A Life Cycle Model of XML Documents

Airi Salminen*
Department of Computer Science and Information Systems
University of Jyväskylä, FI-40014
P.O. Box 35 (Agora)
Jyväskylä, Finland
phone: +358 50 518 6284
fax: +358 14 260 3011
email: airi.salminen@jyu.fi

Reija Nurmeksela
Tieto Finland Oy
P.O. Box 163
FI-40101 Jyväskylä, Finland
phone: +358 20 7257618
fax: +358 20 5496955
email: reija.nurmeksela@tieto.com

Eliisa Jauhiainen
Department of Computer Science and Information Systems
University of Jyväskylä, FI-40014
P.O. Box 35 (Agora)
Jyväskylä, Finland
phone: +358 40 8053094
fax: +358 14 260 3011
email: eliisa.jauhiainen@jyu.fi

* corresponding author

A Life Cycle Model of XML Documents

Electronic documents produced in business processes are valuable information resources for organizations. In many cases they have to be accessible long after the life of the business processes or information systems where they have been created. To improve the management and preservation of documents, organizations are deploying the Extensible Markup Language (XML) as a standardized format for the documents. The goal of the paper is to increase understanding of XML document management and provide a framework to enable the analysis and description of the management of XML documents through their life. We have followed the design science approach. We introduce a document life cycle model consisting of five phases. For each of the phases we describe the typical activities related to the management of XML documents. Furthermore, we also identify the typical actors, systems, and types of content items concerned in the activities of the phases. We demonstrate the use of the model in two case studies: one concerning the State Budget Proposal of the Finnish Government and the other concerning the Faculty Council Meeting Agenda at a university.

Introduction

A great deal of the information resources in organizations consists of documents produced in business processes. Documents serve a number of different purposes, for example, as tools for communication and decision-making. They also serve as recordings of business activities and have to live, in many environments, through generations of

technologies, systems, users, and surrounding organizations (e.g., Volonino, Sipior, & Ward, 2007; Borglund, 2008). Today Extensible Markup Language (XML) has become the *lingua franca* of the data interchange on the Internet and its use is becoming increasingly widespread also for representing documents produced in business processes. In some organizations XML is adopted simply as a document format of an office application. Standard XML-based, open formats for office documents are ODF (OpenDocument Format for Office Applications, ISO/IEC 26300:2006) and OOXML (Office Open XML File Formats, ISO/IEC 29500-1:2008). Today many public domain organizations have published policies to support or enforce the use of open document formats. Examples of them are the government of Norway (Ministry of Government Administration, Reform and Church Affairs, 2007) and the state government of Massachusetts in the United States (Shah, Kesa, & Kennis, 2008). An important motivation for the adoption of open file formats is to achieve vendor-independency as stated by Cerri and Fuggetta (2007):

> The technology supplier cannot claim any right on the customers' data and information or impose limitations and constraints on their manipulation. *The customer must have the true possibility to switch to another supplier and to access its own information without being anyhow limited*. (p. 1936)

Especially in public sector open standards are also an important means to release data as machine readable open data to support transparency and data sharing (see e.g. Peled, 2011).

When using an XML-based format of an office application document instances are encoded as marked-up text where the markup shows the structures identified by the office

application (like titles, paragraphs, and lists). This is not always sufficient: there may be a need to define a domain-specific markup language to incorporate semantic information in document markup. The definition capabilities related to XML are available for the purpose. Semantic markup provides several possibilies, including improved accessibility, reusability, and information integration (e.g., Bernstein & Haas, 2008; Salminen & Tompa, 2011). An open file format together with semantic markup has also been seen as a way to improve the persistence of information through time (Brooks, 2001).

Since the publication of XML as a W3C (World Wide Web Consortium) standard in 1998, the development and research related to XML has been extremely active. In Google Scholar the search term XML provides over 2,000,000 results. A great deal of the practical development concerns software development and development of XML-based standards for particular application domains. The emphasis in the academic research has been in the technologies used for processing and retrieving XML data. Research considering the management of XML documents over their life, including the life after the active processing and use of documents, is rare. This concerns document management research also more generally. Respectively, in the information management of organizations there often seems to be a gap between the active and passive life of documents (Barker, Cobb, & Karcher, 2009). Our goal in the paper is to provide a model to enable the analysis and description of XML document management over the whole life of documents. Another goal of the paper is to increase understanding of XML document management in organizations.

Our research approach is design science. Design science has been widely used in many research disciplines, especially in engineering and computer science but also in

information science and information systems. Design science attempts to create artifacts that serve human purposes (March & Smith, 1995). March and Smith (1995) have divided the artifacts into four kinds: constructs, models, methods, and instantiations. In our case the developed artifact is the life cycle model of XML documents. We demonstrate the use of the model in two different cases: one concerning the State Budget Proposal of the Finnish Government and the other concerning the Faculty Council Meeting Agenda in a university faculty. The case descriptions can also be seen as expository instantiations used to test and help explaining the model (Gregor & Jones, 2007).

Peffers, Tuunanen, Rothenberger, and Chatterjee (2008) have derived from a number of prior design science methodologies (e.g. Nunamaker, Chen, & Purdin, 1990; Hevner, March, & Park, 2004; Gregor & Jones, 2007) a nominal sequence of six steps in design science research (DSR) process: (1) problem identification and motivation, (2) defining the objectives for a solution, (3) design and development of the artifact, (4) demonstration, (5) evaluation, and finally (6) communication. The steps also provide a template for the structure of research outputs. Figure 1 depicts how we have used the process to structure the paper. The figure also shows techniques that we have adopted for the research. Outputs of the communication phase are provided in the paper.

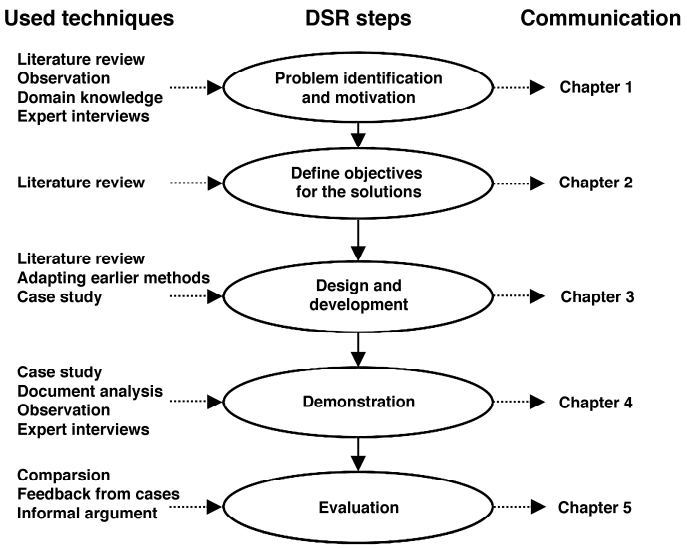

FIG. 1. The design science research steps of the study.

In our case, the design of the new artifact continues the earlier RASKE methodology development. RASKE methods and use cases have been communicated to researchers in scientific articles (e.g. Järvenpää, Virtanen, & Salminen, 2006; Salminen, Kauppinen, & Lehtovaara, 1997; Salminen, Lyytikäinen, & Tiitinen, 2000; Salminen, 2005; Salminen, Nurmeksela, Lehtinen, Lyytikäinen, Mustajärvi, 2008; Salminen, 2010; Tiitinen, Lyytikäinen, Päivärinta, & Salminen, 2000) and to practitioners in project reports, seminars and workshops. Before introducing the life cycle model we discuss the concepts and methodologies related to XML document management. One of the methodologies introduced is the RASKE methodology.

XML Document Management

This section provides the background intended to support understanding of the life cycle model and related case examples later in the paper. We first introduce the main

concepts of XML document management. Then we briefly describe and compare methods that have been proposed for the analysis and description of XML document management. Finally we introduce the core components of an XML document management environment.

Concepts

The term *document management* is used to refer to the creation, storage, organization, transmission, retrieval, manipulation, update, and eventual disposition of documents in organizational context, to fulfill organizational purposes (Sprague, 1995). *XML document management* refers to the management of documents in XML format. XML documents are *structured documents* where a document instance is described as a hierarchic structure of named parts (Bray, Paoli, & Sperberg-McQueen, 1998). The parts are explicitly indicated by systematic markup that enables applications to identify, retrieve, and process those parts, to some extent in a similar manner as data in databases.

Document management can be regarded as a special kind of *enterprise content management* (ECM). Smith and McKeen (2003, p. 648) define the term as "the strategies, tools, processes and skills an organization needs to manage all its information assets (regardless of type) over their life cycle". A term closely related to document management is records management. In the records management standard 15489 (ISO 15489-1, 2001, p. 3), a *record* is defined as "information created, received, and maintained as evidence and information by an organization or person, in pursuance of legal obligations or in the transaction of business" and *records management* as "field of management responsible for the efficient and systematic control of the creation, receipt, maintenance, use and disposition of records, including processes for capturing and maintaining

evidence of and information about business activities and transactions in the form of records". In contrast to document and content management systems that are primarily intended to support on-going business processes involving editing or versioning of content, records management systems are primarily intended to provide secure repository of authentic records (DLM Forum Foundation, 2011). A record captured into a records management system may consist of one or more documents (ISO 15489-1, 2001) or of other kinds of components. In records management systems modifications of records are prevented or strictly controlled whereas in content or document management systems updates and versioning are typical operations. In this paper our focus is in XML documents that require their management and preservation as records.

