Content Management in the Context of Collaboration

Marius Ioan Podean, Ştefan Ioan Niţchi, Dan Benţa
Business Information Systems Department
Babes-Bolyai University, Faculty of Economics and Business Administration
Cluj-Napoca, Romania
Emails: {marius.podean, stefan.nitchi, dan.benta}@econ.ubbcluj.ro

***Abstract*—Collaboration is a very dynamic process that combines functionality that supports communication, management and involves content handling. During the execution of a project team members are not always collaborating and their work alternates with cooperation, when a greater emphasis is placed on a value-chain model of producing results. Focus permanently switches from the flexible content approach to the management tools according to task's specific. Content management is often regarded in term of web or enterprise content, or as document management. Documents are mere representations of content following a particular format and this research is focused on describing a method of providing flexible interaction with content objects in the context of collaboration. Collaboration requires a flexible content management solution in order to support the interaction between group members.**

***Keywords*-collaboration, flexibility, content adaptability, creative interaction, XML**

I. INTRODUCTION

Collaboration builds on communication, coordination, cooperation (the 3C model [1]) but requires some extra components also. Coordination represents the management of people and their activities and is based on altering activities for mutual benefit [2]. Cooperation goes one step further and adds resource sharing in order to achieve a shared vision [3]. Complex problems require aspects of knowledge that reside in the minds of individual stakeholders as tacit knowledge [4]. Reaching a resolution based on consensus building produces a higher-quality decision than other decision-making processes [5]. Collaboration demands a flexible content management solution that can support the managerial efforts in order to stimulate creative insight. Such a content management solution should allow groups of minds to interact with each other and allow content to be used as an externalization of their thinking [4]. Making interesting connections between content elements more evident and supporting the growth of an idea are just a few of the key area that such a system could enable. Starting from these challenges we will present a model for a content management component that handles content in a flexible manner in order to allow users to focus their action on the task at hand an not on handling technology. Our approach is based on exploiting the flexibility of XML and related tools in regard to content management. Constraints imposed by collaboration are modeled following XML principles. For this purpose, a model that describes our findings is presented together with some implementation details.

Usually content management is considered as an individual tools and not as an integrated component of collaboration. This approach shifts the focus from "intelligent content"[6] to web content or document management, leading thus to tools that are used in conjunction with a groupware solution that do not support the needs for collaboration. Our focus is on developing a content management module that could be integrated with other tools that are required by collaboration, in such a manner that is in accordance with the prerequisites of collaboration.

This paper is organized as follows. In the second section we will discuss the main characteristics that we consider that are relevant for a content management component in a collaborative system and present a model that covers this aspects. Following this discussion, in section three we will present in greater detail the main functionality of our model. The final section will present some concluding remarks and future work.

II. CONTENT MANAGEMENT MODEL

A flexible content management solution that must satisfy the requirements of collaboration must take in consideration aspects like handling the entire life cycle of content [7], provide creative means of interaction with the content so that content objects can be used as externalizations of human thinking [4] and serve as a basis for negotiation and critique. The use of structured content [8] standards and open formats together with metadata [9] will allow for more flexibility to be implemented in the system enabling thus a wider range of actions that can be performed on content . The separation of content from presentation [10] and implementing a great adaptability of the content together with a single source / multiple publishing channel and formats focus enable the system to customize content to user needs. Including project and content workflow management and providing a good accessibility and adaptability will enable user to tailor the life cycle of the content according to organizational requirements.

Our model focuses on collaboration, all elements being tailored around major concerns for this process. Most systems focus on web content, document management, pub-

 ISBN: 978-1-61208-157-1

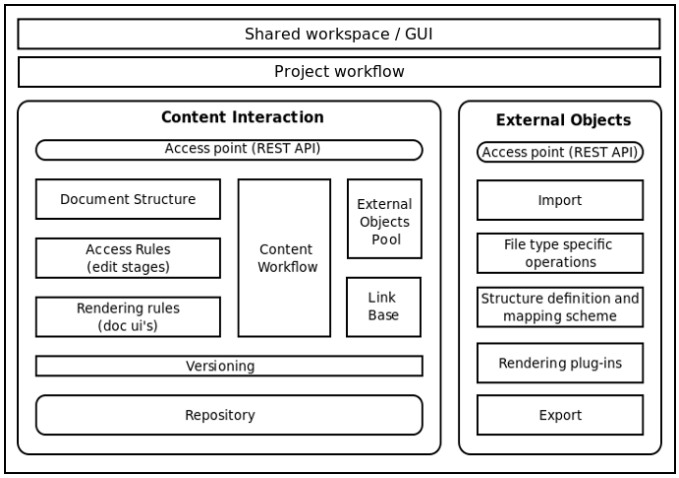

Figure 1. Collaboration enabler content management model

lishing firms and none on the needs related to collaboration. Based on the gaps identified we have defined a model for content management that focuses on four key areas: content interaction, external object management, project workflow and credential management. The principles behind the stated model will allow it to provide a great support for content management in a collaborative system.

