are specialized types of DAMS, such as (see [van Til et al., 2010, p. 53, 56]):
1. MAMS (Media Asset Management Systems) – for storing and retrieval of media assets, for example in video or photo archiving.
2. PAMS (Production Asset Management Systems) – for managing digital assets as they are created during digital media production (video game, feature film, animation, visual-effects shots, etc.).

In the remainder of this paper we shall describe the acquisition of a digital asset management system for the international consortium implementing the BalticMuseums 2.0 Plus project. The main aim of the system is to enable effective content sharing among the participants of the project [Swacha, 2011]. The system cannot be uniquely classified as MAMS or PAMS, as it is used first as PAMS during the development of eGuides, then as MAMS for preserving the content for the purpose of future use.

3. Procedure of acquiring DAMS

It has been decided on the strategic level of the project management that instead of implementing a proprietary solution, an open-source software system will be sought instead.

Expecting that no available DAMS will exactly match the requirements of the BalticMuseums 2.0 Plus project consortium, and some degree of adaptation will be necessary, the FEChADO methodology has been adopted [Swacha et al., 2012].

FEChADO is a development process framework that can be used in various organizations for adapting open source software. Its name is an acronym of the six steps it consists of:
1. Find available solutions.
2. Evaluate solutions from the list.
3. Choose the most appropriate solution.
4. Adapt the solution.
5. Develop new modules.
6. Obtain users’ feedback.

As FEChADO is a flexible high-level framework, it does not enforce a specific method for any of its stages. In the following sections of the paper we shall describe how they were realized during acquiring a digital asset management system for the BalticMuseums 2.0 Plus project. We only skip stage 5 as the applied modifications were too few and too limited to warrant development of a new system module.

4. Selecting system using CMSmatrix

An initial idea of the development team was to use CMSmatrix [Plain Black, 2012] to complete steps one and two of the FEChADO procedure. It is a web-based tool allowing to compare over 1200 content management systems, using criteria grouped in ten categories: (1) system requirements, (2) security, (3) support, (4) ease of use, (5) performance, (6) management, (7) interoperability, (8) flexibility, (9) built-in applications, (10) commerce.

These ten categories contain a total of 145 criteria, mostly in form of features which are assessed for each system using one of the following statements:
- “No” (feature is unavailable),
- “Limited” (feature is available, but only in a limited form),
- “Costs extra” (feature is available for an additional fee),
- “Free Add On” (feature is available as a free add-on),
- “Yes” (feature is built-in).

As the requirements for the BalticMuseums eGuide content sharing system had been specified using different terms, it was therefore necessary to prepare a mapping of the requirements to the set of criteria of the CMSmatrix tool. The procedure and results of this mapping are described in another paper [Komorowski, 2011, p. 61-73].

Using the CMSmatrix tool with mapped set of criteria resulted in a list of ten content management systems. A closer look at these systems revealed that in reality none of them is suitable for the intended use without significant modifications.

The main reasons for this were: lack of relevant criteria in the CMSmatrix set, criteria listed there being too general (e.g., “multimedia management”), different meaning of criteria in the CMSmatrix than in the system requirements specifications. Therefore, even the sophisticated
mapping used here failed to produce useful effects: it did map respective requirements to their most equivalent CMSmatrix criteria, but the analogy turned out to be too weak. It does not mean that CMSmatrix should not be used to support CMS selection; it means, however, that it should not be used to support selection of DAMS, as the set of criteria the tool uses does not truly represent the features specific to digital asset management.

5. Selecting system using Hasse diagram

The failure of using CMSmatrix made the development team resort to find available DAMS using web search engines. As a result, seven candidate systems were proposed. They are listed in Table 1.

Table 1. Systems selected for assessment

<table>
<thead>
<tr>
<th>ID</th>
<th>Name of System</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>OpenKM</td>
<td>5.0.3</td>
</tr>
<tr>
<td>N2</td>
<td>Cynapse/Plone</td>
<td>3.1.3</td>
</tr>
<tr>
<td>N3</td>
<td>TYPO3 DAM</td>
<td>4.6</td>
</tr>
<tr>
<td>N4</td>
<td>ResourceSpace</td>
<td>4.3.2</td>
</tr>
<tr>
<td>N5</td>
<td>Alfresco CE</td>
<td>4.0</td>
</tr>
<tr>
<td>N6</td>
<td>NotreDAM</td>
<td>1.08</td>
</tr>
<tr>
<td>N7</td>
<td>EnterMedia</td>
<td>7.5</td>
</tr>
</tbody>
</table>

Source: Own elaboration.

