Babes-Bolyai University
Faculty of Economics and Business Administration
Business Information Systems Department

Content Management in Collaborative Systems

Thesis Summary

Scientific Advisor:
PhD. Ştefan Ioan Niţchi, Professor

PhD. Student:
Marius Ioan Podean

September, 2011

Abstract

Creative insight is the justification for all collaborative efforts and in order to generate insight, new ideas or new artifacts it is a necessity to bring together different and often controversial points of view. Externalizations create a record of the mental process and represent artifacts that form the basis for critique and negotiation. Extending cooperation to collaboration, the amount of common goal-oriented risk taking, commitment, and resources that are required from the team members increases. Basic instruments that can keep team members in touch with their vision and help them make informed decisions can increase the chances of reaching their goals. Together with a very flexible approach towards content management these tools can help team members externalize their acts of thinking and use them as working materials, thus making collaboration a very efficient process.

Contents

Author's publications

[1] **M. I. Podean**, D. Benta, and L. Rusu, "About Creativity in Collaborative Systems. Why it matters and how it can be achieved," in *ICE-B 2011 - Proceedings of the International Conference on e-Business, Seville, Spain, July 18-21, 2011*, D. Marca, B. Shishkov, and M. van Sinderen, Eds. INSTICC Press, pp. 151–154.

[2] **M. I. Podean**, D. Benta, and R. A. Costin, "On supporting creative interaction in collaborative systems. A content oriented approach," in *The First International Workshop on Sustainable Enterprise Software (SES2011) held in conjunction with The 13th IEEE Conference on Commerce and Enterprise Computing (CEC2011), Luxembourg, September 5-6, 2011.*

[3] **M. I. Podean**, S. I. Nitchi, and D. Benta, "Content Management in the Context of Collaboration," accepted paper at The Third International Conference on Creative Content Technologies (CONTENT 2011), September 25-30, 2011, Rome, Italy.

[4] **M. I. Podean** and L. Rusu, "An XML-based Approach for Content Management in Virtual Organisations," in *ICE-B*, ser. ICE-B 2009 - Proceedings of the International Conference on e-Business, Milan, Italy, July 7-10, 2009, ICE-B is part of ICETE - The International Joint Conference on e-Business and Telecommunications, J. Filipe, D. A. Marca, B. Shishkov, and M. v. Sinderen, Eds. INSTICC Press, 2009, pp. 252–257.

[5] **M. I. Podean**, "The Role of XML Forms in Content Management," *Annals of the TiberiuPopoviciu, Supplement of International Workshopon Collaborative Systems andInformation Society*, pp. 333–340, 2008.

[6] L. Rusu, **M. I. Podean**, S. Szabolcs, and R. Costin, "A hierachical model for medical registrations," *Journal of Information Systems and Operations Management*, vol. 4, no. 2, pp. 32–42, 2010.

[7] **M. I. Podean**, "Document and Workflow Management in Collaborative Systems," *Economy Informatics*, vol. 8, no. 1, pp. 50–53, 2008.

[8] **M. I. Podean** and L. Rusu, "A content oriented approach for collaborative management in public organizations," in *Proceedings of The 16th International Economic Conference IECS2009, Industrial Revolutions, from the Globalization andPost Globalization Perspective*, 2009, pp. 182–188.

[9] S. I. Nitchi, A. Mihaila, and **M. I. Podean**, "Collaboration and Virtualization in Large Information Systems Project," *Informatica Economica*, vol. 13, no. 2, pp. 12–19, 2009.

[10] L. Rusu, M. Sarbu, and **M. I. Podean**, "Multilayer Solution using Multimap for Develope a Mobile Application," in *ICE-B*, ser. ICE-B 2009 - Proceedings of the International Conference on e-Business, Milan, Italy, July 7-10, 2009, ICE-B is part of ICETE - The International Joint Conference on e-Business and Telecommunications, J. Filipe, D. A. Marca, B. Shishkov, and M. v. Sinderen, Eds. INSTICC Press, 2009, pp. 135–138.

[11] L. Rusu, S. Kleinhempel, and **M. I. Podean**, "Entrepreneurship skills development for document management," in *Proceedings of the 6th International Seminar on Quality Management in Higher Education*, Tulcea, Romania, 2010, pp. 679–682.

[12] **M. I. Podean**, "Software as a Service. A new model for collaboration," in *Proceedings of The 9th International Conference on Informatics in Economy*, Bucuresti, 2009.

[13] D. Benta, L. Rusu, and **M. I. Podean**, "Successful Implemented theories for Reference Class Forecasting in industrial field," 2011, Accepted for publication at the International Joint Conference on e-Business and Telecommunications, ICE-B.

[14] D. Benta, **M. I. Podean**, S. Jecan, and C. Mircean, "Simple steps for Risk Management in small and medium projects," in *Proceedings of The 7th International Conference on Management of Technological Changes*, Alexandroupolis, Greece, 2011.

[15] D. Benta and **M. I. Podean**, "Risk Management approaches for successful projects," in *Proceedings of the 2nd Symposium on Business Informatics in central and Eastern Europe.* Osterreichische Computer Gesellschaft, 2011, pp. 39–49.

[16] **M. I. Podean**, D. Benta, and C. Mircean, "Overlapping Boundaries of the Project Time Management and Project Risk Management," *Informatica Economica*, vol. 14, no. 4, pp. 156–163, 2010.

[17] D. Benta, **M. I. Podean**, and C. Mircean, "On best practices for risk management in complex projects," *Informatica Economica. INFOREC Association*, 2011.

[18] **M. I. Podean** and D. Benta, "Tool Development for Reference Class Forecasting. Technical specifications," Corporate Technology, Project and Risk Management Department, Tech. Rep., 2010, Internal use only.

[19] D. Benta and **M. I. Podean**, "Tool Development for Reference Class Forecasting, Software Documentation - Requirements," Corporate Technology, Project and Risk Management Department, Tech. Rep., 2010, Internal use only.

[20] ——, "Tool Development for Reference Class Forecasting. Software Documentation - User Interface Specifications," Corporate Technology, Project and Risk Management Department, Tech. Rep., 2010, Internal use only.

Introduction

Problem statement

Collaboration is a process of shared creation that enables organizations and other forms of collective endeavor to develop innovative concepts and artifacts. As one of the goals of most successful organizations is to improve efficiency, a significant factor concerning distributed and virtual teams is the ability to manage the content involved in everyday activities, content that most often resides in documents.

Collaboration represents the most sophisticated level of inter-organizational relationships, consisting in the efforts to unite organizations and individuals in order to achieve *common goals* [1]. This jointure of organizations and people can be achieved at different levels - business models and technology, and in different forms - enterprise networks, virtual organizations, virtual laboratories [2] [3].

Although paper-based systems have developed in time complex processes, and computer-based systems for the moment do not support all the advantages of paper, more or less successful attempts have been made towards paperless offices [4]. Going paperless is a fundamental shift but is not sufficient, real productivity enhancements can be obtained by providing excellent flexibility in handling mixed content that can serve as human thought externalizations.

The development of web capabilities is significantly changing the way that people work, facilitating content creation and providing an easy way for distribution. A system that aims to improve collaboration in distributed teams should provide efficient and flexible method for content management. This is required in order to allow users to spend more time creating valuable content rather than handling technology. Such a system should try to remove space and time limitations and offer intuitive user interfaces capable of making content creation and management as easy as possible requiring no or minimal knowledge about the technology involved.

Current content management approaches focus on needs related to web and enterprise content, publishing houses or document management. Generally speaking, these approaches provide functionality like updating web pages or capturing, storing and delivering standardized format documents used in large organizations. A gap between the provided functionality and the actual needs related to content in a collaboration process exists. On the other hand, content management and collaborative systems are seen as two independent tools when they should be regarded as supplementing each other.

Discussing e-business innovation, [5] defined a model based on a highly flexible front-end and a highly standardized back-end (enabling thus cost-effective operations). XML is identified as the required technology in order to achieve the concept of flexibility with standardization. In order to be able to provide a reasonable e-workspace that integrates disparate systems and organizations, orchestrate web services and provide rich user interface capabilities, a flexible data interchange format and system is required. XML is seemed as a solution that could provide a lot more flexibility in handling content, but implementations based on the XML technology stack are scarce and provide limited functionality.

Aim and research limitations

The aim of the research is to identify ways in which content management, as a component of a collaborative system, can enhance collaboration. This involves a multi-disciplinary approach that must include a wide range of distinctive areas, such as collaboration and collaborative systems, project and risk management, linked data and shared vocabularies, content management and XML technologies.

In order to achieve this aim, we will focus the research on the following key concerns:

1. identify what makes collaboration efficient and what must a collaborative system provide as functionality in order to stimulate creative work,
2. identify the mean by which content management can support collaboration and compare these prerequisites with current approaches,
3. how to implement a content management system in order to provide considerable flexibility in terms of *i)* user interaction with content, and *i)* interaction with other systems and software.

On the other hand, our research does not take into consideration:

1. all aspects regarding collaboration and content management systems,
2. very structured content that can be managed using databases and/or ERP solutions, and
3. enterprise content.

Chapter 1

Theoretical overview

1.1 Collaboration and collaborative systems

Many domains reached a point in which the knowledge required for skillful, professional practice can no longer be acquired in a decade, factor that generates increased specialization [6]. This increased specialization makes collaboration to be crucial because complex problems require more knowledge than any particular person possesses. The relevant information required to solve complex problems is normally distributed among different persons or stakeholders. In order to create insight, new ideas or new artifacts it is considered a prerequisite to bring different and often controversial points of view together, and create a shared understanding among stakeholders.