III. Main components

The novelty of our research is that all the following components have implementations using mainly XML tools (e.g., content level workflow is handled in XProc, a pipeline-based XML processing tool). This approach enables us to provide a rich set of content interaction mechanisms required by collaboration. In the following we will stress on the design aspects of our content management model.

A. Content interaction

The main purpose of this component is to handle all interaction with the content, from it's creation to publishing. It includes the repository also and the versioning mechanism. The content interaction is separated in defining content rules (definition schemes, edit and rendering rules) and metadata, external object and link handling.

In our approach the term "document" represents a very flexible concept being regarded as a temporary view for the content. Most often refers to web representations but it can include common formats like PDF. A document is constructed from three main components: a definition, a set of editing rules and a rendering definition. A content definition consists in a vocabulary - XML Schema (schema), that defines the structure of the documents, the elements that are allowed in this structure, element data types and restriction regarding occurrences.

The access rules define sets of editing constraints that apply to roles at different editing stages. A separation in edit stages is made because during content's life cycle different sets of actions can be taken that have implications on the quality of the content. The rendering rules define visualizations for the content. These visualizations range from web user interfaces to printable documents. For each publishing format (web, print) the user must define a rendering definition. This definition must describe the relevant elements for display in the targeted media. For example an XSL-FO definition can be used for both formats but can be split and specialized for each of them. These definitions specify how content is going to look in a certain format covering aspects like page formating, styles, specific positioning in the page for elements etc. The way an element is displayed following the rendering rules is in strict relationship with the access rules. The access rules are enforced by the use of user interface elements: if an element is editable it will be displayed as an input field that allows data to be filled in, otherwise as a text element.

A request to view a document implies assembling all aforementioned elements in order to provide a representation of the content. When a representation is requested its definitions are loaded and the structure will be filtered by eliminating elements that are not accessible for the user. For this, the access rules are applied to the document definition marking as hidden elements that the user is not allowed to see. Further, the access rules are applied to the rendering definition marking elements that are editable or read only. This will result in a user interface definition that takes in consideration access rules and a schema that can be used to filter content that should not be accessible to the user.

After a document view is generated in a web format, if the user's credentials or edit stage allow it, the user can edit the content. When a document is edited individual changes are stored separately on the client (as Δ consisting in each edit step applied to content) and only these changes are sent to the server and patched on the version that they refer to (in case no other update has been made on the content). There are two ways to work on a content representation:

1) asynchronous: when the user's Internet connection does not represent a problem, each change is sent individually to the server. In the same respect, a pull operation is made at a predefined interval in order to retrieve changes made by others.
2) synchronous: multiple edit operations are stored on the client and sent to server only when the user decides to save them. This approach may lead to edit conflicts if other users have edited the same content objects.

The metadata is an important component of each content object (Figure 2). The **Repository** acts as an file system that stores all content in XML and manages content, data about content and how are they organized. Content is organized in namespaces and projects. A namespace is the higher level (e.g. it can refer to a department) that can have multiple

 ISBN: 978-1-61208-157-1

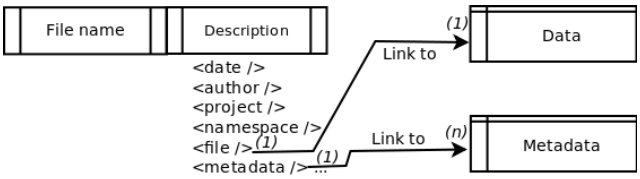

Figure 2. The structure of a content object

projects. Projects have content objects and roles associated with them. A user has access only to those projects (and inside the projects, to the files) that it has the required authorization to, according to the active role.

B. External objects

External objects are the gateway to introduce non standardized content in the system. This module handles *i)* the import of content from other formats, *ii)* the definition and execution of file type specific operations, *iii)* conversion to other vocabularies (where possible), *iv)* hosting and execution of special rendering plug-ins, and *v)* content export back to original format or other (closed-source) formats.