The two consequent steps of the FEChADO framework are to evaluate candidate solutions and choose the best among them.

There are various methods for assessment of alternative solutions. The purpose of these methods is to arrange a set of candidates according to an adopted classification rule and to distinguish a possibly smallest subset, which will form the basis for making further choices. The most commonly used among them are multi-criteria methods such as Hasse diagram [Brueggemann, Halfon, 1995], outranking methods (e.g., ELECTRE) [Roy 1991], and hierarchy processes (e.g., AHP) [Saaty, 1980]. It was decided that the first of the mentioned methods will be used, mainly because of its simplicity.

Hasse diagram is an oriented graph $G=(W,R)$, where $W$ means a set of compared elements, while $R$ is a partial order relation, $R \subset W \times W$, which can be written as $(x, y) \in R \iff y$ is better than $x$. In this case "better"
means that the sum of the values of all criteria for element $y$ is greater than for element $x$, which can be expressed by the formula:

$$\sum_{i=1}^{N} K_i(y) > \sum_{i=1}^{N} K_i(x)$$  \hspace{1cm} (1)$$
or using a weighted sum:

$$\sum_{i=1}^{N} w_i \cdot K_i(y) > \sum_{i=1}^{N} w_i \cdot K_i(x)$$  \hspace{1cm} (2)$$

where $w_i$ – weight assigned to the $i$-th criterion, $\sum_{i=1}^{N} w_i = 1$, $w_i \in [0,1]$, $i = 1, N$. Such summing of criteria leads to creation of one metacriterion from many partial criteria.

Each candidate system has been assessed in accordance with the Hasse diagram procedure. All relevant criteria were aggregated into six primary criteria and their weights where specified. The highest weight was assigned to the criterion of basic system features and the lowest weight was assigned to the non-functional requirements (see Table 2).

Table 2. Aggregated evaluation criteria of candidate systems

<table>
<thead>
<tr>
<th>ID</th>
<th>Criterion</th>
<th>Short</th>
<th>Preference direction</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>Basic System Features</td>
<td>BSF</td>
<td>max</td>
<td>0,3</td>
</tr>
<tr>
<td>K2</td>
<td>Content Editing</td>
<td>CE</td>
<td>max</td>
<td>0,2</td>
</tr>
<tr>
<td>K3</td>
<td>User-Interface Related</td>
<td>UIR</td>
<td>max</td>
<td>0,1</td>
</tr>
<tr>
<td>K4</td>
<td>Non-Functional Requirements</td>
<td>NFR</td>
<td>max</td>
<td>0,05</td>
</tr>
<tr>
<td>K5</td>
<td>Hosting Technologies &amp; Requirements</td>
<td>HTR</td>
<td>min</td>
<td>0,1</td>
</tr>
<tr>
<td>K6</td>
<td>Workflow</td>
<td>W</td>
<td>max</td>
<td>0,25</td>
</tr>
</tbody>
</table>

Source: Own elaboration.

As the criteria for systems evaluation had differentiated point scales for each of them, so it was necessary to calculate the normalized weighted ratings. The following formula has been used for normalization:

$$K_i(x) = \begin{cases} 
  1 - \frac{K_i^{max} - K_i(x)}{K_i^{max} - K_i^{min}}, & \text{if } K_i \text{ was maximized} \\
  \frac{K_i^{max} - K_i(x)}{K_i^{max} - K_i^{min}}, & \text{if } K_i \text{ was minimized} 
\end{cases}$$  \hspace{1cm} (3)$$

Table 3 shows the ranking of the candidate systems according to the normalized weighted ratings.
Table 3. Systems ranked by normalized weighted rating

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Rating</th>
<th>Weighted Rating</th>
<th>Normalized Weighted Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>N4</td>
<td>29.50</td>
<td>9.95</td>
<td>0.83</td>
</tr>
<tr>
<td>N5</td>
<td>23.50</td>
<td>8.00</td>
<td>0.81</td>
</tr>
<tr>
<td>N7</td>
<td>26.50</td>
<td>9.20</td>
<td>0.80</td>
</tr>
<tr>
<td>N3</td>
<td>32.00</td>
<td>8.45</td>
<td>0.72</td>
</tr>
<tr>
<td>N1</td>
<td>32.00</td>
<td>9.55</td>
<td>0.55</td>
</tr>
<tr>
<td>N2</td>
<td>8.00</td>
<td>5.40</td>
<td>0.49</td>
</tr>
<tr>
<td>N6</td>
<td>2.00</td>
<td>4.60</td>
<td>0.24</td>
</tr>
</tbody>
</table>

Source: Own elaboration.