It is generally considered that insight moments for creative individuals are the result of working in isolation, but it has been proven that the role of interaction and collaboration is critical [7]. The power of the individual mind following the Renaissance model and of unaided work has been overrated, while creative activity emerges from the relationship between individual and the world of his work and especially from the ties between an individual and other human beings [6].

1.1.1 Collaboration

Collaboration derives from the Latin *collaborare* that means to work together and can be seen as a process of shared creation. In a general sense, collaboration represents the act of working with other persons in order to achieve a common goal [8].

David Osher [1] identifies collaboration as being the most sophisticated level of relationship because it requires efforts to unite people and organizations in order to achieve common goals that could not be achieved by any particular individual or organization acting alone. On the other hand, collaboration [9] is regarded from a more project man-

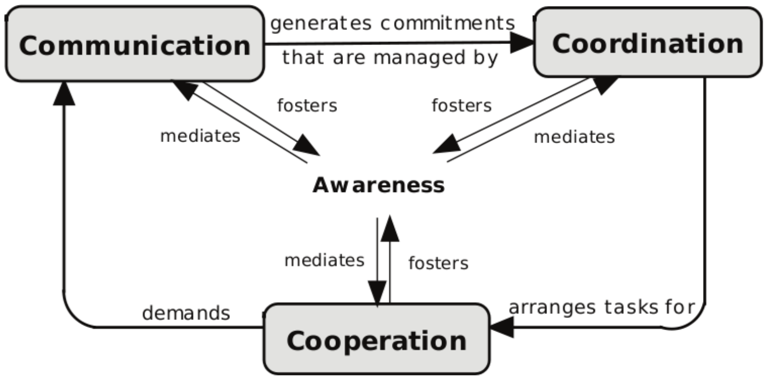


Figure 1.1: The 3C model of collaboration [11]

agement oriented point of view focusing on the elements required to achieve this level of relationship.

Michael Schrange [10] focuses mainly on the novelty of the groups goals and defines collaboration as a process of shared creation: two or more individuals with complementary skills interacting to create a shared understanding that none had previously possessed or could have come to on their own. From this point of view, collaboration creates a shared meaning about a process, a product, or an event; this shared understanding that no member could achieve on his own is regarded as being the result of real collaboration.

This concept is often confused with cooperation. Because the two terms are indistinguishable [9] for many people, in the following we will take a closer look at what collaboration is and how can it be attained.

1.1.2 The 3C model of collaboration

The 3C Collaboration Model is one common, traditional model that describes what is collaboration, and what are its main components. This model states that collaboration is attainable by implementing three main processes: communication (networking), coordination and cooperation. In the following, we will discuss this model starting from its description in [11].

Communication (or networking) is the starting point in each collaborative process (Figure 1.1) and represents the exchange of information for mutual benefit [12]. Collaboration has an iterative nature [11] because the members involved in a collaborative process obtain feedback from their actions and feed-through from the actions of their fellows. Feed-trough is received by using the information related to the interaction among participants.

Coordination refers to the management of people, their activities and resources [11]. It allows team members to manage conflicts and activities in order to increase the efficiency of communication and cooperation efforts. Communication is used as a foundation [12], but coordination involves also altering activities for mutual benefit and *common* goal. Coordination is usually implemented using tools like project management and process maps.

Cooperation [9] is a building block for collaboration is a process that requires that members exchange information, regulate their activities, and share resources in order to achieve *compatible* goals. As an example of cooperation, we can consider a traditional supply chain model based on client-supplier relationships and predefined roles in the value chain. In this model, each participant performs its part of the job, in a quasi-independent manner (although coordinated with others) [9]. A general plan exists, but defines only low-level co-working activities as their goals are compatible only in the sense that an end-product or service will be created in a value-chain model by putting together individual results.

Extending cooperation to collaboration, the amount of common goal-oriented risk taking, commitment, and resources that are required from the team members increases [9]. From this perspective, the degree of these various interactions can be regarded as collaboration maturity level or a degree of involvement towards collaboration. Common purpose is different from mutual benefit because is based on a shared vision.

Collaboration is about creating new ways to interact with each other [12]. It adds to the aforementioned elements enhancing the capabilities of another for mutual benefit and to achieve a common purpose by sharing risks, resources, responsibilities, and rewards. This enforces the idea that collaboration is based on the 3C model, but it is something more than that. In the following, we will discuss the critiques and limitations of the 3C model.

1.1.3 Creativity

The rationality behind collaboration is creativity. Creativity, and especially scientific creativity, is a process of achieving an outcome that is recognized as innovative by the relevant community. As defined in [13], this process does not occur in one persons head, but in the interaction between that persons thoughts and a sociocultural context.

Creativity can refer to the work of artists, but can also refer to every-day problem-solving abilities. This type of creativity is essentially, equally significant because enables people to become more productive and make better results. Support for divergent and convergent thinking, development of shared objectives and reflexivity [14] are identified as key requirements for creativity.

Clarity of goals is a necessary requirement for creativity flow [13] but team members have to take in consideration how they express these goals. Having clear objectives helps convergent thinking filter with greater precision. Developing shared objectives is a necessary condition for creativity because it requires group members to share their domain-specific knowledge and generates less resistance to change.

Obtaining immediate feedback is essential in having complete participation in the task at hand [15, p. 54]. In the context of a group, this refers to the extent to which members collectively reflect on the groups objectives. This process is known as reflexivity and consists of three elements: reflection, planning and action or adaptation.

Reflection is based on critical thinking, which is a form of thinking that is focused, disciplined, consistent and constrained. Planning creates conceptual readiness for relevant opportunities and guides group member attention towards actions and means to achieve goals. Planning generates high reflexivity if during the process factors like potential problems, hierarchical ordering and short/ long term planning are taken in consideration. Action or adaptation refers to the continuous renegotiation of groups reality during interaction between group members, and members and the environment. Adaptation consists in goal-directed behaviors that are relevant to achieving the desired changes in group objectives, strategies and processes identified by the group during the stage of reflection. Risk management is used to identify, mitigate and define action plans for the full range of uncertainty, including both risks and opportunities.

Externalization objects are essential to collaboration [6] because they *a*) create and store mental effort records, evidence that is outside the memory; and *b*) represent artifacts that provide information and form the basis for critique and debate. Very valuable assets for a group or organization are not only the results but also the way people think, the way they get to god results. It is a significant challenge to try to capture the thinking process in tools that are remarkably easy and intuitive to use.

1.2 Content management

As we have seen in Chapter 1.1, a particularly fundamental requirement for collaboration is the support for creative interaction with content. This can be achieved using content as an externalization of human thinking. This should allow users to model with ample flexibility how content is manipulated during the collaboration process. Using content as an externalization mechanism raises two challenges: how content should be stored and how can it be delivered to user so that they can interact with it in a manner close to their field.

The chapter will begin by identifying key elements of a content management system and discuss what is intelligent content. A closer look will be taken to related technologies

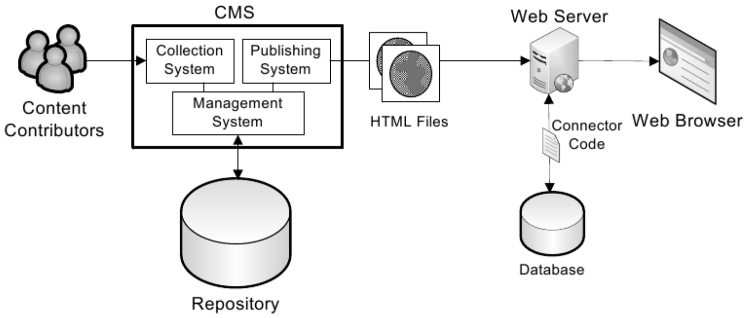


Figure 1.2: Basic structure of a content management system [16]

such as linked data, shared and open vocabularies in order to identify approaches that can promote flexibility in content management. XML and related technologies will be presented briefly since they are considered in many respects a viable solution for this problem. Access control is an issue concerning all interactions in collaborative systems, so some concepts will be briefly introduced.

1.2.1 Defining content management

Content management systems (CMS) represent solutions that manage the life-cycle of content and have to ensure that the integrity and meaning of the content is not altered by the system. They typically deal with different content types and formats, and comprises in [16] content acquisition and conversion (when data does not have the required format or structure).

A CMS consist in three main components [16] (Figure 1.2):

1. the collection system - responsible with conversion and aggregation. Conversion adapts the content to the system requirements regarding format and structure and aggregation prepares it for editing.

2. the management system - provides the administrative base. This component builds on three subcomponents: a) the repository, b) the administrative module to maintain the system by handling configuration parameters, access policies and content types, and the (c) workow module is responsible with the scheduling, coordination and enforcement of tasks.

3. the publishing system provides templates to extract data from the repository and prepare it for publication.

Ann Rockley defines **intelligent content** [17] as "content which is not limited to one purpose, technology or output. It is content that is structurally rich and semantically aware, and is, therefore, discoverable, reusable, reconfigurable and adaptable. It is content that helps you and your customers get the job done. It is content that works for you and it is limited only by your imagination. She also describes intelligent content as having the following characteristics:

structurally rich the structure has meaning, is semantically structured.

semantically aware content has a meaning, it can be tagged with metadata to determine the type within it.

discoverable this is true if the aforementioned characteristics are available and especially if the structure is defined using XML. Using tools like XQuery content can be retrieved, prepared and published.

reusable refers to the fact that content can be created once and used several times.

reconfigurable structured content has content separate from format or presentation thus making it easily to customize for different publishing channels in order to meet the needs of the channel. Content can also be automatically mixed to provide personalized information or transformed from one structure to another.

adaptable content is usually created with a particular audience in mind but, it can be adapted to meet different needs and illustrative examples are mashups who allow content to be aggregated.