Similarly to databases, the structure and other constraints for a class of XML documents on a domain is described by a *schema*. The schema defines a markup language for the domain. A great number of schemas have been defined for specific sectors or application domains. For example, the earlier mentioned ODF and OOXML for office applications, HL7 standards for the health sector (www.hl7.org), and the Universal Business Language (UBL, www.oasis-open.org) for business documents. Organizations sometimes adopt a pre-defined schema like ODF or OOXML as such for documents produced in their processes. In other times they define their own schemas either from a generic schema earlier developed by a standardization organization, for example, from HL7 or UBL, or from the very beginning.

The layout of XML documents on an output medium is usually defined by means of *style sheets*. This enables the separation of structure, layout, and content of a document from each other, and processing each of them separately (see Figure 2). There are

different *schema languages* available to define the structure and different *style sheet languages* to define the layout. The figure lists three of the available schema languages and two of the available style sheet languages. From the schema languages DTD (Document Type Definition) is defined in the XML specification. The style sheet languages CSS and XSL, as well as the schema language XML Schema, have been developed and published by W3C. RELAX NG is the schema language developed by Clark and Murata through OASIS (Advancing open standards for the information society; Clark & Murata, 2001).

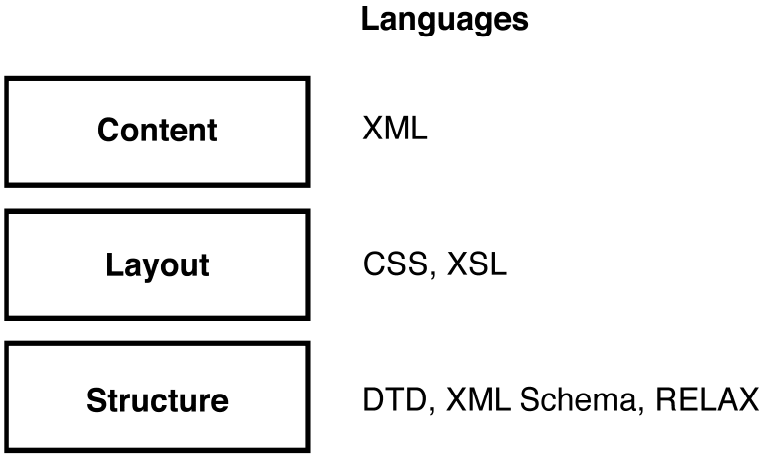

FIG. 2. Three facets of an XML document.

From the point of view of XML document management it is important to realize that XML documents actually have two structures at the same time: logical structure and physical structure. The core components of the *logical structure* are *elements* indicated in the marked-up document by tags of the form <...> and </...>. The *physical structure* consists of storage units called *entities*. The entities of a document are either separate files or named pieces of text reused in different places. The entities are connected to each other by entity references. The management of XML documents requires the management of both structures, logical and physical.

Systematic management of any kind of content items requires *metadata*. Metadata may be related to collections of items or to individual items (Gilliland, 2008). In the first case we use in this paper the term *class metadata*, in the latter *instance metadata*. Instance metadata is often regarded as a surrogate of the instance like, for example, a library record for a publication is a surrogate of the publication (Greenberg, 2010). Examples of class metadata in an XML document management environment are schemas specifying the structural and other constraints for a class of documents, or *ontologies* defining concepts and their relationships to be used, for example, for annotating documents. In fact, also schemas define ontologies for the structural components of documents. Instance metadata attached to an XML document might be described, for example, by using Dublin Core (DC) standard (Dublin Core Metadata Initiative, 2010) and embedded in documents or stored externally, possibly in XML format (Powell & Johnston, 2003).

Methodologies

Methods and techniques from various information, information systems, and business process analysis methodologies can be used to analyze and describe XML document management. Since XML was derived by a number of restrictions from the older meta markup language SGML (Standard Generalized Markup Language; Goldfarb, 1990), the methods originally developed for SGML are applicable also in XML environments. The life cycle model of this paper extends the RASKE methodology that has been gradually created and extended in real-life development projects. In the following we briefly describe and compare RASKE and three other widely published methodologies that have been introduced and used to support the deployment of SGML or XML. The

methodology of Maler and El Andaloussi (1996) is included in the comparison because it was the pioneering methodology for the design of SGML documents and some of its methods have also been adopted as a part of the RASKE methodology. From the later methodologies Unified Content Strategy (Rockley, Kostur, & Manning, 2003) and the Document Engineering approach (Glushko & McGrath, 2005) have been chosen because, similarily to RASKE, they consider documents in a wider context of work processes, not only as structured information content. The comparison of the methodologies is summarized in Table 1. In addition to the aforementioned methodologies used especially for the development of structured document solutions, we have included in our comparison Dirks, guidelines for the management of digital records (State Records Authority of New South Wales, 2007). It represents methodologies especially developed for records management. In the following we briefly describe some characteristic features and core concepts of each of the five methodologies.

TABLE 1. Summary of the five methodologies.

	Primary application domain	Types of documents	Document format	Modeling concerns
RASKE	electronic document management in organizations	narrative	SGML/XML recommended	organizational environment, business and work processes, roles, documents, metadata
Maler&ElAndaloussi	publishing industry & technical documentation	narrative	SGML	document and component structures
Unified Content Strategy	content management and multichannel publishing	narrative	XML recommended	content life cycles, element structures, reuse, documents, workflows, content management processes

Document Engineering	automated business processes	narrative/transactional	XML recommended	business processes, documents, components
Dirks	records management	narrative/transactional	various formats, XML recommended for archival format	modeling is not explicitly included in the methodology

The RASKE methodology was started in 1994 in a joint project of researchers, Finnish Parliament, some ministries, and a software company (Salminen, Kauppinen, Lehtovaara, 1997). The name RASKE was adopted for the project from the Finnish words "Rakenteisten AsiakirjaStandardien KEhittäminen" meaning the development of standards for structured documents. A practical objective in the project was to find means for the management of Finnish parliamentary documents in a standard format. The first RASKE project was followed by long-term collaboration with the Finnish Parliament and ministries. As a result SGML and later XML was widely deployed for the management of Finnish parliamentary documents.

The emphasis in the methodology development from the beginning has been in the holistic analysis of document/content management environments where content items are in the organizational context of business processes. We have not made a separation of the terms "record" and "document", or the corresponding Finnish terms "asiakirja" and "dokumentti", respectively. In the development cases we have been dealing with documents that have to be stored as records of some process activities. The RASKE methodology includes methods for data gathering, requirements analysis, modeling, evaluation, and documentation. It has adopted and adapted some object-oriented modeling methods for process modeling, document modeling, and role modeling. From the process models the most extensively used in the methodology have been the *input* and

output models showing the resources used or produced in business processes (Salminen, Lyytikäinen, Tiitinen, 2000). For example, the *document output model* shows the documents created in a process. *Document modeling* is divided into object, state, and content modeling. RASKE methods have been used also in some development projects where the goal has not been the deployment of SGML/XML for documents. For example, in a project the methods were used and adapted to support the integration of information resources by means of metadata standardization. A special focus in the project was in the needs of the Finnish legislative work and adoption of semantic web technologies (http://www.it.jyu.fi/raske/index_en.html). Metadata was modeled by means of RDF (Resource Description Framework).

The methodology of Maler and El Andaloussi (1996) has emphasis on the design of SGML DTDs providing means for the DTD project management, document type needs analysis, document type design, DTD development, validation, and documentation. The tree structures of documents are modeled by *elm diagrams*. They have been widely adopted and adapted in SGML/XML syntax-oriented editors and also in other methodologies for describing SGML/XML structures. Also in RASKE application cases elm diagrams have been used in content modeling for graphical descriptions of hierarchic structures.

The fundamental concept in the Unified Content Strategy is *reuse*. The main purpose is to avoid content “silos” by effective content reuse. The implementation of the strategy is divided into eight phases starting with analysis, design, and selecting tools and technologies (e.g. XML). The analysis phase includes the analysis of content life cycle processes but no particular process modeling technique is suggested. The most common

life cycle is considered to extend from creation to delivery; problems related to content archival and retention are outside the scope of the methodology.

While the previous methodologies have been developed to support the management of narrative documents that are primarily intended for human use and human communication, the Document Engineering Approach provides concepts and methods to design effective business transactions and Web services by means of documents, and in particular, by means of XML documents. The central concept of *model matrix* is used as a roadmap to advance from models at different levels of abstraction and granularity to effective implementations of transactions. No modeling technique is inherently required in the approach but the use of XML-encoded implementation models is emphasized. Similarly to RASKE, documents are regarded as inputs and outputs of business processes. RASKE, however, considers process models as descriptive tools to support human communication and understanding, not as prescriptive tools to enable automated business processes. The problems related to document publishing, archival, and retention are left outside the scope of Document Engineering.