When an import request is received by the **External objects** module, based on the information regarding file type, a format plug-in is called and the import process is triggered. In the first phase the content is converted to XML following a structure defined by the format plug-in. If the conversion is successful the new content object is registered in the Object Pool from the **Content Interaction** module as raw content. The raw content will be stored in the **Repository** following the structure depicted in Figure 2.

If the format plug-in defines special operations that can be applied to the content, they will be presented to the user and then the content will be edited following the steps defined by the selected operation. For example, for a large Microsoft Excel file one useful specific operation could be a dataset cleanup that will search for similarities in string columns in order to identify typos. This operation will help increase the quality of data with minimal effort. After following the steps included in the file type specific operation, the content object is updated in the Object Pool. The next step consists in showing the user the available formats that the content can be converted to, and if the user requests the content to be converted to some format, following a mapping schema the content is transformed. The last step consists in associating the content with the active project and metadata that describes its characteristics. Applying file type specific operations and converting to other vocabularies are optional steps and can be called any time during the life cycle of the content.

All external objects plug-ins must be defined as web-services and provide an API. The module require users to provide the API and URL of these web-services in order to integrate them in the system. The plug-ins are not

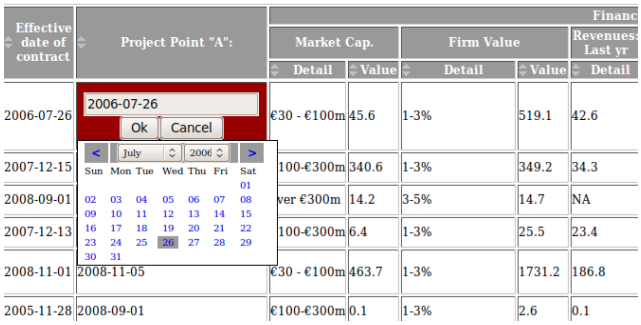

Figure 3. External object representation: tabular data editor

necessary stored on the same server as the framework, but the application should provide a sandbox so that users can install and test their own plug-ins.

Since the content is converted to a user specified format, the possibility to transform it to a standard vocabulary should be implemented by allowing users to include transformation schemes. All schemes defined for a specific file type are managed by mapping mechanism that is responsible also with applying these transformations when requested.

The content that can be included using this mechanism is very divers, thus a mechanism to provide the proper means to display the content and interact with it is an essential requirement. Rendering mechanisms are included similar to file type specific operations, as web services. The document representation that includes an external object is similar to web mash-ups since it aggregates content and content representations from multiple sources. In the initial document only a description of the external object is present and according to the required delivery channel a proper representation is returned. If the document is delivered as a web document, then the appropriate web-service is called in order to provide a user interface that will allow users to interact with the content. For example, an Excel file that has been imported can have a user interface similar to the original editor that will allow some basic operations to be executed on the content (Figure 3). In the same manner, a definition of a genome can be provided in a 2D or 3D representation in a web document.

If the delivery channel is print media or similar, a different representation of the external object or a snapshot can be included. Delivering document representations as mash-ups enables the framework to handle a large number of content formats and providing thus a great flexibility in customizing it to user needs. External objects can be further exported back to their original format or other appropriate formats through the use of export plug-ins.

C. Project workflow

The workflows involved in the model are separated on two levels: worflows that concern content object processes and workflows that integrate project processes and content

 ISBN: 978-1-61208-157-1

objects in order to support collaborative endeavor. This module must include an workflow engine and a process map generator. The process editor is an optional component and can even be implemented separately.

After defining the project management plan, in a semi-automatic way a workflow schema must be generated by one of the subcomponents of the workflow engine. From the project management plan data regarding the tasks, their sequencing in time, dependence between them, time constraints can be extracted an imported in the workflow schema. This process will result is a partial workflow schema that must be customized by adding connections to the documents involved by each step, defining variables for decisional steps, select content representations for each edit stage of a document and attach access rules. Coming back to our article example, in the idea gathering step users are allowed to create arguments using the argument map rendering. After a certain amount of time the task will end and the next process will involve using and document like article editor that will allow users to edit the content of an argument but not allow them to remove arguments. Another use case might be a large project that includes reporting stages that require users to fill-in certain document in order to track work progress. When a report is created the owners can edit it but soon after the report has been approved they must no longer have access to this functionality.