Content reuse can improve the course content is created by increasing quality and consistency [18]. In addition, reuse provides support for fast reconfiguration, thus allowing easy re-purposing of content. Modular reusable content supports building entirely new content starting from existing components. Content reuse [19] makes possible the assembly of documents only when they are requested, avoiding the limitations of static content. Dynamic content views documents as a set of information objects that are assembled only in response to the users request and requirements. Dynamic content is based on personalization, which refers to providing specific and relevant content to defined users of user groups.

1.2.2 Linked data and shared vocabularies

In a content repository, metadata has a crucial role because complex authoring processes require some means of classifying and identifying content components. This is necessary to be able to retrieve and combine them in meaningful ways [19]. Metadata is information about information and results from the processes of labeling, cataloging and describing content. Metadata allows content to be properly processed and searched by computers. It can be used to describe processes, rules and structure of the content, not just provide descriptive information.

Metadata enables effective retrieval, content reuse (automatic population of existing content in document templates), routing based on workflows, status tracking and reporting. According to the activities performed in relation to the content, metadata can have three functions [19]:

1. **reuse**: eliminates content authoring redundancies but has to be applied at the element level,

2. **retrieval**: allows content to be retrieved by searching the content repository, and

3. **tracking**: this type of metadata is particularly useful when implementing workflow as part of the content management system.

Structured content allows content reuse without the need for manual adjustments [19]. It provides improvements on readability, usability, consistency and reduces maintenance efforts. Structured content relies on content standards in order to determine the type of content in each element and does not refer to format standards. Format is critical in helping users understand the information and refers to how information must look. Format specifications should be kept separate from structure in order to ensure reuse.

Linked data refers to a set of practices for publishing and connecting structured data over the Internet [20]. The focus here is not on the structure of linked data but rather on the prerequisites that content has to meet in order to be later used in linked data structures. Following these prerequisites will help get content that normally resides in traditional documents, and enable rich operations on it. The motivation is to get content out of complex format that make content reuse by automatic means difficult.

Linked data has three particular characteristics that are of interest for our approach [20]:

- data is present on the Internet in machine readable and non-proprietary formats [21],

- data is strictly separated from formatting and presentation details,

- data is self-describing: when an unfamiliar vocabulary is encountered, using the URIs that identify that particular vocabulary additional information about its meaning can be discovered

A system that aims to improve collaboration in distributed organizations or teams should provide efficient and flexible means for content management in order to allow the user to spend more time creating valuable content rather than handling technology [22].

As we have seen, content standards exist to enable information sharing among groups with common interests. These standards emphasize on the fact that content must be separated from its presentation in order to allow reuse and operations not possible in closed format documents.

1.3 Conclusions

As we have seen earlier, the rationality behind collaboration is creativity and in order to generate insight, new ideas or new artifacts it is a prerequisite to bring together different and often controversial points of view. Collaboration is about creating shared understanding that no member could achieve on his own. This process can be achieved if defined in a goal-oriented framework. Collaboration uses as a backbone communication, coordination and cooperation, but it is seen as something more than that.

Collaboration can be achieved if defined in a goal-oriented framework. Vision specifies the scope and scale of these benefits that a team can achieve if their enterprise is successful, but does not provide the means to achieve them. Project management is a tool used to provide a team the capabilities required to produce the benefits defined by vision. While vision delineates a strategy, project management sets the tactics by detailing the steps required to put it in practice. On the other hand, risk management is about handling uncertainty so that teams take advantage more efficiently when new opportunities arise and develop responses for actions that can have a negative impact on goals and vision.

Intelligent human performance is based on the interaction between the mind and tools and groups of minds in interaction with each other. A shared workspace is particularly beneficial because it allows group members to count on group memory and it provides also some basic awareness mechanisms. As we have seen, externalizations are of considerable importance because they create a record of the cognitive process and represent artifacts that form the basis for critique and negotiation.

Based on this, we can state that basic instruments that can keep teams in touch with their vision and help them make informed decisions, are particularly required. On the other hand, a highly flexible approach towards content management can help team members externalize their acts of thinking and use them as working materials. In the followings, we will take a closer look at content management focusing on aspects that support collaboration.

A content management system is a solution that must manage the entire content life-cycle, starting with content collection (import from other sources or create using a graphical user interfaces), managing content objects and publishing. This process has to implement the meanings so that both integrity and meaning is not altered by the system. In order to support collaboration, these components have to be implemented using a focus on intelligent content.

Intelligent content has been defined as material not limited to one purpose, technology or output and is structurally rich. It is content that limited only by the teams imagination, as Ann Rockley has defined it. Content reuse is a critical feature because makes possible the assembly of documents only when they are requested, avoiding the limitations of static content. Structured content allows content reuse without the need for manual adjustments, and it provides improvements on readability, usability, consistency. Following the prerequisites of linked data, we have seen that content has to be in a nonproprietary format and have a distinction between data and formatting in order to provide interlinking.

Chapter 2

Collaboration and content management models

In the background information, we have found that collaboration is an extraordinarily complex process that must be approached using a multidisciplinary perspective. Collaboration as a process can become more effective if all required steps are considered and appropriate tools adapted to modern needs. As we have seen in the previous chapters, besides tools to help team members manage their progress (e.g. managing goals, actions, uncertainty responses), tools to help them focus on the task at hand by minimizing the effort required to manage content are required. In order to be able to evaluate how different technologies can be integrated to solve the aforementioned issues, a design phase must precede.

The aim of this chapter is to define models for a collaborative system and a content management component starting from what has been identified in the background material.

2.1 Collaboration model

In Section 1.1.1 we have found that collaboration is the process of achieving a shared understanding that no member could accomplish acting alone, and it builds on communication, cooperation and coordination. Since collaboration can be attained in a shared-goal framework and cooperation and coordination are just some of the instruments that can support it, tools from the project and risk management must be added.

Taking in consideration that generating insight, new ideas or new artifacts, or simply putting it, creativity, is the rationality behind collaboration, tools for flexible content management must be regarded when designing a collaborative system.

2.1.1 Requirements for a collaborative system

Communication tools. In order to support collaborations needs regarding communication, tools like video and audio conferencing, chat, instant messaging, shared space to support group memory (wiki and forum) should be implemented. Annotation based communication should be integrated also in order to serve as a foundation for content negotiation.

Flexible coordination mechanisms. A coordination mechanism should allow dynamic redefinition of the interaction procedure in order to allow users to customize the process to their needs.

Accessible coordination scheme. A coordination mechanism should provide access to the coordination scheme and not have it deeply embedded in the systems implementation.

Shared workspace. A shared workspace that allows team members to work together and that provides tools like version and access control and authorization is required.

Shared goals. A tool that supports the definition of a shared vision and a description of this vision in terms of shared goals is essential in upgrading from 3C to collaboration. This tool must allow users to (re)adjust their goals according to changes in the environment.

Project management. Project management, through its main subprocesses, covers all this aspects and we consider that its need in a collaborative system is critical because its main feature is to make complexity manageable.

Integrated risk management. Integrated risk management tools are a must in order to understand, mitigate and establish action plans that cover decisions at both strategic and tactics level and include not only threats but also opportunities.

Workflow management. A technique that can manage the sequence of tasks and their execution for a particular process is required in order to enable the coordination process. This component should be a part of the content management subsystem but, it should derive its jobs from the project and risk management plan. *Process maps.* A tool that provides a clear understanding for the planning process is required. This instrument should render in an easy to understand style the required workflow for reaching a goal. A process map should provide also a time line for the processes.

Rings of involvement. A communication and management extension that will enforce the use of rings of involvement strategy is required in order to keep all relevant members up to date with the course of the project, and make skillful use of their valuable contribution.

Consensus building. The process of building consensus requires support from both communication tools and content management solution. Building consensus should allow people express their view-points for a particular topic, describe and state an argumentation for them, and at a group level, allow content to serve as a negotiation object (using the symmetry of ignorance approach [6]).

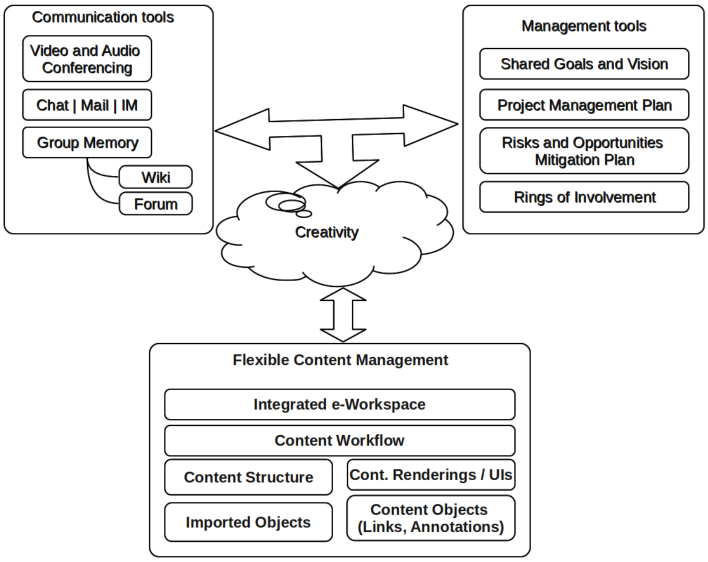


Figure 2.1: Collaboration model centered on creativity

Flexible content management. A tool that will allow users to *i)* use content as an externalization of their thinking and stimulate new connections between concepts and *ii)* take care of the publishing aspects (formating, export etc.) and provide multiple user interfaces for content according to context, will allow users to achieve original results.

2.1.2 Proposed model for a collaborative system

Our model is build around this requirements and starts from the premise that all are a must in order to achieve efficient collaboration. It separates functionality in three main responsibility areas, namely communication, management and content. As in contrast with the 3C model (communication - coordination - cooperation), our approach focuses on providing a holistic model that tries to cover all aspects of collaboration.