Dirks methodology was especially designed for building good recordkeeping systems into Australian Government agencies. In the methodology a recordkeeping system may be a separate information system but often it is (or should be) embedded in a business information system, in order to keep records of business transactions. The methodology includes a great number of guidelines, for example, for capturing, storing, and securing digital records, creating metadata about them, determining how long to keep them, enabling business continuity also in disaster situations, preserving digital records for long term, and finally disposing them. Dirks also defines a process of eight steps for

designing and implementing recordkeeping systems. In contrast to the previous four methodologies, Dirks does not provide methods and techniques for analyzing and designing documents. The use of open standards for records is introduced as a possible design strategy and XML in particular as a possible archival format to which records are converted and then packaged together with related metadata.

XML Document Management Environment

In this section we describe the components of an XML document management environment using the concepts and models that we have earlier introduced in the RASKE methodology. Having well-defined concepts and methods for the description of the environment provides the basis for document life cycle descriptions.

The main components of a document management environment can be divided into activities and resources (see Figure 3). In the figure the activities are depicted by the oval, resources by rectangles, and information flow between resources and activities by arrows. An *activity* is a set of actions performed by one or more actors in a work process. An activity can be divided into smaller activities and described by means of a *process model* and instantiated in a *process case*. The resources are of three types: actors, systems, and content items. An *actor* is an organization, a person, or a software agent. A *system* is a tool that is used to support the performance of activities. Various kinds of systems are needed in computerized environments, including hardware, software, networks, standards, and mandates. Mandates can be, for example, regulations and legislation governing the document management of the domain. *Content items* are documents and other units of stored data accessible as meaningful pieces of information by means of systems. All resources are information repositories in which the information produced in

an activity can be stored or from which information can be taken and used in an activity. Thus information is not stored only in documents and other content items but also in the heads and experience of people, in the organizational culture, and in systems.

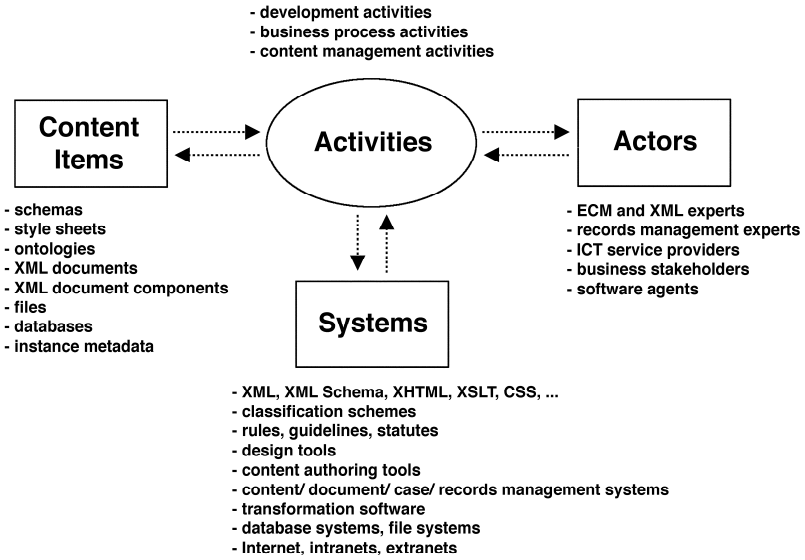

FIG. 3. An XML document management environment.

By analyzing documented cases (e.g. Aversano, Canfora, de Lucia, & Gallucci, 2002; Broberg, 2004; Sartor, Palmirani, Francesconi, & Biasiotti, 2011; Sohn, Ko, Lee, Kim, Lim, & Choy, 2002) as well as using the information gathered by observations and interviews from a number of XML deployment cases (e.g. Nurmeksela, Jauhiainen, Salminen, & Honkaranta, 2007; Salminen et al., 2004), we have identified the typical components in an XML document management environment. The *activities* may be divided into three categories: development activities, business process activities, and content management activities. The content management activities include the creation and update of documents and related metadata, records management activities, and archival activities. *Actors* include organizations and experts needed in the three kinds of activities.

The most typical *content items* accessible from the data repository of an XML document management environment are XML documents, their schemas and style sheets. The components of the physical structure of documents are stored as files or in a database. The types of metadata resources vary: in some environments all instance metadata associated with XML documents is embedded in documents and in their file names, in others the instance metadata is stored externally, for example, as Dublin Core metadata in self-contained XML files. In simplest cases the structural concepts expressed in schemas are the only ontologies managed systematically as content items. Some environments maintain term dictionaries, some others more complex ontologies. Also these ontologies can be represented in XML format, for example, using the XML syntax of RDF (Resource Description Framework). Information from the content items that are stored in open formats should be accessible by several systems, not only by those that are available in the environment at the time the items have been created.

The *systems* needed in an XML document management environment include a great number of standards besides XML, for example, XML Schema for schemas, XSLT for transformations, HTML and XHTML for Web pages, and CSS for style sheets. In many cases also sectoral standards like Open Document Format (ISO/IEC 26300, 2006) or Office Open XML (ISO/IEC 29500, 2008) for office documents, or ebXML (www.ebxml.org) standards for business documents are needed. In all organizations there also are rules and regulations concerning the document management. Some of the rules are produced internally, some of them are external rules expressed, for example, in the legislation of the country or as best practices of the industry. Classification schemes are needed for organizing content units. Also specification of access rights may require

classification schemes. Especially in the case of inter-organizational business processes, the number of different software systems involved in a process case may be great, including different authoring tools, transformation software, content or document management systems, database systems, workflow systems, case management systems, and records management systems.

XML Document Life Cycle

Since documents produced in a business process have to be accessible longer than the life of the process, it is important to consider the special features of XML document management from the point of view their own life, not only their role in the business process. In this section we introduce a document life cycle model of five phases. We describe the typical activities related to the management of XML documents in each phase. Furthermore, we identify the typical actors, systems, and types of content items concerned in the activities of the phase. The model utilizes the concepts that we have earlier introduced in the RASKE methods.

In Figure 4 we have used the RASKE process modeling technique to depict the document life cycle activities. The major content units produced in the activities are shown on the right. Conceptually the units are stored in a repository accessible by different systems. In practice, however, a number of different repositories are often in use and different software systems have access to different repositories during the process. The activities in the figure are connected to each other with *weak control flow* meaning that a control flow arrow (solid line) from an activity A to activity B indicates that activity B typically starts after activity A has started. The modeling technique allows

iteration and parallel activities without showing them explicitly. For example, the design activities may continue parallel with all other activities.

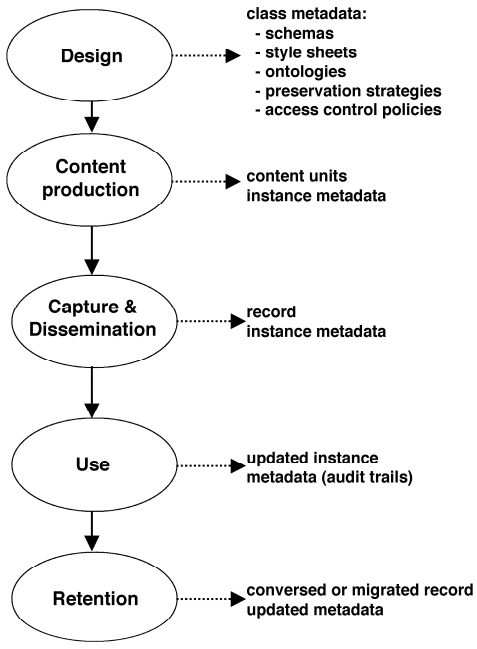

FIG. 4. Activities of an XML document life cycle.

The *design* activities produce the class metadata for the document repository, including schemas for documents and metadata, style sheets, and ontologies. Also preservation strategies and access control policies may be formally defined and stored as content units. *Content production* activities produce the content units from which documents are constructed. The content units are stored with related metadata. The *capture and dissemination* activities refer to the capture of documents as records, making them accessible for use by some system, and publishing the documents for use. The *use* activities refer to the access of documents and information from documents. The *retention* activities are intended to maintain the accessibility of documents and information in them, as well as their integrity and authenticity as records. Retention also

includes the possible disposal of documents. Below each of the activities is described in more detail, emphasizing the XML-related features in all of the phases.

Design

The design includes the planning, implementation, and maintenance of the document management solutions for the environment. The schema design lies in the core of the design activities. Besides schemas, the design concerns the external layout of documents, tools to be used (e.g. for document authoring, distribution, and exchange), business and content management processes, responsibilities, the ways people and organizations use documents to collaborate and communicate with each other, as well as the ways various software systems communicate with each other. The design also concerns ontologies and metadata to be associated with documents. All methods briefly introduced earlier are applicable for the design activities. Additional methods or methodologies may be adopted to support special design areas like enterprise architecture (Liu, Wang, & Quan, 2009), access control (Bertino & Ferrari, 2002; Bhatti, Ghafoor, Bertino, & Joshi, 2005; Bertino, Ferrari, Paci, & Provenza, 2007; Kundu & Bertino, 2008), security (Bernard, 2007), records management (DLM Forum Foundation, 2011), ontology (Noy & Hafner, 1997), or digital preservation (Stanescu, 2005; Runadotter, Mörtberg, & Mirijamdotter, 2011).

A great number of various systems may be involved in the design, both as enablers and constrainers. Operational software systems in use in the business processes often set critical constraints to the design. The new solutions may be implemented as changes in the old software or new software may be built or obtained. The design concerns all phases of the document life cycle and therefore diverse expertise is needed in the design.