The obtained results indicate solution N4, that is ResourceSpace system, version 4.3.2. The Hasse diagram for the normalized criteria is shown in the Figure 1.

Figure 1. Hasse diagram for candidate system ranking

Interpreting the Hasse diagram, there is no better solution than N4 (ResourceSpace). It dominates N7, N1, N5, and N3 in every criterion, whereas N6 and N2 in the normalized weighted rating.

6. Adapting DAMS

There were two major issues regarding differences between the specification of the BalticMuseums 2.0 Plus eGuide content sharing
system and ResourceSpace capabilities. Two of them were solved by proper configuration of the DAMS without applying any modification to its source code; the third one required such modification.

The main problem was that the specification stated that the content was organized hierarchically, depending on its purpose and the institution providing it (i.e., one of the project partners) [Swacha, 2011]. Unfortunately, ResourceSpace lacks the content hierarchy, as it uses a flat space for content storage. Although there are tags that let the content be filtered in search results, but this does not truly mimic the hierarchical organization of the content.

In order to enable hierarchic organization of content, the theme feature of ResourceSpace was used. Themes were defined for individual institutions participating in the project (the four museums and two universities), and the purpose of keeping the content (documenting project activities or using in eGuide).

The second problem was with limiting user access rights only to the content created in their institution or marked as “shared”. This has been attained using the search filter feature of ResourceSpace, which allows to hide content lacking specified metadata from the specific user class.

The third problem was with limiting users’ delete rights. According to the system specification, it should be possible to grant resource metadata edit rights to specific user class, at the same time forbidding them to delete such resource. ResourceSpace allows no such fine-tuning of access rights, and the only possible workaround was to modify the source code, so that resources could only be deleted by their creators or administrators. There is a risk that if the relevant fragment of code is significantly changed in a future version of ResourceSpace, the fix will have to be reimplemented. In case of minor changes, the fix just has to be reapplied according to a specified maintenance procedure.

7. Users’ feedback

Contrary to the development team’s expectations, ResourceSpace turned out to be an unintuitive system, in practice, impossible to use by the end-users without proper manual. And the first distributed version of the manual prepared by the development team was considered too technical and too extensive. As a result, only few end-users managed to perform their tasks within the system, and even fewer did it according to
the rules established to ensure high quality of the repository and keep it manageable in the long term.

Consequently, a lot of complaints were registered, most of them were due to system misuse. In order to overcome this problem, an additional training was carried out, and a shorter, more comprehensible version of the manual was prepared and distributed.

There was, however, one important technical issue reported. Even though unprivileged users were unable to protect other users’ content (after the fix described in section 6), they were still able to remove it from collections they were not owners of. The difference is, the removed content remains in the repository, but is no longer listed in a specific collection.

As ResourceSpace does not differentiate access rights for adding content to a collection (desirable feature) and removing content from a collection (undesirable feature), the development team found no way of fixing this issue by means of system reconfiguration and resorted to modification of the system source code. Various ways of removing content from collection were blocked on the level of the user interface. As in case of the fix described in section 6, there is a risk that if the user-interface-related code of a future ResourceSpace version is significantly changed, the fix will have to be reimplemented. In case of minor changes, the fix just has to be reapplied according to a specified maintenance procedure.

Conclusion

The procedure of acquiring a digital asset management system described here is based on the FEChADO framework, whereas the critical system selection step uses Hasse diagram.

The procedure allowed the Baltic Museums 2.0 Plus international project consortium to implement, in a short time, a system for sharing eGuide content. These successful results suggest that the procedure could be used as a model to be followed by any institution or project consortium that needs a convenient acquisition of a digital asset management system.

References


**Abbreviations**

CMS – Content Management Systems
DAMS – Digital Asset Management Systems
DMS – Document Management Systems
Acquiring Digital Asset Management System for an International Project Consortium (Summary)

The paper describes a procedure of acquiring DAMS developed within the Baltic Museums 2.0 Plus project. The procedure is based on the FEChADO development process framework for adapting open source software, and uses Hasse diagram for system selection. The procedure can be implemented by any institution or project consortium for DAMS acquisition.

Keywords
DAMS, system selection, FEChADO framework, Hasse diagram