All three modules should be implemented in direct relation with each other (Figure 2.1), so that they enhance their functionality. As an example, we can consider the tool for managing shared goals that should be supported by appropriate communication tools, and changes in its structure reflected in the project and risk management plan.

As a requirement for collaboration, flexible content management has a substantial share. Together with consensus building, this is one of the components that have very little support in nowadays implementations. 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 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 approach is based on an integrated, intelligent content focused content management module that can support knowledge workers.

In our view, a content management module must include a shared workspace that integrates access to the rest of the tools and allows interaction with content and team members. Starting from the project management plan, in a semi-automatic manner, a workflow scheme should be generated in order to coordinate people and content artifacts. A workflow management subcomponent should handle these requirements and should serve as an enactment engine for the content elements along the execution of a project. From the workflow scheme a process map that reflects the rings of participation should be derived in order to keep users up-to-date with their responsibility coverage and upcoming tasks.

A flexible content management tool serves as a support in order to use content as an externalization of the thinking process and is able to handle all aspect related to importing, editing and publishing content. Such a system should provide also the means for content annotation and referencing. All these aspects concerning content management will be discussed in greater extent in the following sections.

2.2 Content management model

A flexible content management module is an essential requirement for a collaborative system. In the following, we will derive the requirements for such a component from following the findings in Sections Collaboration and collaborative systems and Content management.

Based on these we will outline our model for flexible content management that serves as a support and enabler for collaboration. Possible implementation techniques and tools are suggested were appropriate. Following the presented model, a use case analysis will be taken in consideration to explore two principal situations that a team will be confronted with: *a)* a phase focus on deriving creative insight, and *b)* a phase that is focused on individual endeavor and synthesis of results. This two use cases are typical for all knowledge worker teams since not all interaction during the execution of a project is strictly based on producing original artifacts, but involves also individual work and combining results, ordinary tasks execution etc.

2.2.1 Requirements for a content management component

Content management is generally regarded in term of web or enterprise content, or as document management. We find that documents are mere representations of content following a certain format (most often closed-source).

Handle entire content life-cycle. A solution that handles content should take in consideration all aspects from acquisition (either by creating new content, either by importing it), management (automatic handling all life-cycle tasks) and publishing in different types of formats.

Externalization. Content should represent and store members mental effort by emulating the requirements of the problem at hand and not force users to express their thinking in terms of a universal model that tries to satisfy all situations.

Content can be viewed as a knowledge container that implies what human beings transform their expertise in when they want to communicate with other people [23]. On the other hand, the concept of knowledge can be regarded as the result of human experience and reflection based on a set of beliefs and residing as fictive objects in peoples mind [24].

Basis for critique, negotiation and consensus building. Content objects should allow user interaction in the form of criticism and negotiation in order to have a clear understanding of each ones point of view and drive work based on consensus building.

Content must be structurally rich and semantically aware. In order to make content easy to operate it must be structurally rich and its elements should have a predefined sense and allow metadata to be attached in order to provide additional information.

Content standards and structured document formats. In order to establish a common understanding between team members and facilitate the exchange with other applications, a content management module should be based on content standards.

Use non-proprietary formats. In order to allow metadata to enhance the sense of content, enable transformation from one format to another, and mostly content reuse, the use of nonproprietary formats is a must.

Separation of content and presentation. Separation of data from the presentation will provide a greater flexibility and provide a richer range of functionality that can be defined for a particular content type.

Content should be adaptable. Content adaptability refers to the ability to easily change the destination of a particular content object.

Single source / multiple publishing formats. This requirement allows content to be reconfigurable, to be able to add new layout models to it with no alteration to the content object or document.

Project and content level automation. All the interaction that the execution of a project involves should be automated with the use of workflows. Workflows should be detailed on the project and content level.

As discussed earlier, project management is a vital tool for effective collaboration and this has implications at the content level, as well. The project plan should be transposed in a workflow scheme that will integrate all content components involved in the project and enforce the following of a specific course of execution. The orchestration of a process represents the automatic coordination and execution of a number of services required in order to obtain a specific output. Each process the system is required to complete must have a workflow definition schema in order to be implementable.

System accessibility and adaptability. In order to allow users to customize the system according to their content use cases, elements like content structure, content presentation, workflow definitions and variables should not be embedded in the system but fairly accessible to users and to allow changes to be made.

2.2.2 Proposed model for a content management component

Our model of content management focuses on handling the entire life-cycle of the content and is targeting content users that must interact in a creative manner with the content. Since the process of collaboration does not consist strictly in moments when creativity is highly required and includes also stages that consist in information gathering, preparing reports and documentation, our model is built around both these two use cases.

Our model (Figure 2.2) consists in three main components that constitute the centre and other auxiliary modules that make it behave like a system. The main modules are responsible with handling the content interaction, the processing of external objects and project level management. A credential module is used to handle authentication and authorization and evaluate access policies. The workspace that makes all this separation transparent to the user includes a user interface module and a component that manages communication with the other modules.

The model is based on loosed coupling between components, each of them being designed to solve a particular problem in an independent manner. Components follow a contract based model of providing functionality (the contract being actually defined by their API). The model is designed in such a manner that it can be extended by others, since we consider that no design can cover the full spectrum of needs regarding content management. The model is built starting from the assumption that the needs of a particular area could not be fully anticipated in advance and that the best approach is to create enablers for key areas of concern (content interaction, external objects and project workflow) and allow users to build their use cases starting from this. Following this principle, a Community ecosystem component is available is our model in order to indicate the importance of having an ecosystem built around the system (be it at organizational or industry level) that should validate the existing uses cases and create new ones. We will return to this issues in the following chapter when we will provide more details about the functional behavior for the core components.

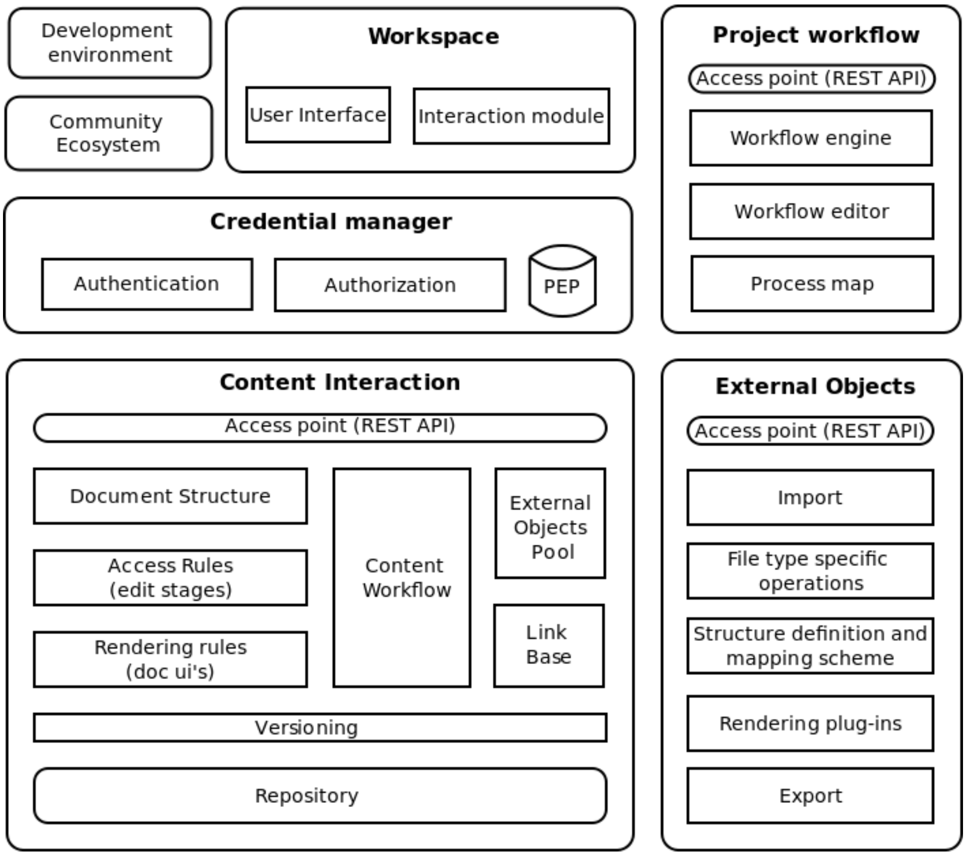


Figure 2.2: Collaboration enabler content management model

Content interaction. This module is responsible with handling the content definitions (content standards) that are used to organize the content, and using the rendering rules to generate the appropriate representation of the content. The main responsibilities of the module covers: content structures, access rules, rendering rules, content workflow, link base management, external object pool handling, versioning, and repository management.

The component provides a communication interface so that other component should know how to interact with it. The interface of the module follows the Representational State Transfer (REST) [25, 26] method of designing web services. It provides also an application programming interface (API) so that can be accessed and extended.

External objects. This component is responsible with integrating content from closed formats into the system. It manages processes like *i)* transforming content from one closed format to a structured document, *ii)* define a set of actions (operations) that can be performed on a particular type, *iii)* handle transformations to other formats that are used in the system, and *iv)* allow the conversion of the content back to the originating format. This component allows users to define rendering plug-ins for imported content. This are particular helpful if we consider the variety of content and its display formats (e.g. the description of a genome can be stored in various formats but a genome has also both 2D and 3D representations).

Project workflow. This component coordinates the aforementioned two components in order to fully manage the users use cases. This component is based largely on a workflow engine that acts as an enabler for the other two components and communicates with them through their interface. It should provide a mechanism for users to customize the workflow, but as mentioned in the previous chapter, this should be derived from the project management plan defined in the collaborative system.