After a workflow schema has been defined, a process map can be generated in order to provide users the information they need regarding the execution of the project. A process map is a simplified form of a workflow schema that does not contain all elements required to be executed by a worflow engine but incorporated enough details to keep users informed about the steps required in order to reach the goals and more important, the positioning in time of the project's execution and deadlines for content objects. This is a simplified form of project and risk management plan and a workflow schema.

As mentioned earlier, the workflow editor is an optional component, but the module should include a minimal customization tool that will allow users to edit the semi-automatic generated workflow schema in order to associate content objects and access rules. A workflow editor would allow user to fully customize a schema (or create one from the beginning) by adding or removing steps. Since the workflow schema must be derived from the project management plan such functionality is not compulsory.

IV. Conclusion

During the execution of a project team members are not always collaborating and their work alternates with cooperation, when a greater emphasis is placed on a value-chain model of producing results. Focus permanently switches from the flexible content approach to the management tools according to task's specific. In order to fully support the process of collaboration, the aspects that precede it or come in-between the collaboration sessions must be fully supported so that they will not represent a problem that can hinder collaboration. Both collaboration and cooperation require content support and management tools, what they differ in is their main focus.

A flexible content management solution that must satisfy the requirements of collaboration must take in consideration aspects like *i)* handling the entire life cycle of content, *ii)* provide creative means of interaction with the content so that content objects can be used as externalizations of human thinking and serve as a basis for negotiation and critique. The proposed framework takes in consideration all the aforementioned aspects and implements a very flexible approach that is targeting customization according to domain specific needs. We have designed our model starting from the idea that content needs are very divers and so rapidly changing that a content management module should not try to offer a holistic approach that will accommodate all needs, but instead provide a general framework that will allow users to define and customize both the processes involved in their projects and the content objects that are used to harness team's efforts. The actual model is the result of continuous refactoring starting from prototypes and since the implementation of the model is in an early stage we intend to finalize it in order to test the validity of the model by using it in the current projects running in our university.

Acknowledgment

This work was supported by Romanian National Authority for Scientific Research under the grant no. PN2 92-100/2008 SICOMAP.

References

[1] L. M. Camarinha-Matos and H. Afsarmanesh, *Collaborative Networks: Reference Modeling*. Springer US, 2008, ch. Collaboration forms.

[2] H. Fuks, A. Raposo, M. A. Gerosa, M. Pimental, and C. J. P. Lucena, *Encyclopedia of E-collaboration*. Hershey, PA: Information Science Reference - Imprint of: IGI Publishing, 2008, ch. The 3C Collaboration Model, pp. 637–644.

[3] T. Wolff, "Collaborative Solutions - True Collaboration as the Most Productive Form of Exchange," Collaborative Solutions Newsletter, 2005, Last Accessed: October 2010. [Online]. Available: http://www.tomwolff.com/collaborative-solutions-newsletter-summer-05.htm

[4] E. Arias, H. Eden, G. Fischer, A. Gorman, and E. Scharff, "Transcending the individual human mind - creating shared understanding through collaborative design," *ACM Trans. Comput.-Hum. Interact.*, vol. 7, no. 1, pp. 84–113, March 2000.

[5] D. Straus, *How to Make Collaboration Work: Powerful Ways to Build Consensus, Solve Problems, and Make Decisions*, ser. 107-8. San Francisco: Berrett-Koehler Publishers, 2002.

 ISBN: 978-1-61208-157-1

[6] A. Rockley, "What is Intelligent Content?" http://thecontentwrangler.com/2011/01/17/what-is-intelligent-content/, 2011, Last Accessed: February 2011. [Online]. Available: http://thecontentwrangler.com/2011/01/17/what-is-intelligent-content/

[7] U. Bartlang, *Architecture and Methods for Flexible Content Management in Peer-to-Peer Systems*, 1st ed., A. W. Ute Wrasmann, Ed. Wiesbaden, Germany: Vieweg+Teubner, 2010.

[8] A. Salminen, "Building Digital Government by XML," in *HICSS*, ser. 38th Hawaii International Conference on System Sciences (HICSS-38 2005), CD-ROM / Abstracts Proceedings, 3-6 January 2005, Big Island, HI, USA. IEEE Computer Society, 2005.

[9] T. R. Group, "The Role of Content Standards in Content Management," The Rockley Group Inc., Markham, White Paper, 2005.

[10] D. Clark, "Content Management and the Separation of Presentation and Content," *Technical Communication Quarterly*, vol. 17, no. 1, pp. 35–60, 2008.

 ISBN: 978-1-61208-157-1