Design of XML document management solutions is a continuous process typically including a great number of different tasks after the first implementations, for example, the design of XML transformations for multiple-channel publishing and planning schema, style sheet, or ontology updates. XML schemas evolve over time and this evolution should be taken into account already in the original schema design (Genevès, Layaïda, & Quint, 2011).

Content production

In content production the main purpose is to create, maintain, and store the content units that are used as components in XML documents. In some environments the content is entered directly to an XML document. At the content production stage content items are regarded as editable data, be they XML documents, files in various formats, or data in a database. Important information repositories for XML documents are databases that are created and maintained in various operational systems. In some cases large amounts of data are collected by migration from legacy content repositories, for example, from documents in pdf, SGML, HTML, or proprietary formats (e.g. Reuben, 2003). The transformation into XML format is a semi-automated process where a human expert of the content solves the problems that cannot be handled automatically.

The creation and editing of a content unit may be performed using any kind of authoring system like, for example, an office system or a syntax-directed editor. In the last case, the validity of the content is checked against the schema at the time of content authoring. A workflow system designed especially for collaborative authoring of XML documents may enable, for example, adding multiple digital signatures to the document as fragment signatures (Brooke, Paige, & Power, 2010). In some cases the content units

are stored simply in a file system. A great number of more advanced systems are however available. They can be roughly divided into three categories: content management systems, XML-enabled database systems, and native XML database systems. Characteristic features of these three kinds of systems have been described in (Salminen & Tompa, 2011). Research on version management of XML data has produced a number of articles during the last years (e.g. Mella, Ferrari, Bertino, & Koglin, 2006; Rönnau & Borghoff, 2009; several articles published in the ACM Document Engineering symposiums during 2005-2011).

Capture and Dissemination

At capture documents become under systematic records management. They are associated with metadata that provides evidence of the creation of the records and enables the access of the documents, information in them, understanding their content in the context of their creation, and their retention. If the content of a record is not stored in the form of an XML document at the time of content authoring, then the XML document is first assembled from units available in the content repository, validated, and then captured. The ISO standard 15489 for records management defines the necessary functions included in the capture of a record (ISO 15489-1, 2001; ISO 15489-2; 2001). The capture may be implemented in the same content management system where the content is produced. Alternatively the documents are registered and stored in a separate system designed especially for records management.

There are various ways to capture documents as records in an XML document management environment. One possibility is that the records management operations as defined in records management standards concern only the publishing format of the XML

document, for example, pdf. In that case, however, the benefits of XML for dissemination, use, and retention are missed. One of the benefits is that XML format enables selective dissemination of document content, according to the stated access control policies (Bertino & Ferrari, 2002). On the other hand, XML as an open standard format is also suitable for the publication of documents as *open data*, available for software processing in commercial services.

MoReq2010 (DLM Forum Foundation, 2011) specification is a *de facto* industry standard for software systems that manage records. In the MoReq terminology, records are organized in *classes*. A record consists of one or more *components* and is associated with metadata, event history, and access control list. In case of XML documents, documents of the same type might be organized as records in a class. The metadata associated with a document of the class would include schema and style sheet information. The components might consist of the external entities of the document. If different style sheets are in use for different documents of the class, then the style sheets might be stored as components.

MoReq2010 defines for every record, independently of its format, an XML export format. The export in the specified format is mandatory to all MoReq2010-compliant systems. Content in non-XML data formats can be either embedded in the XML export data format, or linked by an URI to the XML data.

Use

The use of XML documents includes location, retrieval, presentation, interpretation, and reuse of the data stored in documents and in the associated metadata. The data access is performed either by human users or by software systems, inside or outside the context

where the content items and documents were created. In commercial enterprises the use of documents is usually strictly controlled by access rights and tracking. Some of the documents created in an organization can however be publicly accessible, as part of the continuingly growing ecosystem of heterogeneous Web data. The use of documents often involves the reuse of their parts in other documents.

Compared to traditional databases and document repositories, XML document repositories are more complex in many respects. For example, the content contains both formal and natural languages, the document structures and tagging in the repositories is often heterogeneous, access needs concern content units on different levels of granularity, and access criteria may concern content, structure, and context alike. This complexity has established XML retrieval as an active field of research and development as demonstrated, for example, in Luk, Leong, Dillon, Chan, Croft, and Allan (2002), Liu, McMahon, and Culley, (2008), Pérez, Berlanga, Aramburu, & Pedersen (2008), Arvola, Kekäläinen, and Junkkari (2011), and Costa, Manco, Ortale, and Ritacco (2013). Theoretical frameworks for the evaluation of XML retrieval have been introduced by Ali, Consens, and Lalmas (2012) and Blanke, Lalmas, and Huibers (2012). In many situations the diversity of repositories potentially containing useful data is great, including data both in XML and non-XML format, institutional repositories and open-access repositories, on local, regional, national, or international level. It is important that the user has capabilities to check the trustworthiness of the data. The Online Computer Library Center and The Center for Research Libraries have jointly developed and published criteria and checklist for the audit and certification of trustworthy repositories (OCLC and CRL, 2007). No

special criteria are expressed for XML repositories but the checklist still provides a tool for evaluation also for those repositories.

Retention

Retention includes activities for maintaining the usability, integrity, and authenticity of documents. Usable documents can be located, retrieved, presented, and interpreted, in spite of the possible changes in the technological and organizational environment. Metadata plays a critical role in retention. The retention policies developed for corporate document repositories must include rules concerning the length of retention. Unless the retention has been defined permanent, the retention activities include disposal according to the disposal schedule.

In XML document management environments there often is a need to change XML schemas. The documents conforming to the old schemas and stored in the document archive do not necessarily conform to the new schemas without changes. In these situations maintaining the accessibility of information possibly requires the transformation of old documents so that they conform to the new schemas and preserve information. Methods for transformations needed by schema evolution have been developed earlier for databases and later for XML documents (e.g. Kwietniewski, Gryz, Hazlewood, & Van Run, 2010). Another alternative is to maintain different schema versions.

The retention activities in an environment may include the transfer of non-XML content items into XML format. The National Archives of Australia (Heslop, Davis, & Wilson, 2002) has described a preservation process where items in any format are transferred into a normalized XML format. In the normalization the core idea is to

preserve the *essence* of the source document, to enable the recreation of the essential performance over time. Thus the preservation policies have to include the specification of the essential features of the source documents. Several other proposals have been made for migration strategies where the obsolescence of file formats has been solved by migration to XML (e.g. van Horik & Roorda, 2011). The source documents being originally XML documents, the preservation policies in some environment might determine schemas belonging to the essence of the documents, in some other environment they might be seen non-essential from the point of view of long-term access. Similar differences may occur concerning style sheets. In some environment the archival of different XML document versions is important for enabling historical queries concerning the evolution of documents and their contents (Wang & Zaniolo, 2008).

Demonstration in Two Cases

In this section two very different XML document management cases are described, both from Finland. We have been involved in both cases mostly as researchers, partly as consultants and teachers. The first case describes a part of a complex information and content management environment having importance at the national level. The other case concerns content management in a Faculty Office. In both cases one central document type of the environment has been chosen for the life cycle description. Our case descriptions are divided into four parts as shown in TABLE 2.

TABLE 2. Components of a case description.

Data gathering methods	Explains the information sources used for the case description.
XML document management	Identifies and describes the core components of the environment where

environment	the life cycle activities of the case take place: systems, actors and content items (see Fig. 3).
Life cycle description	Describes the most important activities, actors, systems, and content items in each of the five phases of the life cycle (see Fig. 4). In an implemented case, the design phase provides some history background, the descriptions of the other phases concern the current situation.
Impact analysis	Consequences of the deployment of XML in the environment are assessed.

The Case of the Finnish State Budget Proposal

The State Budget Proposal of the Finnish Government is prepared for supporting the budgetary process of the Government. Annually one State Budget Proposal and typically 1-5 Supplementary State Budget Proposals are created.

Data gathering methods

Data for the case description was gathered from literal sources, by expert interviews, and by observations done by one of the authors when working as an ECM consultant for the Finnish Parliament. The Internet service for the State Budget Proposal (http://budjetti.vm.fi/) was also utilized.

XML document management environment

The major information producers and users during the life of a State Budget Proposal are illustrated in Figure 5. Next we briefly describe them.

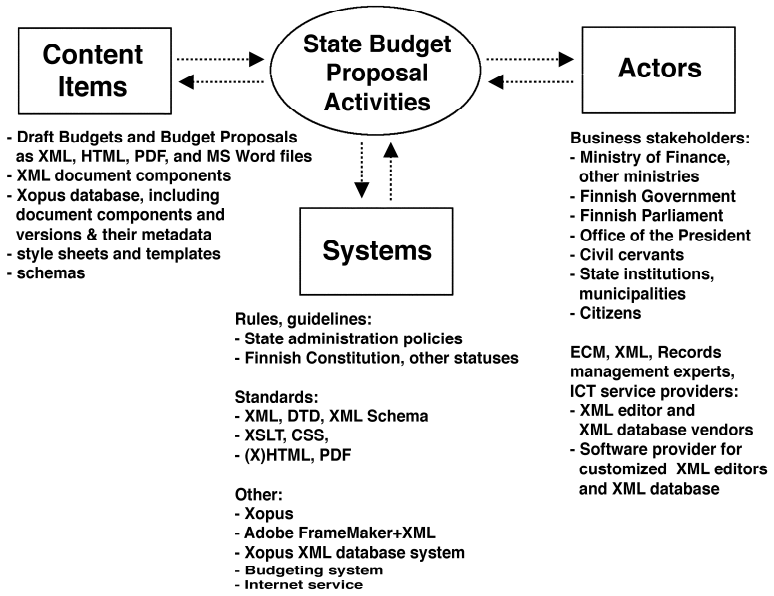

FIG. 5. XML document management environment for the State Budget Proposal.