Credential manager. Its primary role is to handle authentication and based on access policies to determine authorization rules. This is particularly useful in large organizations where delegation of authority for specific tasks is a practice. The organization defines its set of access policies and when a user connects to the system first its credential are checked and for each project that he or she tries to access, policies that define the tasks that it can operate on content are computed. Following the outcome provided by this component the content interaction module will generate the appropriate representation of the content.

Workspace. The workspace is the user interface component that will make transparent to the user all this modular structure implementing a project based shared workspace where users can collaborate. This module should incorporate access to tools defined in the collaborative model as well.

This model provides flexibility in handling content since content is separated from representation and it can have multiple user interfaces according to user needs, thus making it easy customizable and adaptable. Along this feature, content automation is included in order to manage the processes involved in executing a project. The model places the aforementioned requirements in a collaboration context, thus being more concerned with human interaction with content. We consider that a framework that implements the aforementioned requirements using the XML stack of technologies can serve as a flexible externalization mechanism in order to promote creative insight in collaboration and together with management tools increase the teams chances in reaching their goals.

2.3 Conclusions

A collaborative system requires tools to cover three main areas of concern: communication, management and content. These tools are more or less present in current implementations. A collaborative model that is focused on the creative outcome of the process must implements the appropriate means to manage goals and working plan management, uncertainty mitigation, consensus building and the involvement of all shareholders and a flexible content management component that will allow users to customize the way they interact with content. Our collaborative model integrates the aforementioned requirements and it is focused on the outcome of collaboration - creativity.

In order to allow team members to develop new understanding, a content management component must provide the mean to interact with content in such a way that content objects could be used as the basis for negotiation and consensus building. On the other hand, such a system must handle the entire content life-cycle and make it structurally rich, semantically aware and adaptable.

Our model is divided in three main components: *Content Interaction*, *External Objects* and *Project Workflow*. Content interaction is responsible with creating content representations, content level workflow, storing and versioning, while External objects is responsible with importing content from closed source formats, defining format specific operations, handling rendering plug-ins, and exporting to other formats. Project workflow interacts with the collaborative systems management module in order to convert the management plan into a workflow schema that can coordinate the content workflow at the project level.

The presented content management component model raises particular implementation issues since it is based on principles like single source - multiple delivery / publishing channels, separation of content from presentation and use of content standards. These aspects are available in certain implementations, but they are not integrated to form a whole system. Our content model covers the need to integrate these aspects and includes also requirements derived from collaboration.

Chapter 3

Implementation

The content management framework raises some implementation challenges since it requires *a)* to import content from different formats, *b)* provide a minimal set of format specific actions in order to interact with content objects, and *c)* have a flexible content level workflow mechanism. Following the model presented in Chapter 2, in the current chapter we will analyze some of the implementation challenges.

3.1 General considerations

The framework is based on a model designed to evolute and be extended and uses libraries of reusable components [27]. The implementation is designed as a web application that follows a resource-oriented model [28] based on loose coupled [29] components. Component integration / interaction is based on Representational State Transfer based (RESTful) web services [26] because they focus on resources and loose-coupled components.

Our proposal regarding implementation [30] is to use XML as storage and exchange medium for all content involved in the system [31]. This implies that metadata must be managed in XML as well [32]. The Content Interaction and External Objects components presented in the following sections are based on XML technologies that process content directly with no additional translation. Other languages are used (e.g. Java) only where it is more efficient to do so. This will allow us to use technologies like XQuery [33], XSLT [34] and XProc [35] for server side processing. Content is described using XML Schema [36] that will be used to validate XML documents. On the client side, XForms [37] in conjunction with JavaScript will be used.

XQuery has been widely tested as a server-side technology to create enterprise applications [38]. eXist[1] will be used as a native XML database because it integrates a web server and provides support for the majority of technologies required by our implementation.

[1]eXist-db is an open source native XML database : http://exist.sourceforge.net/

3.2 Handling document representations on the client-side

The term document will be further referred as a temporary representation of a content object. These representations will consist in web applications that will allow interaction with content objects. Content representations require a flexible method that is able to manage on the client the content object and provide an easy to use graphical user interface. In order to determine the most effective method, we will compare two approaches, using JavaScript and XForms. Aspects that are taken into consideration refer to the capability of handling and displaying content objects on the client.

3.2.1 JavaScript

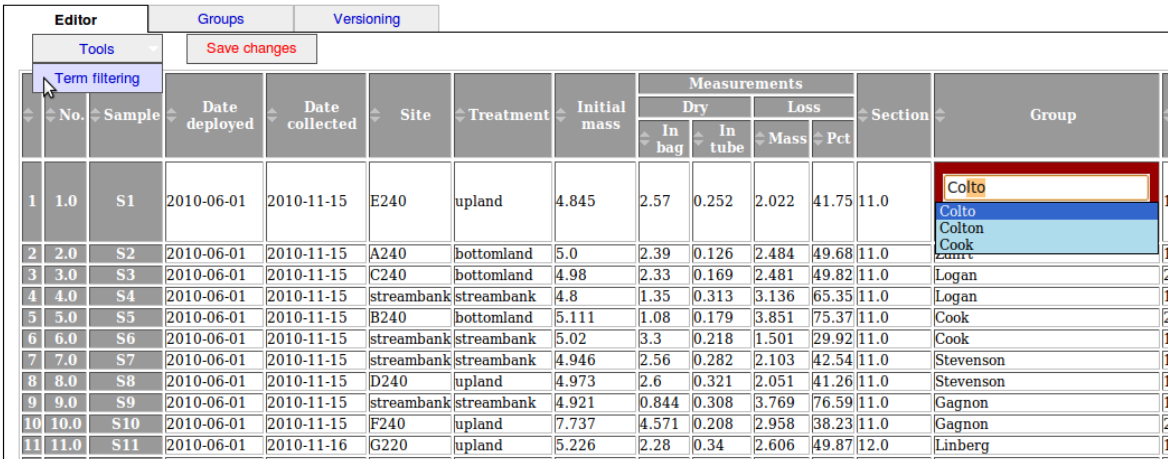
Having all content stored in XML imposes some constraints on developing client-side web applications. For all content object types, JavaScript packages must be created that describe the interaction, on the client-side, with the content. This requires on the server-side to create the structure where JavaScript function calls will be placed and bind functions with the user interface.

Processing content representations on the client using JavaScript requires two main steps. First, the DOM element that will hold the content structure has to be built, and then bindings with the user interface elements must be created. The DOM element implements all required methods to interact with the XML structure, like adding a new element or attribute, updating an existing one or deleting a node.

The DOM is a powerful tool to work with document elements since it allows control over all defining aspects. One serious drawback is that interaction is done at a low level in JavaScript. This requires that each action must be explicitly defined and very little abstraction is provided. Handling content using DOM usually results in lengthy and complex implementations. A simpler solution to work with XML content is required.

3.2.2 XForms

The effectiveness of a web form depends upon the ability to provide a rich user experience, validate user input and reduce round trips for server-side processing [39]. An HTML form requires additional JavaScript in order to perform client-side validation, but taking in consideration the fact that any user can easily disable JavaScript support in his browser, the need for additional server-side validation is crucial. Even with AJAX, these round trips to the server could not be eliminated, which often results in inadequate end-user experience, especially when dealing with complex forms. Providing rich user interaction



	No.	Sample	Date deployed	Date collected	Site	Treatment	Initial mass	Measurements: Dry: In bag	Measurements: Dry: In tube	Measurements: Loss: Mass	Measurements: Loss: Pct	Section	Group
1	1.0	S1	2010-06-01	2010-11-15	E240	upland	4.845	2.57	0.252	2.022	41.75	11.0	Colto
2	2.0	S2	2010-06-01	2010-11-15	A240	bottomland	5.0	2.39	0.126	2.484	49.68	11.0	[illegible]
3	3.0	S3	2010-06-01	2010-11-15	C240	bottomland	4.98	2.33	0.169	2.481	49.82	11.0	Logan
4	4.0	S4	2010-06-01	2010-11-15	streambank	streambank	4.8	1.35	0.313	3.136	65.35	11.0	Logan
5	5.0	S5	2010-06-01	2010-11-15	B240	bottomland	5.111	1.08	0.179	3.851	75.37	11.0	Cook
6	6.0	S6	2010-06-01	2010-11-15	streambank	streambank	5.02	3.3	0.218	1.501	29.92	11.0	Cook
7	7.0	S7	2010-06-01	2010-11-15	streambank	streambank	4.946	2.56	0.282	2.103	42.54	11.0	Stevenson
8	8.0	S8	2010-06-01	2010-11-15	D240	upland	4.973	2.6	0.321	2.051	41.26	11.0	Stevenson
9	9.0	S9	2010-06-01	2010-11-15	streambank	streambank	4.921	0.844	0.308	3.769	76.59	11.0	Gagnon
10	10.0	S10	2010-06-01	2010-11-15	F240	upland	7.737	4.571	0.208	2.958	38.23	11.0	Gagnon
11	11.0	S11	2010-06-01	2010-11-16	G220	upland	5.226	2.28	0.34	2.606	49.87	12.0	Linberg

Figure 3.1: Main representation for Excel content[1]

requires a vast amount of scripts which often have to be separately implemented for different types of browsers. Another drawback is the lack of support for transactions. Each phase has to be treated separately, as data is gathered on server-side, processed, send back to the client and the cycle reiterates for each stage. HTML forms have little support for data representation, as urlencoded and multipart represent actually at data and name/value pairs.

3.2.3 Combining XForms with JavaScript

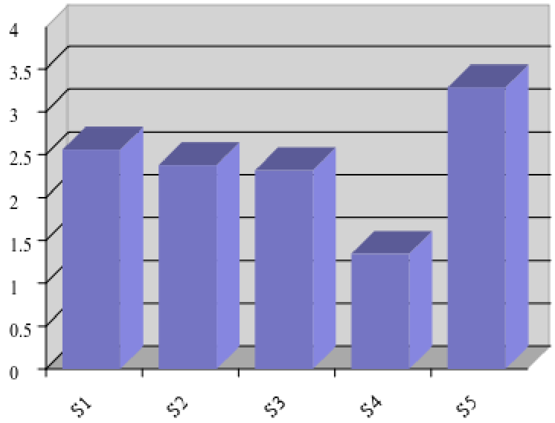
Figure 3.1 depicts the main representation of a dataset imported from an Excel file. The application is built using XForms and additional functionality added using JavaScript. The concerns related to this problem divide into content and interaction related. Content issues refer to the ability of creating new elements (e.g. creating a new table line), updating existing values, storing these changes and sending them to the server. Interaction issues refer, for example, to how the cell update functionality is provided, how the dataset can be viewed (e.g. sorting dataset content using certain column as index).

XForms is best suited at interacting with XML data, therefore it is used to manage content interaction concerns. As seen in the previous sections, using XForms to update content objects is more efficient than JavaScript, since it is more specialized on this kind of operations. While creating a new table row requires in JavaScript a large number of actions to be performed (e.g. creating the element, adding it to the DOM etc.), in XForms this can be achieved with just a few lines of code with a lesser degree of complexity.

[1]Dataset created using Excel file from http://www.hope.edu/academic/biology/classdata/bio280/

```
<graphics>
    <barChart width="450" height="500">
        <yAxis start="0" end="4" step="0.5"/>
        <display>
            <bar name="S1" value="2.57" />
            <bar name="S2" value="2.39" />
            <bar name="S3" value="2.33" />
            <bar name="S4" value="1.35" />
            <bar name="S5" value="3.3" />
        </display>
    </barChart>
</graphics>
```

(a) Chart data and configuration



(b) Dataset graphical display using SVG

Figure 3.2: Displaying data as a chart using SVG

3.2.4 Handling graphical content representations

Content can have different representations, either as multiple document representations, or as object representations in the same documents. As an example, we can consider a tabular dataset that can be displayed as a table or as a chart. In order to display graphical data renderings, technologies like Scalable Vector Graphics (SVG) [40] will be used.

SVG can be created using Java[1], but since it is an XML format, it can be generated more efficiently using XSLT. As it can be seen in Figure 3.2, by applying on a simple dataset (Figure 3.2a) a XSLT stylesheet specialized on producing a certain type of graphical data, a chart representation (Figure 3.2b) can be obtained and embedded in the main representation. If the team works on a report that uses as a data source content from an Excel file, they can create a group that selects only the required data, and define it as an object that is included in the main report. For this object, two main representations can be selected: a tabular or a graphical one. Having charts delivered in SVG, the quality of the graphical elements in the final document will not be negatively influence by the delivery medium (web, print, mobile etc.).

SVG is easy embeddable in XForms [41] and since it is natively supported by the majority of browsers it represents an effective method for our framework to provide graphical data representations.

[1]Batik is a Java-based toolkit for SVG: http://xmlgraphics.apache.org/batik/index.html

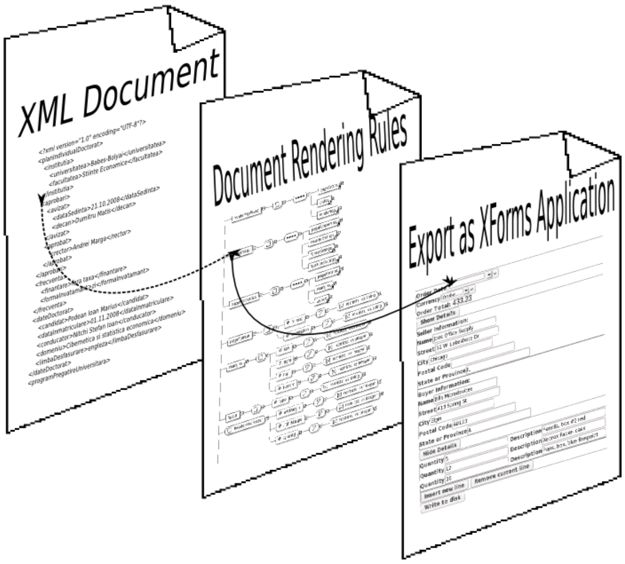


Figure 3.3: Creating a content object representation

3.3 Server-side content handling

Delivering content representations on the client-side using XML technologies, requires an appropriate solution on the server-side. The required method must be specialized on handling XML content in order to cover two key functionality sets: generating content representation and managing content-level workflow. In the following we will analyze XSLT and XQuery for handling content representations, and XProc for content workflow.

3.3.1 Generating XForms using XSLT and XQuery

XForms applications being XML based can be easy generated using tools like XSLT or XQuery. In the following we will examine two use cases of generating content representations in XForms using XSLT and XQuery.

Each content object is described using three metadata (Figure 3.3) files:

1. a description file that provides general information regarding the current instance of the object, like the document object type that it instantiates or extends, versioning data, link to the source content file, roles involved in the edit, links to rendering rules and services.

2. rendering rules that describe the visual representation

3. a set of editing rules that describe the actions that can be taken on objects present in the document. These rules must be derived from the access policies.

Together with the source file that stores just the content of the document, these files represent the minimum required in order to describe a content object. Additional metadata files can be added in order to assign additional functionality to documents, as we will see in the following section.

In XSLT, our implementation starts with loading all metadata and from elements that are often used variables are created. XSLT is based on template rules that execute a sequence of commands when a pattern is detected. The template for the root element builds the main HTML document. From the rendering rules, page details and style definitions are loaded and transformed into CSS classes. The main structure of the XForms application will be constructed, starting with the submission details related to how to save changes made on the document.

In the XQuery implementation the difference is that it allows a more fine grained extraction of details from the XSD definitions. XQuery uses FLWOR expressions (for, let, where, order by, and return) to combine / create data elements that make the language easier to use on certain content. XQuery allows functions to be grouped in modules, which makes code easier to manage. As a comparison, we can say that the XQuery code required to produce the document rendering is lengthier then the XSLT one, but it allowed as to solve easier and in a more intuitive way some issues (e.g. recursive functions).

The idea of implementing the same problem using two different technologies started from one basic need: decide which one is more suitable for the implementation of the framework. We reached the conclusion that it is not appropriate to compare the two and say one is better over the other. Each technology focuses on different aspects concerning XML processing. XSLT is specialized on transforming one document to another based on patterns present in the source file. XQuery on the other hand provides functionality closer to a procedural programming language and its constructs make it effective as a scripting language for the server side. As a resolution, we can conclude that it is more suited to join them and take advantage of each ones focus. Integrating the two languages raises no difficulties since the XQuery implementation used allows XSLT calls from inside a XQuery script.

3.3.2 Content workflow

Content level workflow defines the interaction between content objects, their associated rules, transformation schemas and other tools in order to ensure the delivery of a particular representation according to edit stage. According to stage, certain actions can be performed by associated roles (edit rules) and these actions are allowed using a specific representation (e.g. edit, print). In the same respect, interaction with external objects must be taken into consideration. External objects provide an interface using web services, therefore the content level workflow mechanism must be able to interact using this method.

Taking in consideration that all content is stored in XML and handled by related tools, a mechanism that intends to integrate these tools must be able to interact with minimal cost with them. Certain programming languages have been taken into consideration (PHP, Perl and Python), but they all introduce some degree of complexity when handling XML and XML tools. Our choice of programming languages for content level workflow implementation is XProc [35], which is a language designed to illustrate operations to be performed on XML documents. This approach is more appropriate because it is designed with processing XML documents in mind and integrating tools that can work with XML documents.

3.4 Metadata

Metadata is an essential aspect for the framework because it helps organize content, associate actions and renderings. Managing metadata in XML in our prototype has its roots in results from managing health records [42]. In the framework, there are two points where metadata is managed: in the main repository and in the external objects pool.

The first organizes content objects created starting from supported content standards. It defines for them rendering and edit rules, roles that apply for a particular instantiation of a content object, versioning information and tools to be used to manage the content. The main repository includes also, for each object, a link base that defines the inclusion of external objects in a content object.

In the external objects pool, objects metadata has a relatively heterogeneous structure. All objects have an *id*, *type* and some versioning information in a similar manner, but the rest of the description is content type dependent. The versioning data states the time when the revision took place, the action that triggered the change (initial import, edit using the main representation and term filtering) and files that store the changes that lead to the new version.

3.5 Managing content from closed-source formats

In order to present the functionality of this module we will use as an example content stored in Microsoft Excel 2003 format. For this format, we will discuss how can it be converted to XML, define format specific operations and export it back to the originating format. The prototype discussed in this section, together with parts from the previous, has been presented at the *DemoJam*[1] event held by MarkLogic in conjunction with the XML Prague 2011 conference.

3.5.1 Importing from closed-source formats

The import from closed-source formats is handled in our model with the use of format-specific plug-ins. These plug-ins must be provided either as a web- service or as a Java package in order to be integrated in the system. Each file-type will have associated at least one plug-in in order to have access to content. The External Objects module will designate the appropriate module when a request is made. The main module should provide also a sandbox where users can load their personal plug-ins. If the plug-in is accepted by the community, it can be included in the main distribution.

As an example of such a plug-in we can examine the use case of handling data stored in a Microsoft Excel file. This type of file imposes specific challenges since the material is split in headers that describe the content and actual data. Other types of objects are available also in a regular Excel file, such as charts, but these are ignored at import since they are mere representations of the data. The content imported using the External Objects module will serve as data sources and will have defined in the system specific representations.