Systems. The creation of the Budget Proposal is regulated by the Finnish Constitution and Acts. Besides XML, important XML-related standards that have been adopted in the environment. The budgeting system provides access to the financial data and spending limits of the budget. Xopus XML editor is a tool for producing XML content and Adobe FrameMaker+XML for editing the Budget Proposal for printing.

Actors. The most important business stakeholders in the creation of the Budget Proposal are the Finnish Government and Parliament, and the decision-makers there. In the Government, Ministry of Finance has the central role. In the decision-making process the most important user of the Budget Proposal is the Finnish Parliament using it as the basis to create and accept the State Budget.

Contents items. The Budget Proposal consists of a number of separately produced content items. The State Budget Proposal as a whole typically consists of about 900

pages. The XML content is stored in the Xopus XML database, according to the component structure defined in the schema.

The life cycle of the State Budget Proposal

Design. The current solution is based on three major design cycles, the first starting in 1995. The analysis of the content management environment and requirements for the new solution was done in a research project where the first versions of RASKE methods were developed and tested. The project was participated by researchers, representatives from the Finnish Parliament, Ministry of Finance together with a few other ministries, and a software company. Preliminary schemas as SGML DTDs were designed for the major document types of the budgetary process by the researchers of the project. Later selected ICT service providers designed and implemented final SGML DTDs and customized software products for document authoring, visualization, and storage. The first solution concerned mainly the capture and dissemination phase of the document life cycle. The content production of the Draft Budgets remained unstructured, paper copies and text editor (Word 6) files were used for distribution. Transformation into SGML format was done manually in the Ministry of Finance. The shift to structured content production took place in 1998, using Adobe FrameMaker+SGML software.

A few years later, problems in content authoring and managing motivated to start the second design cycle. Important objectives in the redesign included better support for last-minute changes, automated multi-channel publishing, and switch from SGML to XML. The major reason for the change from SGML to XML was better availability of suitable, user friendly XML tools for content production. In the redesign project, an ICT service provider transformed SGML DTD into XML DTD and customized new software

products: Microsoft Word with X4O extension for content production and XHive XML database for capture, dissemination and use. Ministry of Finance continued using Adobe FrameMaker for content authoring. Content production in the renewed environment started in 2003. The earlier produced SGML documents were transformed into the new XML format.

The third design cycle started at the time of the renewal of the budgeting system in 2008. Thus far major financial calculations for the budget had been done in a separate system and the results were copied manually into documents. The aim in the renewal was to integrate budget calculations, content production, and document capture into a single, centralized system. The new system was intended to support collaborative content production in all ministries and in the Office of the President. Furthermore, a goal was to provide new support for the Parliament for using the Budget Proposal. A selected ICT provider redesigned the XML schema and customized new software products: Xopus XML editor for content production and Xopus database for capture, dissemination and use. Adobe FrameMaker+XML was re-customized for preparing the print format. Content production in the redesigned environment started in 2012. The documents produced during 1998-2002 were transformed to conform to the new schema.

As a result of the three design iterations, several XML editors and XML databases have been customized and many transformations have been implemented. From the users' point of view, the most important result is the Internet service. The design has been done in deep collaboration of ECM, XML, records management and business process experts. The foundation of the design work is the XML schema describing the structure of the State Budget Proposal.

Content production. Annually each of the 12 ministries, Office of the President, and the Finnish Parliament create their Draft Budgets entering the data to the budgeting system with the Xopus XML editor. The system allows parallel editing of different parts of the same document by different users. The structure of the document and related metadata is controlled by XML schemas. The content of the document is stored in the Xopus XML database. The related metadata includes information about the actors, document type and status of the content. Metadata is used, for example, to control the access to the content. The Draft Budgets are not available for other ministries until the Ministry of Finance has published the first version of the Budget Proposal.

Capture and dissemination. The Draft Budgets are assembled into a draft version of the State Budget Proposal using the XML content of the Xopus database. The final version is a result of negotiations between the ministries and decisions done in the Cabinet Finance Committee and Government sessions. The versions handled in the budgeting negotiations of the ministries are transformed from the budgetary system into Microsoft Word XML or pdf formats. The final State Budget Proposal is published from the budgeting system on the Internet service in HTML and pdf formats. To the Parliament the document is submitted in the XML format and as printed books. The printed version is edited with Adobe FrameMaker+XML application. The metadata required for the creation of HTML pages and for supporting advanced searches is included in the document already in the content production phase.

Use. A new State Budget Proposal is of interest to state institutions, municipalities and citizens, because it regulates the government funding, taxes and loans for the next year. The Finnish Parliament creates and accepts the State Budget and thus is the main user of the State Budget Proposal. The Parliament captures the Proposal into its own Parliamentary System. Members of the Parliament use the printed book. For processing and reuse, the content of the Proposal is available in the budgeting system. Typically Members of the Parliament accept some part of the Proposal as it is, but propose some changes to the content. These changes are assembled in the Finance Committee into a Finance Committee Report. The content of the Committee Report is partly created by reusing the data of the Proposal. The content of the Proposal is reused again when preparing the final State Budget. The content of the Proposal is later used in the ministries when preparing Supplementary Budget Proposals or a new State Budget Proposal. Advanced search capabilities for these reuse needs have been implemented to the budgetary system and to the Internet service.

Retention. The Budget Proposal is archived permanently in paper format into external diary systems. All Budget Proposals that have been produced in structured format since 2002 are available on the Internet service. The three design iterations have resulted in three different schemas and correspondingly three classes of structured documents, each class originally conforming to its own schema. The selected retention strategy in the case has however been to transform the documents to conform to the latest schema. The transformations have concerned 13 Budget Proposals and about 50 Supplementary

Budget Proposals. Between the major design iterations small changes have been done to schemas: some structures have been added while others have been removed.

Impact analysis

The implementation of the structured approach for the State Budget Proposal has improved effectiveness of the administration: the budget data management, financial calculations and content management activities are integrated into a single budgeting system. The technical content management process for authoring, publishing and delivering of the content is automated. Thus the last minute changes in the decision-making are better managed and copying mistakes in content production are avoided. Delivery of the documents between organizations is done in digital format, which is fast and lowers printing costs. Better availability of the Budget Proposals for citizens and interest groups on the Internet has enhanced transparency of public administration processes. The XML structure and associated metadata enables advanced search capabilities. Furthermore, the XML format can be offered as open data for processing by software systems.

The transfer from SGML to XML has enhanced the utilization of open standards and offered more choices when selecting suitable tools. However, lack of maturity of the tools has caused problems. Implementation of the structured approach in the environment has not been fast and easy: several iterations have been done during last ten years and in each increment more complexity has been added to the environment. The latest design took 3,5 years and the cost for the design and implementation of the unified budgetary system was 1,5 million euros.

The Case of the Faculty Council Meeting Agenda

The Faculty Council meetings of the Faculty of Information Technology at the University of Jyväskylä are called by Meeting Agendas 11-14 times each year.

Data gathering methods

Data for the case description was collected by one of the authors. As a student she had participated in the development project were the XML-based agenda production started. Later she participated as a consultant in the update of the document production system. During these projects, data was collected by analyzing earlier agendas and meeting minutes, by interviewing the office personnel of the faculty, and by prototyping.

XML document management environment

The major information producers and users during the life cycle of a Faculty Council Meeting Agenda are illustrated in Figure 6. Next we briefly describe them.

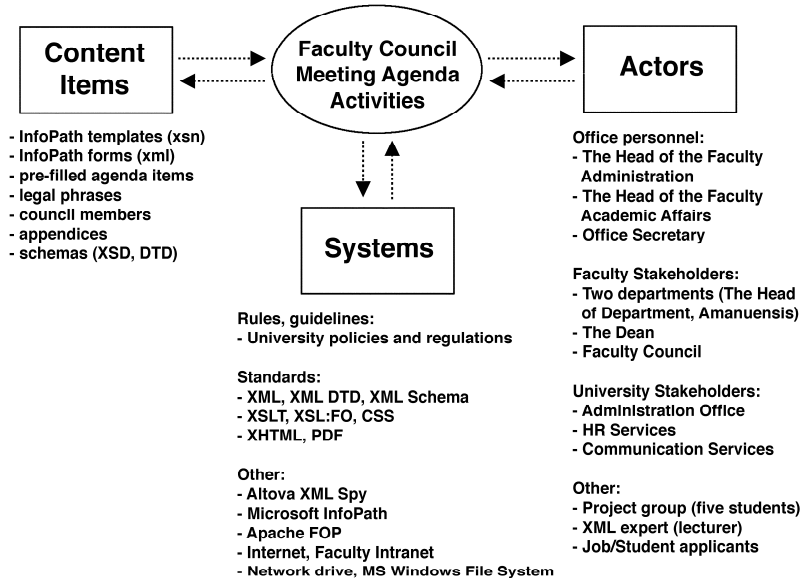

FIG. 6. XML document management environment for the Meeting Agenda.