Importing data from an Excel file is vital for the majority of projects since this format is often used thanks to its flexibility to work with data. The limitations of this format are visible when using data in different places, such as reports, articles etc. If the initial dataset is modified, the user is required to manually update all occurrences of the data. Our approach is to establish a mechanism that will import an entire file and define it as a dataset that can be referenced in other content objects involved in the task at hand. Following this approach, sets of data can be selected and referenced with the use of links in other documents. This will eliminate the need to manually update all content occurrences and avoid data inconsistencies.

The file type will have a rendering schema that will provide a limited number of actions that can be applied to the content by mimicking the originating application that generated the content. If modifications occur on the initial file, a new import can be done and by the use of file type specific operations it can be merged with the dataset. A new version of the dataset will be generated.

This Java package extracts all information required to describe the dataset, but does not further process it since it is more efficient to do that in more specialized languages like XQuery. The choice was to deploy the package using the .jar format and include it

[1] DemoJam website: `http://developer.marklogic.com/news/xmlprague-2011-demojam-winner`

in the web server in order to be called from XQuery as a Java command. This reduced the burden on the system and made the interaction more natural.

3.5.2 Format specific operations

Tabular data is commonly used as a data source for further processing before being actually included as a representation in some format that is meant for printing (e.g. reports, articles etc.). A productive operation that can be applied on the imported dataset is to clean the typos or other errors that can lead to inconsistencies. If the intention is to import a set of data that has been gathered during some experiment and starting from this, together with team members, to analyze it in order to derive some results, it is crucial to remove, from the very beginning, all quality issues that might negatively influence the final results. Since most statistical analysis tools do not provide a mechanism to clean typos in the dataset, errors pass unnoticed in large datasets.

In order to avoid this, as a file-type specific operation we have defined a cleaning operation that computes similarities (Levenshtein Distance [43]) between all string values from a column in order to identify misspelled values (e.g. for data gathered from questionnaires one common mistake appears in typing region names, like for Cluj-Napoca could result from misspelling "Cluj-Napoca␣" or "Cluj Napoca"). Being able to remove all this errors at import will lead to faster access to clean data and avoid wasting team members time.

The file type specific operations are defined similar to import plug-ins, as extensions for the XQuery main application. When the cleaning process is triggered, for each column that is of type *string* similarities between unique values are computed. The results are then sent to the client, and using a XForms application these can be visualized and replace rules created. The rendering for this operation presents the results in three columns (Figure 3.4): all singular values identified for a column, all terms that have similarities and the substitute rules defined by the user.

After all replacing rules have been defined on the client-side, these are sent to the server using a *POST* request. The replacing rules are stored on the server and used to update the unique values repository that is created for each dataset. This repository stores all unique values for columns that contain string data. This repository is used for computing similarities and as a support for the user interface. On the user interface the values stored in this repository will be used for auto-complete lists. After updating the unique values repository, the replace rules are applied on the dataset as well. If the update finishes with success, a new version of the dataset will be created and stored in the External Objects collection.

When writing a report or research paper, not all data gathered might be needed to

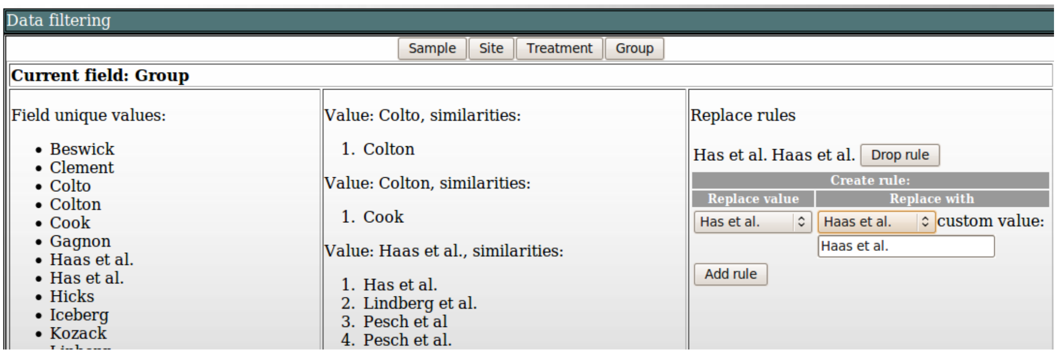


Figure 3.4: Data clean-up for content imported from Excel

be included. The full dataset is useful to work on with the colleagues, but in the work documents just some subsets might be required. In order to allow users to keep all data in one source and use for other purposes just some subsets, we considered relevant a grouping operation. The grouping file-type specific operation will allow users to select from the main dataset data fields (columns) and records (rows) that are relevant for a particular task. For example, if the team needs a chart that contains just some data from three columns that are present in the dataset, they create a set containing just the required data, and in the working document, the group can be referenced as data source for the new chart.

The problem of creating and displaying subsets is one of data representation, hence we considered more appropriate to handle it using XQuery. These operations are defined as web-services and included in the file-type specific operations index. In order to create a new group, the user is asked to provide a name that will identify it, and select the columns (data fields) and rows (data records) that will be included. In order to display the user interface that will allow the group definition, the web service will require the dataset identifier. Based on this identifier, the content is retrieved and an user interface that mimics the editor will be returned.

If the team creates a new version of the Excel file using the MS Excel editor, that is significantly different from the one that is currently loaded in the workspace (e.g. new columns are added, old ones are renamed and data is updated), a new version of the dataset can be created by merging them. The new version of the Excel file must be first imported in the application and using the merge file-type specific operation all new data is added automatically to the existing dataset.

The merging feature is implemented as a web-service and built in XQuery with calls to Java functions included. In order to combine two datasets the degree of similarity between the two is computed. First the similarity between data field names (headers) is computed and second, a similarity between the content from the datasets is determined.

After determining these two types of similarities, a XForms application is returned to the user that will allow him to inspect the results. At this point the user is requested, starting from the similarities presented, to create merge rules. These rules determine which columns are to be merged, and if the two header names are distinct, choose or create a new one.

We consider that the file-type specific operations should be created in languages that provided the richest set of functionality in order to interact with the content. This will lead to an implementation that is based on a great variety of languages that will lead to one significant problem: communication with the main module. This is solved by requiring each feature to deliver its functionality as a web service and provide an API.

3.5.3 Exporting the content

Exporting an external object can be done either to the originating format, or to another content standard. Exporting to other closed source formats (like MS Word) might be useful but currently will not be taken in consideration.

Exporting to a content standard, where appropriate, requires a mapping schema that state elements equivalence. Since this is a transformation from and to XML, following this model, only a transformation schema is required. Transformation schema will be defined in XSLT, which allows content from one format to be transposed into another based on transformation rules.

Exporting to a closed source format will require the use of other programming languages since XSLT and XQuery are XML oriented and such an approach is either deficient or probably impossible. The export steps will be built in a similar manner as the import procedures and defined as well as web-services or integrated in the External Objects component.

3.6 Conclusions

Content interaction manages content life cycle aspect, like creating content objects, instantiating an object and defining rendering and access rules. Our focus was on using specialized technologies that focus on specific sets of actions that can be executed in order to implement aspects concerning the interaction with the content. Using multiple technologies provides as a benefit a specialized tool kit, but demands also an additional binding language.

He have found that using XForms as the main mechanism for providing document representations is more efficient than JavaScript because the latest offers modest support for changes if modifications in content objects structure occur. JavaScript works with the DOM at a low level of abstraction, whereas XForms is focused on handling tree node using basic constructs. Using XForms allows changes to be implemented faster and with less effort since it defines a high-abstraction level.

The content level binding mechanism that integrates tasks performed in different languages has been implemented using a technology that is based on principles from Unix pipelines, namely XProc. This technology allows to interact with content objects through straightforward steps that represent calls to module defined in different XML processing tools. This allowed us to create operations like content representations generation in a clear and easy to use manner.

Having an implementation built using XML tools switches the focus on content objects, and on resources in general, therefore allowing a significant degree of flexibility to be implemented. In our research reaching a high degree of flexibility in managing and interacting with content was one of the main goals. Managing content in XML is a highly effective method since it reduces the number of translation steps required for components to talk one with each other. This approach allows the application to be focused on resources and not on processing steps. On the other hand, building a content management component that stores all data in XML imposes some constraints and raises some challenges, like how to deploy user interface elements, integrate a wide range of XML related technologies or group processes defined in different languages in packages.

Conclusions

The aim of this research was to identify ways in which content management, as a component of a collaborative system, can enhance collaboration. We have conducted our research by following a general framework that had the following goals: *i)* identify what collaboration is and what makes it an efficient process, *ii)* design a content management component can improve collaboration, and *iii)* determine appropriate technical solutions that can be used to implement such a content management component.

Collaboration is about the pursuit of insight and new ideas. It is based on communication, coordination and cooperation, and it is often confused with the latest. Whereas cooperation is about working in a value-chain model, collaboration is about creating shared understanding. Collaboration must be defined in a goal-oriented framework, and requires tools that can manage complexity, opportunities and risks. A flexible content management solution that enables content objects to act as externalizations is required in order to create records of mental processes, and use content objects for critique and negotiation.

Content management is a solution that handles content collection, administration and publishing. Content collection requires the means to import content created using other tools, and providing the means to create new content elements from scratch. Content can be regarded as a compromise between the multitude of uses of data and the richness of information. In order to support collaboration, such a system must be based on intelligent content, that is content not limited to one purpose (is reusable, reconfigurable and adaptable), or one technology and is structurally rich. XML technologies stack provides tools specialized on processing and publishing content that represent an advantageous alternative to current practices.