Systems. The creation of Meeting Agendas is regulated by the rules concerning the decision-making process in the faculties and departments of the university. Besides XML, important standards that have been adopted in the environment are XML DTD, XML Schema, XSLT, CSS, XSL:FO, XHTML, and pdf. From the software systems Altova XMLSpy has been the tool for schema design and Microsoft InfoPath for authoring agendas.

Actors. Student groups and their supervisors have participated in the development of the XML-based solutions. The most important organizational business stakeholders involved in the creation and use of the Meeting Agendas are the two departments of the Faculty, the Faculty Office, the Faculty Council, and the University Administration Office. In the Faculty Office mainly the Head of the Faculty Administration and the Head of the Faculty Academic Affairs prepare the issues for handling.

Content items. In the content repository the schemas, InfoPath templates, InfoPath forms, XSLT transformations, as well as the XSL and CSS style sheets represent class metadata. The agendas and related meeting minutes are stored in the network drive in XML format as InfoPath forms. The folders also include the meeting specific documents transformed from XML into HTML and pdf formats for publishing purposes. Instance metadata related to a meeting agenda is embedded in the agenda. Also properties attached to files and folders, their names in particular, carry instance metadata.

The life cycle of the Faculty Council Meeting Agenda

Design. Problems in document production, publishing, and delivery were the main motivations for starting the design of an XML-based solution for Faculty Council Meeting Agendas in 2004. Since there was knowledge and experience about the use of

the RASKE methodology at the faculty, the methodology was adopted for analyzing and describing the document management environment. No suitable predefined schema was publicly available at the time, therefore a decision was made to design custom schemas for the agenda front page, list of items page, and agenda items. In the design an important objective was to produce minimal changes to content authoring in the Faculty Office environment where MS Word had been the main authoring tool for a long time. Therefore the schemas were designed to be used with MS InfoPath.

The content authoring using InfoPath was facilitated by InfoPath form templates, associated with schemas. Several pre-filled InfoPath forms were created to support content authoring. XSLT transformation definitions and CSS style sheets were designed for HTML and pdf outputs.

The XML-based document production started in 2005. For the first time in the history of the IT Faculty the entire agenda had a coherent layout. In the beginning of 2010, the schemas and InfoPath form templates were modified due to the university re-form. The update required only minor changes in the schemas and templates. In spite of that, the earlier pre-filled InfoPath forms conforming to the earlier template and schema versions could no longer be opened and used with InfoPath. Therefore new pre-filled forms had to be created. The latest upgrade of the solution was carried out in June 2012 together with the update of the MS Office Suite to the 2010 version.

Content production. The two departments of the faculty and the Faculty Office deliver the documents needed for handling in the next meeting. Typically the documents end up in the appendices of the Meeting Agenda as such. Some appendices may be received

from external sources. If paper documents are received, they are scanned and saved as pdf documents. The Head of the Faculty Administration and the Head of the Faculty Academic Affairs, possibly by support of other Faculty Office staff, enter content to the agenda gradually using InfoPath forms. Typically there is a draft version of each agenda item. All forms for items to be included in the agenda of an upcoming meeting are saved in the same folder on a network drive. Some of the text content may be copied from the published minutes of an earlier meeting available on the intranet. The agenda authors use the HTML versions of the archived documents for the needs of copying pieces of content. Legal phrases reoccurring in agenda items are retrieved from an external XML document, which can be modified in InfoPath when needed.

Capture and dissemination. Once the agenda items have been finalized, the separate document files are assembled into one agenda document. The document author carries out this task with the *list of items* form. The content of the form ends up as the second page of the agenda. The assembly includes also the execution of the format conversion to create HTML and pdf outputs of the assembled agenda and adding of reoccurring information on agenda pages automatically without any input from the document author. For example, the name of the document type, date of the meeting, and page numbers are added to each agenda item header as they are on the *list of items* form. The HTML and pdf formats of the agenda are published to the Faculty Council on the intranet, and the list of agenda items is delivered via email to the whole personnel of the faculty. Most of the agenda metadata is available on the agenda front page. It includes organizational and

meeting-specific information. Each agenda item comes with a header section including, for example, the name of the person who has prepared the item for the meeting.

Use. The published agenda is used by the Faculty Council members to support the decision-making in the meeting. After the meeting the Head of the Faculty Administration and the Head of the Faculty Academic Affairs use the agenda to create Meeting Minutes. The XML documents available as InfoPath forms for the agenda are complemented with new content, such as decisions made in the meeting and the names of the faculty council members present in the meeting. The minutes serve as evidence of and information of the decisions made in the meeting. Unlike the agenda, the minutes are public documents and therefore viewable for everyone. Access to the minutes appendices, however, is restricted.

Retention. Long-term archival is guided by the regulations of the university. The official archival records of the Faculty Council meetings are the meeting minutes. The number of people involved in the management of agendas and related documents in the Faculty Office is small. Therefore simple rules concerning file and folder names, together with the rules concerning the metadata embedded in documents have been sufficient. No explicit preservation policies have been defined for the digital content repository. Instead of the earlier repository consisting mostly of MS Word files, the new content repository includes different kinds of documents in different formats. All document instances related to the agenda of a meeting are stored on a network drive in a folder. The folder includes the XML documents (InfoPath forms) as well as assembled HTML and pdf output

documents, both for the agenda and the corresponding minutes. None of the documents created for the Faculty Council meetings have ever been destroyed, following the current archival policy at the university.

The InfoPath form templates, XSL transformation files, CSS style sheets, and custom XML schemas are archived in their own folders on a network drive. Only the latest versions of the class metadata files are stored. The office personnel creating agendas has no need to access these folders for agenda content authoring.

Impact analysis

The design of the XML document management was activated by problems related to laborious document preparation and publishing process, and frequently occurring errors in Faculty Council Meeting Agendas. The main concern was in solving these problems. Therefore improvements in the work of people at the Faculty Office and decrease in the number of errors in the agendas were regarded as the main objectives to the development project. In the new environment the work is supported by the adaptation of the authoring environment and by the support of automation. Members of the office staff were briefly trained before the deployment of the new system. The transfer did not cause major problems or complains as the users were closely involved in the development process. This ensured commitment to the use of the new solution. In the content production phase the need for manual typing and copy pasting has decreased and the automated creation of the HTML format for online publishing has replaced the earlier laborious manual gathering of content to the HTML format. No systematic evaluation of the number of errors in meeting agendas has been carried out, but most obviously the form-based data input with constraints controlled by the schema has decreased the number of errors.

In the case the focus in the development was in the work of document authors. The management of documents as records and their long-term archival was left outside the development project. The daily work at the Faculty Office does not require XML knowledge. For occasional changes needed to XML schemas, InfoPath form templates, transformation files, or style sheets, an XML expert has been requested. So far such expertise has been available at the faculty but some concerns have been stated about the lack of policies for ensuring the needed XML knowledge.

Evaluation

Our main objective was to develop a model for analyzing and describing XML document management throughout document life. Our development process was iterative. We first designed a preliminary model and then gathered data from two cases. After several discussions and test descriptions we ended up to the life cycle consisting of the five phases as shown in Figure 4. It provided, together with the XML document management environment model (Figure 3), a good basis to analyze and describe the cases in a uniform way. In the case environments the case descriptions were seen as useful tools to discuss and analyze the solutions and also as tools to record the core characteristics of the environments. One of the principles related to the artifact created in design science concerns the level of abstraction (Österle et al., 2011). According to this principle the artifact should be applicable to a class of problems. Analysing parallel two different cases during the model development was an important means to avoid developing a model suitable just for one case.

A great number of life cycle models have been earlier introduced and the term is understood in different communities in different ways. Life cycle models usually describe the life of an object as a process of sequential stages or phases or states. For example, the traditional life cycle concept of records management and archival has consisted first of records management activities and then of archival activities. The *records continuum model* has been introduced as an alternative to the life cycle model to avoid the strict temporal separation of the records management and archival activities, and to enable holistic, multi-dimensional analysis of the complexity of recordkeeping and archiving (see e.g. Atherton, 1985-1986; McKemmish, 2001; Shepherd, 2010; Upward, 2000). In the model records management is considered as a continuum of four kinds of activities: create, capture, organize, and pluralize. The four kinds of activities are called dimensions and rather than seeing them as temporally proceeding phases of a single process they are regarded as different perspectives to the management of records. Each of the dimensions is divided by four axes: recordkeeping, evidential, transactional, and identity.

We have earlier pointed out that our life cycle model is flexible, allowing iteration and parallel activities. In its own way it describes XML document management as continuum. While the continuum model intends to describe the complexity of records management in a single model, we provide different modeling techniques to analyze document/content management from different perspectives. Our life cycle model emphasizes the role of design as a continuous process, starting before the creation of documents, and continuing parallel with other kinds of activities.

Compared to the earlier SGML/XML document management development methodologies like those described earlier in this article, our life cycle model emphasizes

the role of retention in XML document management. The retention of business documents has become an important topic especially after the Sarbanes-Oxley Act of 2002 of the United States. The law mandates the retention of electronic documents and criminalizes the altering or destroying electronic records (Volonino, 2003; Stephens, 2005). Organizations are required to facilitate *e-discovery*, meaning the process of gathering electronic information for legal, regulatory, or administrative actions (Volonino et al., 2007). The retention of electronic records is important also in public sector. In Finland, for example, the law about electronic public sector services (24.1.2003/13) requires the archival of electronic documents so that their authenticity and integrity can be later demonstrated. Lack of understanding of the special features of XML document management environments may cause problems for retention and e-discovery. One of the special features of XML document management environments is their dynamics. It is typical to XML document management environments that the design is an ongoing process. Therefore we see it important to show the design phase explicitly in the model as the first phase.

Conclusion

In the paper we first introduced the concepts related to XML document management and described methodologies that have earlier been used to analyze XML document management in organizations. We argued that the methodologies lack support for the analysis of XML document management throughout their life, from the design to the retention. To fill the gap, we developed a model for the purpose, following the design science approach.

The main contribution of the study is twofold. First, we have shown that an XML document management environment is a complex combination of various content items, processes, actors with different backgrounds, and continuously evolving systems. This contribution should help researchers in achieving better understanding of the characteristics of XML environments and invent new research ideas. It should also help practitioners to create and maintain solutions that enable document retention and e-discovery. Second, we have created a new life cycle model as an extension of the earlier RASKE methods and showed how it can be used to describe an implemented case. This contribution provides for practitioners a tool for systematic analysis and description of XML document management and thereby also a tool to describe different cases in a uniform manner. A great number experts from different backgrounds and viewpoints is often involved in XML environments and in their development. Well-defined concepts and models, and the capability to compare different cases, should help the experts to collect and communicate their knowledge and see possible development needs in their own environments.

During the development of the model we tested its applicability only in cases where the deployment of XML had already been implemented. Further research is needed to show how the model could support the design of new solutions for XML document management. Basically there are two different kinds of support. On the one hand, earlier case descriptions can be used as a means to learn about XML document management and earlier implementation cases. On the other hand, the concepts and models of the framework may be used as a tool to describe the new XML document management solutions.

Both in public and private sector organizations there is a strong tendency to shift from paper archival to digital archival. At the same time, new legislation is created to mandate the preservation of digital information produced in business processes. Especially in case of legal auctions organizations are required to discover all relevant documents and demonstrate their reliability and authenticity. E-discovery is a complex and often costly procedure. An interesting area of future research is to consider XML document management especially from the point of view of e-discovery.

As mentioned earlier, the RASKE methodology has provided models and methods for holistic analysis of document management environments. It has some special support for analyzing structured documents, but its use is not restricted to XML-encoded documents. The same concerns also the life cycle model introduced in this paper. The five phases of Figure 4 are not restricted to XML documents. Therefore the model is applicable in analyzing electronic document management also more generally.

References

Ali, M.S., Consens, M., and Lalmas, M. (2012). Extended structural relevance framework: a framework for evaluating structured document retrieval. Information Retrieval 15(6), 558-590.

Arvola, P., Kekäläinen, J., and Junkkari, M. (2011). Contextualization models for XML retrieval. Information Processing and Management 47(5), 762-776.

Atherton, J. (1985-1986). From life cycle to continuum: Some thougths on the records management-archives relationship. Archivaria 21, 43-51.

Aversano, L., Canfora, G., de Lucia, A., & Gallucci, P. (2002). Integrating document and workflow management tools using XML and Web technologies: A case study. In T. Gyimóthy, & F. Brito e Abreu (Eds.), Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02). Washington, DC: IEEE.

Barker, R.M., Cobb, A.T., & Karcher, J. (2009). The legal implications of electronic document retention: Changing the rules. Business Horizons 52(2), 177-186.

Bernard, R. (2007). Information lifecycle security risk assessment: A tool for closing security gaps. Computers & Security 26(1), 26-30.

Bernstein, P.A., & Haas, L.A. (2008). Information integration in the enterprise. Communications of the ACM 51(9), 72-79.

Bertino, E., & Ferrari, E. (2002). Secure and selective dissemination of XML documents. ACM Transactions on Information and System Security 5(3), 290-331.

Bertino, E., Ferrari, E., Paci, F., & Provenza, L.P. (2007). A system for securing push-based distribution of XML documents. International Journal of Information Security 6(4), 255-284.

Bhatti, R., Ghafoor, A., Bertno, E., & Joshi, J.B.D. (2005). X-GTRBAC: an XML-based policy specification framework and architecture for enterprise-wide access control. ACM Transactions on Information and System Security 8(2), 187-227.

Blanke, T., Lalmas, M., and Huibers, T. (2012). A framework for the theoretican evaluation of XML retrieval. Journal of the American Society for Information Science and Technology 63(12), 2463-2473.

Borglund, E.A.M. (2008). Design for recordkeeping: areas of improvement. PhD Thesis, Department of Natural Sciences, Mid Sweden University Doctoral Thesis 52.

Retrieved April 18, 2013, from http://miun.diva-portal.org/smash/get/diva2:1952/FULLTEXT01

Bray, T., Paoli, J., & Sperberg-McQueen, C. M., (Eds.) (1998). Extensible Markup Language (XML) 1.0, W3C Recommendation, 10 February 1998, W3C Consortium. Retrieved April 18, 2013, from http://www.w3.org/TR/1998/REC-xml-19980210.

Broberg, M. (2004). A successful documentation management system using XML. Technical Communication 51 (4), 537-546.

Brooke, P.J. Paige, R.F., & Power, C. (2010). Document-centric XML workflows with fragment digital signatures. Software – Practice and Experience 40(8), 655-672.

Brooks, T.A. (2001). Where is meaning when form is gone? Knowledge representation on the Web. Information Research 6 (2).

Cerri, D., & Fuggetta, A. (2007). Open standards, open formats, and open source. Journal of Systems and Software 80(11), 1930-1937.

Chen, M. (2003). Factors affecting the adoption and diffusion of XML and Web services standards for E-business systems. Human-Computer Studies, 58(3), 259-279.

Clark, J., & Murata, M. (Eds.) (2001). RELAX NG Specification, Committee Specification 3 December 2001, OASIS. Retrieved April 18, 2013, from http://www.oasis-open.org/committees/relax-ng/spec-20011203.html.

Costa, G., Manco, G., Ortale, R., and Ritacco, E. (2013). Hierarchical clustering of XML documents focused on structural components. Data & Knowledge Engineering 84, 26-46.

DLM Forum Foundation (2011). MoReq2010®: Modular Requirements for Records Systems – Volume 1: Core Services & Plug-in Modules. Retrieved April 18, 2013, from http://moreq2010.eu/

Dublin Core Metadata Initiative (2010). Dublin Core Metadata Element Set, Version 1.1. 2010-10-11. Retrieved April 18, 2013, April 18, 2013, http://dublincore.org/documents/dces/

Genevès, P., Layaïda, N., & Quint, V. (2011). Impact of XML schema evolution. ACM Transactions on Internet Technology 11 (1), 4:1-4:27.

Gilliland, A.J. (2008). Setting the Stage. In T. Gill, A.J. Gilliland, Whalen, M., & Woodley, M.S., Introduction to Metadata, Online Edition, Version 3.0. Los Angeles, CA: Getty Publications. Retrieved April 18, 2013, from http://getty.edu/research/publications/electronic_publications/intrometadata/setting.html

Glushko, R.J., & McGrath, T. (2005). Document Engineering: Analysing and Designing Documents for Business Informatics and Web Services. Cambridge, MA: MIT Press.

Goldfarb, C. F. (1990). The SGML Handbook. Oxford: Oxford University Press.

Greenberg, J. (2010). Metadata and digital information. In M.J. Bates & M. Niles Maack (eds.), Encyclopedia of Library and Information Science, Third Edition, 1:1 (pp. 3610-3623). New York: Taylor & Francis.

Gregor, S., & Jones, D. (2007). The anatomy of a design theory. Journal of the Association for Information Systems 8(5), 312-335.

Heslop, H., Davis, S., & Wilson, A. (2002). An approach to the preservation of digital records. national Archives of Australia. Retrieved April 18, 2013, from http://www.aa.gov.au/Images/An-approach-Green-Paper_tcm16-47161.pdf

Hevner, A.R., March, S.T., & Park, J. (2004). Design research in information systems research. MIS Quarterly 28(1), 75-105.

Horik, van, R., & Roorda, D. (2011). Migration to intermediate XML for electronic data (MIXED): Repository for durable file format conversions. The International Journal of Digital Curation 2(6), 245-252.

ISO 15489-1 (2001). Information and documentation – Records management. Part 1: General.

ISO/TR 15489-2 (2001). Information and documentation – Records management. Part 2: Guidelines.

ISO/IEC 26300 (2006). Information technology – Open Document Format for Office Applications (OpenDocument) v1.0.

ISO/IEC 29500-1 (2008). Information technology – Document description and processing languages – Office Open XML File Formats – Part 1: Fundamentals and Markup Language Reference.