The required tools for a collaborative system are available to a small extent in current implementations, as they focus primarily on communication and some management aspects. A collaborative system must implement appropriate means to manage goals and their associated working plan, uncertainty mitigation, consensus building with the involvement of all stakeholders and a flexible content management module that allows users to customize how they interact with content according to their needs. Our model covers all the aforementioned requirements and provides a framework for a content management component that allows content to be used in concordance with tasks domain specifics. Current content management solutions focus either on web content, documents or enterprise content, whereas our model is focused on knowledge workers that have to create new understating and artifacts. Our content management framework is divided in several components that handle all required functionality: content Interaction (responsible with creating content representations, content level workflow, storing and versioning), external objects (managing and interacting with content from closed source formats) and project workflow (coordinate content interaction in accordance with the collaborative systems management module).

Implementing such a system requires technologies that can allow content to be transformed and delivered with exceptional flexibility. XForms has been found to be a better solution than JavaScript for delivering content representations since it provides a greater level of abstraction when working with content elements. On the other hand, JavaScript can be used successfully to compensate for some of XForms user interface limitations. In order to manage graphical content representations (e.g. charts) SVG is an acceptable solution since it can be used both on the client and for the printed versions of the content, avoiding thus having different content representations on print and web. It is inappropriate to consider XSLT and XQuery competing technologies, instead these tools should be used together because they have different approaches to processing content that can supplement each others functionality. Our model requires combining content and metadata in order to deliver content representations. XProc has been identified as suitable solution because it is based on a pipeline style and is built in order to allow other technologies from the XML stack to interact. In order to allow users to interact with highly divers types of content, it is required that a component that manages this interaction is provided. This component must define a general framework that allows plug-ins to be defined in languages that are best suited for a specific format so that content can be manipulated with minimal effort and the result easily integrated in the framework.

Bibliography

[1] D. M. Osher, "Creating Comprehensive and Collaborative Systems," *Journal of Child & Family Studies*, vol. 11, no. 1, pp. 91–99, 2002.

[2] L. M. Camarinha-Matos and H. Afsarmanesh, *Collaborative Networked Organizations*, 2004, ch. Support Infrastructures for New Collaborative Forms, pp. 175–192.

[3] S. I. Nitchi, A. Mihaila, and M. I. Podean, "Collaboration and Virtualization in Large Information Systems Project," *Informatica Economica*, vol. 13, no. 2, pp. 12–19, 2009.

[4] B. Plimmer and M. Apperley, "Making paperless work," in *CHINZ 07: Proceedings of the 7th ACM SIGCHI New Zealand chapters international conference on Computer-human interaction*. New York, NY, USA: ACM, 2007, pp. 1–8.

[5] D. A. Marca, "E-Business Innovation - Surviving the Coming Decades," in *SECRYPT*, ser. SECRYPT 2008, Proceedings of the International Conference on Security and Cryptography, Porto, Portugal, July 26-29, SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications, E. Fernndez-Medina, M. Malek, and J. Hernando, Eds. INSTICC Press, 2008, pp. 5–16.

[6] 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.

[7] D. C. Engelbart, "Toward augmenting the human intellect and boosting our collective IQ," *Commun. ACM*, vol. 38, no. 8, pp. 30–32, August 1995. [Online]. Available: http://doi.acm.org/10.1145/208344.208352

[8] K. A. Hribernik, K.-D. Thoben, and M. Nilsson, *Encyclopedia of E-collaboration*. Hershey, PA: Information Science Reference - Imprint of: IGI Publishing, 2008, ch. A Generic Definition of Collaborative Working Environments, pp. 308–313.

[9] L. M. Camarinha-Matos and H. Afsarmanesh, *Collaborative Networks: Reference Modeling*. Springer US, 2008, ch. Collaboration forms, pp. 51–66.

[10] M. Schrage, *Shared Minds: The New Technologies of Collaboration*. Random House, 1990.

[11] 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.

[12] 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.h%tm

[13] M. Csikszentmihalyi, *Creativity: Flow and the Psychology of Discovery and Invention*. New York, USA: Harper Perennial, 1997.

[14] U. Farooq, J. M. Carroll, and C. H. Ganoe, "Supporting creativity in distributed scientific communities," in *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, ser. GROUP 05. New York, NY, USA: ACM, 2005, pp. 217–226. [Online]. Available: http://doi.acm.org/10.1145/1099203.1099242

[15] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York: Harper Perennial Modern Classics, 2008.

[16] 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.

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

[18] A. Rockley, P. Kostur, and S. Manning, *Managing Enterprise Content: A Unified Content Strategy*, 1st ed. Indianapolis, Indiana: Pearson Education, 2003.

[19] Rockley Group, "Managing Enterprise Content: A Unified Content Strategy," The Rockley Group Inc., Markham, ECM UCS Whitepaper, 2003.

[20] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data - The Story So Far," *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.

[21] B.-L. Tim, "Linked Data - Design Issues," 2009, Last Accessed: December 2010. [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html

[22] M. I. Podean and L. Rusu, "An XML-based Approach for Content Management in Virtual Organisations," in *ICE-B*, ser. ICE-B 2009 - Proceedings of the International Conference on e-Business, Milan, Italy, July 7-10, ICE-B is part of ICETE - The International Joint Conference on e-Business and Telecommunications, J. Filipe, D. A. Marca, B. Shishkov, and M. v. Sinderen, Eds. INSTICC Press, 2009, pp. 252–257.

[23] H.-C. Chu, M.-Y. Chen, and Y.-M. Chen, "A semantic-based approach to content abstraction and annotation for content management," *Expert Systems with Applications*, vol. 36, no. 2, Part 1, pp. 2360–2376, 2009.

[24] C. Frank and M. Gardoni, "Information content management with shared ontologiesat corporate research centre of EADS," *International Journal of Information Management*, vol. 25, no. 1, pp. 55–70, 2005.

[25] L. Richardson and S. Ruby, *RESTful web services*. Sebastopol, CA: OReilly, 2007.

[26] A. Subbu, *RESTful Web Services Cookbook*, First ed., T. Mary E., Ed. Sebastopol, CA: O'Reilly Media, Inc. & Yahoo! Press, 2010.

[27] A. T. Bahill and R. Botta, "Fundamental Principles of Good System Design." *Engineering Management Journal*, vol. 20, no. 4, pp. 9–17, 2008.

[28] X. Xu, L. Zhu, Y. Liu, and M. Staples, "Resource-oriented business process modeling for ultra-large-scale systems," in *ULSSIS 08: Proceedings of the 2nd international workshop on Ultra-large-scale software-intensive systems*. New York, NY, USA: ACM, 2008, pp. 65–68.

[29] C. Pautasso and E. Wilde, "Why is the Web Loosely Coupled?: A Multi-Faceted Metric for Service Design," in *Proceedings of the 18th International World Wide Web Conference (WWW2009)*, Madrid, Spain, 2009, pp. 911–920.

[30] M. I. Podean, D. Benta, and L. Rusu, "About Creativity in Collaborative Systems. Why it matters and how it can be achieved," in *ICE-B 2011 - Proceedings of the International Conference on e-Business, Seville, Spain, July 18-21*, D. Marca, B. Shishkov, and M. van Sinderen, Eds. INSTICC Press, 2011, pp. 151–154.

[31] K. Duffy, "XML-Based Content Management - Control is King." *AIIM E-DOC*, vol. 21, no. 3, pp. 48–49, 2007.

[32] S. Whittemore, "Metadata and Memory: Lessons from the Canon of Memoria for the Design of Content Management Systems." *Technical Communication Quarterly*, vol. 17, no. 1, pp. 88–109, 2008.

[33] P. Walmsley, *XQuery*, S. Simon, Ed. Sebastopol, CA: O'Reilly Media, Inc., 2007.

[34] T. Jeni, *Beginning XSLT 2.0: From Novice to Professional*, ser. The Expert's Voice in XML Technology , M. Chris, Ed. Berkeley, CA: Apress, 2005.

[35] W. Norman, M. Alex, and T. Henry S., "XProc: An XML Pipeline Language," W3C Recommendation 11 May 2010, Last Accessed: April 2011. [Online]. Available: http://www.w3.org/TR/xproc/

[36] v. d. V. Eric, *XML Schema*, First ed., S. Simon, Ed. Sebastopol, CA: O'Reilly, 2002.

[37] J. M. Boyer, "XForms 1.1, World Wide Web Consortium (W3C) Recommendation 20 October 2009," Last Accessed: March 2011. [Online]. Available: http://www.w3.org/TR/xforms

[38] M. Kaufmann and D. Kossmann, "Developing an Enterprise Web Application in XQuery," in *ICWE 9: Proceedings of the 9th International Conference on Web Engineering*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 465–468.

[39] M. I. Podean, "The Role of XML Forms in Content Management," *Annals of the TiberiuPopoviciu, Supplement of International Workshopon Collaborative Systems andInformation Society*, pp. 333–340, 2008.

[40] D. Erik, D. Patrick, G. Anthony, L. Chris, M. Cameron, S. Doug, and W. Jonathan, "Scalable Vector Graphics (SVG) 1.1 (Second Edition)," W3C Working Draft 12 May 2011, Last Accessed: June 2011. [Online]. Available: http://www.w3.org/TR/SVG/

[41] A. Couthures, "Dynamic SVG graphs for XForms. From business data to high quality graphs within client-side MVC architecture," in *Proceedings of the 8th International Conference on Scalable Vector Graphics* , Paris, France, 2010.

[42] L. Rusu, M. I. Podean, S. Szabolcs, and R. Costin, "A hierachical model for medical registrations," *Journal of Information Systems and Operations Management*, vol. 4, no. 2, pp. 32–42, 2010.

[43] M. Gilleland, "Levenshtein Distance, in Three Flavors," Last Accessed: October 2010. [Online]. Available: http://www.merriampark.com/ld.htm