Järvenpää, M., Virtanen, M., & Salminen, A. (2006). Semantic portal for legislative information. In M. Wimmer, H.J. Scholl, Å. Grönlund, K. Viborg Andersen (Eds.), Proceedings of the Fifth International Conference on Electronic Government (EGOV 2006). Lecture Notes in Computer Science 4084 (pp. 219-230). Berlin: Springer Verlag.

Kundu, A., & Bertino, E. (2008). A new model for secure dissemination of XML content. IEEE Transactions on Systems, Man, and Cybernetics 3 (3), 292-301.

Kwietniewski, M., Gryz, J., Hazlewood, S., Van Run, P. (2010). Transforming XML documents as schemas evolve. Proceedings of the VLDB Endowment 3(1-2), 1577-1580.

Liu, H., Wang, X., & Quan, Q. (2009). Research on the enterprise' model of information lifecycle management based on enterprise architecture. Proceedings of the Ninth International Conference on Hybrid Intelligent Systems (pp. 165-169). New York, NY: IEEE Computer Society.

Liu, S., McMahon, C.A., & Culley, S.J. (2008). A review of structured document retrieval (SDR) technology to improve information access performance in engineering document management. Computers in Industry 59(1), 3-16.

Luk, R.W.P., Leong, H.V., Dillon, T.S., Chan, A.T.S., Croft, W.B., & Allan, J. (2002). A survey in indexing and searching XML documents. Journal of the American Society for Information Science and Technology 53(6), 415-437.

March, S.T. & Smith, G.F. (1995). Design and natural science research on information technology. Decision Support Systems 15 (4), 251-266.

Maler, E., & El Andaloussi, J. (1996). Developing SGML DTDs. From text to model to markup. Englewood Cliffs, NJ: Prentice Hall.

McKemmish, S. (2001). Placing records continuum theory and practice. Archival Science 1 (4), 333-359.

Mella, G., Ferrari, E., Bertino, E., & Koglin, Y. (2006) Controlled and cooperative updates of XML documents in byzantine and failure-prone distributed systems. ACM Transactions on Information and System Security 9(4), 421-460.

Ministry of Government Administration, Reform and Church Affairs (2007). Open document standards to be obligatory for state information. Press release. Retrieved April 18, 2013, from http://www.regjeringen.no/en/dep/fad/pressesenter/pressemeldinger/2007/Open-document-standards-to-be-obligatory.html?id=494810

Noy, N.F., & Hafner, C.D. (1997). The state of the art in ontology design. A survey and comparative review. AI Magazine 18(3), 53-74.

Nunamaker, J.F., Chen, M., & Purdin, T.D.M. (1990). Systems development in in-formation systems. Journal of Management Information Systems 7(3), 89-106.

Nurmeksela, R., Jauhiainen, E., Salminen, A., & Honkaranta, A. (2007). XML document implementation: Experiences from three cases. In Y. Badr, R. Chbeir, & P. Pichappan (Eds.), Proceedings of the Second International Conference on Digial Information Management (pp. 224-229). Los Alamitos, CA: IEEE.

OCLC and CRL (2007). Trustworthy Repositories Audit & Certification: Criteria and Checklist. Version 1.0. Retrieved April 18, 2013, from http://www.crl.edu/sites/default/files/attachments/pages/trac_0.pdf

Peffers, K., Tuunanen, T., Rothenberger, M.A., & Chatterjee, S. (2008). A design science research methodology for information systems research. Journal of Management Information Systems 24(3), 45-77.

Peled, A. (2011). When transparency and collaboration collide: The USA Open data Program. Journal of the American Society for Information Science and Technology 62(11), 2085-2094.

Pérez, J.M., Berlanga, R., Aramburu, M.J., & Pedersen, T.B. (2008). Integrating data warehouses with Web data: A survey. IEEE Transactions on Knowledge and Data Engineering 20(7), 940-955.

Powell, A., & Johnston, P. (2003). Guidelines for implementing Dublin Core in XML. Dublin Core Metadata Initiative. Retrieved April 18, 2013, from http://dublincore.org/documents/dc-xml-guidelines/

Reuben, E. (2003). Migrating records from proprietary software to RTF, HTML, and XML. Computers in Libraries 23(6), 30-33.

Rockley, A., Kostur, P., & Manning, S. (2003). Managing Enterprise Content: A Unified Content Strategy. Indianapolis, IN: New Riders.

Runardotter, M., Mörtberg, C., Mirijamdotter, A. (2011). The changing nature of archives: whose responsibility? Electronic Journal of e-Government 9(1), 68-78.

Rönnau, S., & Borghoff, U.W. (2009). Versioning XML-based office documents. An efficient, format-independent, merge-capable approach. Multimedia Tools and Applications 43(3), 253-274.

Salminen, A. (2005). Building digital government by XML. In R.H. Sprague, Jr. (Ed.), Proceedings of the Thirty-Eighth Hawaii International Conference on System Sciences (HICSS-38). Los Alamitos, CA: IEEE Computer Society.

Salminen, A. (2010). Modelling documents in their context. In M.J. Bates & M. Niles Maack (Eds.), Encyclopedia of Library and Information Sciences, Third Edition. New York: Taylor & Francis. DOI: 10.1081/E-ELIS3-120044399.

Salminen, A., Kauppinen, K., & Lehtovaara, M. (1997). Towards a methodology for document analysis. Journal of the American Society for Information Science 48(7), 644-655.

Salminen, A., Lyytikäinen, V., & Tiitinen, P. (2000). Putting documents into their work context in document analysis. Information Processing & Management 36(4), 623-641.

Salminen, A., Lyytikäinen, V., Tiitinen, P., & Mustajärvi, O. (2004). Implementing digital government in the Finnish Parliament. In W. Huang, K. Siau, & K.K. Wei (Eds.), Electronic Government Strategies and Implementation (pp. 242-259). Hersley, PA: IDEA Group Publishing.

Salminen, A., Nurmeksela, R., Lehtinen, A., Lyytikäinen, V., & Mustajärvi, O. (2008). Content production strategies for e-Government. In A.-V. Anttiroiko (Ed.), Electronic Government: Concepts, Methodologies, Tools, and Applications. Hersley, PA: Information Science Reference.

Salminen, A., & Tompa, F.W. (2001). Requirements for XML document database systems. In E.V. Munson (Ed.), Proceedings of the ACM Symposium on Document Engineering (DocEng '01) (pp. 85-94). New York: ACM Press.

Salminen, A. & Tompa, F. (2011). Communicating with XML New York: Springer-Verlag New York Inc.

Sartor, G., Palmirani, M., Francesconi, E., & Biasiotti, M. A. (Eds.) (2011). Legislative XML for the Semantic Web. Law, Governance and Technology Series 4. Dordrecht: Springer.

Shah, R., Kesa, J., & Kennis, A. (2008). Implementing open standards: A case study of the Massachusetts open formats policy. In Proceedings of the 2008 International Conference on Digital Government Research (pp. 262-271). Los Angeles, CA: Digital Government Society of North America.

Shepherd, E. (2010). Archival Science. In M.J. Bates & M. Niles Maack (Eds.), Encyclopedia of Library and Information Sciences, Third Edition (pp. 179-191). New York: Taylor & Francis.

Smith, H.A., & McKeen, J.D. (2003). Developments in practice viii: enterprise content management. Communications of AIS 11(33), 1-26.

Sohn, W.-S., Ko, S.-K., Lee, K.-H., Kim, S.-H., Lim, S.-B., & Choy, Y.-C. (2002). Standardization of eBook documents in the Korean industry. Computer Standards & Interfaces 24(1), 45-60.

Sprague Jr., R.H. (1995). Electronic document management: Challenges and opportunities for information systems managers. MIS Quarterly 19(1), 29-49.

Stanescu, A. (2005). Assessing the durability of formats in a digital preservation environment: The INFORM methodology. OCLC Systems & Services, 21 (1), 61 – 81.

State Records Authority of New South Wales (2007). Strategies for documenting government business: The Dirks Manual. Retrieved April 18, 2013, from http://www.records.nsw.gov.au/recordkeeping/dirks-manual

Stephens, D.O. (2005). The Sarbanes-Oxley Act: Records management implications. Records Management Journal 15(2), 98-103.

Tiitinen, P., Lyytikäinen, V., Päivärinta, T., & Salminen, A. (2000). User needs for electronic document management in public administration: a study of two cases. In H.R. Hansen, M. Bichler, & H. Mahrer (Eds.), Proceedings of ECIS 2000, European Conference on Information Systems, Volume 2 (pp. 1144-1151). Wien: Wirtschaftsuniversität Wien.

Upward, F. (2000). Modelling the continuum as paradigm shift in recordkeeping and archiving processes, and beyond – a personal reflection. Records Management Journal 10 (3), 115-139.

Volonino, L. (2003). Electronic evidence and computer forensics. Communications of the Association for Information Systems 12 (Article 27), 457-468.

Volonino, L., Sipior, J.C., & Ward, B.T. (2007). Managing the lifecycle of electronically stored information. Information Systems Management 24(3), 231-238.

Wang, F., & Zaniolo, C. (2008). Temporal queries and version management in XML-based document archives. Data & Knowledge Engineering 65(2), 304-324.

Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., …, & Sinz, E.J. (2011). Memorandumdesign-oriented on information systems research. European Journal of Information Systems 20 (1), 7-10.