Information Resources Management Journal

January-March 2006 Vol. 19, No. 1

Table of Contents

A Metadata-Based Approach for Unstructured Document Management in Organizations

Federica Paganelli, University of Florence, Italy

Maria Chiara Pettenati, University of Florence, Italy

Dino Giuli, University of Florence, Italy

ABSTRACT

Effectively managing documents is a strategic requirement for every organization. Available document management systems (DMSs) often lack effective functions for automatic document management. One reason is that relevant information frequently is conveyed by unstructured documents, whose content cannot be easily accessed and processed by applications. This article proposes a metadata model, the DMSML (Document Management and Sharing Markup Language) to enable and to ease unstructured document management by supporting the design of DMSs. We argue that the extensive use of this metadata language will render organizational information explicit, promoting information reuse and interoperability in a more profitable way than what is guaranteed by proprietary DMSs. We also briefly depict the design and deployment phases of a Web-based DMS prototype based on DMSML. Our overall intent is to increase the awareness of what managers should account for when considering the possibility of adopting a DMS.

Keywords: design methodologies; document management systems; information system design; metadata; software development tools; Web-based applications

INTRODUCTION

Document Management (DM) is the scientific domain dealing with the use of ICTs for the effective "storage, organization, transmission, retrieval, manipulation, update, and eventual disposition of documents to fulfill an organizational purpose" (Sprague, 1995, p. 32). Existing ICT-based DM solutions, hereafter called document management systems (DMSs), do not completely fulfill the expectations of providing enough effective tools for information creation, sharing, and retrieval inside an organization, often causing user frustration, dissatisfaction, and inefficiencies (Ginsburg, 2001).

A typical situation that creates problems in many organizations is the management of unstructured documents that often convey important information and knowledge (the451, 2002); due to their lack of structure, these documents cannot be easily and effectively accessed and processed by applications, thus limiting effective document management. As a consequence, members of organizations have difficulty retrieving the information contained in these documents. Moreover, existing DMSs seldom are designed according to a general and/or standard methodological approach and are built around open data and process models. Thus, related disadvantages are vendor dependence, difficult maintenance, and poor interoperability with other information systems (Stickler, 2001).

In order to deal with these issues, we propose in this article a metadata model, enabling the design of DMSs and aiming at combining the benefits of metadata for document description with the use of Web standards. The metadata language described in this work has been named DMSML (Document Management and Sharing Markup Language). DMSML offers a solution to representing a set of document properties that are relevant to document management and to rendering business and organizational information explicit in a way that promotes reuse, user-driven extensibility, and interoperability with heterogeneous systems. A Web-based prototype developed according to DMSML specifications will help make the theoretical arguments presented throughout this article concrete.

This article is organized as follows: initially, some considerations on the management of unstructured documents are made in order to point out which relevant characteristics of an unstructured document are worth being described in order to improve its efficient management (its *content* and *context of use*). Then, an example of a typical document frequently managed in many types of organizations, the *project proposal*, will sustain what is said from a general perspective. The following paragraphs will be devoted to the analytical description of the requirements for high-quality DMSs. The fulfillment of these requirements will be taken as the basis for the design of the metadata language (DMSML) as well as for the development of the Web-based DMS prototype presented in this work. Examples and comparative evaluation of available products — both commercial and open source — will show that meeting all of the aforementioned requirements is a characteristic satisfied by none of the presented products, to the best of our knowledge.

The central part of this article is aimed at describing (from a general point of view) the use of metadata for document management, illustrating the benefits of their usage in this domain. Existing metadata languages related to document management at the state of the art also will be recalled in this part of the article.

In the sequel, the DMSML metadata specification will be described, highlighting its fulfillment of high-level requirements. DMSML will be proposed as a declarative language for the specification of DMSs, based on existing standards and on a rigorous modeling approach. Metadata modeling, using proper formal representation techniques, then will be illustrated. The advantage of using DMSML for DMS design and development will be described in the related paragraph.

Finally, the Web-based prototype using DMSML metadata specifications will be described in its functional components and implementative details. Using these arguments, we demonstrate that the extensive use of this metadata language in document management systems will help to exploit business and organizational information in a more profitable way than what is guaranteed by proprietary document management applications, because the knowledge (properly codified through a metadata language) will allow both human and machine readability and, consequently, more effective reusability.

MANAGEMENT OF UNSTRUCTURED DOCUMENTS

Unstructured documents are text and multimedia documents such as e-mails, reports, and so forth, stored in various formats that do not provide an explicit, formal, and separate representation of either content structure (also called logical structure) or presentation structure (physical structure). Often, these are encoded in proprietary and binary formats. For instance, a project proposal has a logical structure, because it is organized in a specific set of fields: project title, objectives, activity plan, business plan, and so forth. It also has a physical structure related to presentation instructions; for instance, a project title should be rendered in bold, centered, font size 14.

In unstructured formats, this information usually is blended and cannot be easily and/or automatically extracted and processed by applications. As a consequence, applications do not have explicit references to specific elements of the content, thus preventing automatic document processing and leading to poor interoperability among applications. Currently, effective indexing, retrieval, and processing would require the system to access document content with a degree of granularity, which cannot be provided by unstructured document formats.

As a matter of fact, collaboration among colleagues can be slowed down dramatically by factors such as the use of different word processing applications and graphical and CAD tools. This dramatically compromises the quality of information reuse and sharing in organizations. It is important to highlight that organizational information not only refers to document content but also to its context of use. With the term *context of use* of a document, we hence refer to by whom, where, how, under which constraints, and for which purpose a document is being accessed, transmitted, and modified (Gilliland-Swetland, 2000; Päivärinta, 2001). For instance, a project proposal document contains a set of information that is strategic in the accomplishment of organizational goals (e.g., objectives and planned activities for innovation in terms of products, processes, and/or services), but it also refers to a set of organizational information that describes its context of use. This kind of information is related strictly to organizational processes, roles, and responsibilities. Usually, business information, such as access right policies and document lifecycle descriptions, is encoded in the application logic of a DMS. In most commercial products, this information is specified in proprietary formats and cannot be reused by other systems. On the other hand, many open source solutions are based on open standards, but the organizational information is encoded in a technology-dependent way. As a consequence, the reuse or extension of this business information and the migration to new technologies can be a difficult process requiring high technical expertise, time, and resources.

EXAMPLE OF A TYPICAL DOCUMENT TO BE MANAGED: A PROJECT PROPOSAL

To clarify the benefits of metadata usage for document management, we describe through this example the characteristics of a common type of document, which is relevant in private as well as academic domains — the project proposal.

In a generic organization, a typical process of project proposal editing can envisage different steps toward the completion of the document. A heterogeneous group of persons can participate in this process, including administrative staff, technical experts, marketing staff, external consultants, and so forth. Different responsibilities then will be assigned to each person according to his or her skills and organizational role.

Possible steps in project proposal editing are:

- Editing an abstract, describing the main objectives of the future project, and
- Editing of the project proposal, usually according to a predefined document template; in this phase, actors can collaboratively edit the document.

While undergoing these phases, the project proposal passes through different states, such as draft document, document under review, document under revision, final proposal, and submitted document. The transition between states is regulated by actions operated by persons with specified roles; for instance, the transition from *document under review* to *final* is conditioned by the approval of the person in charge of the project (e.g., a project manager).

According to what is illustrated in this example, it becomes evident that a lot of strategic information needs to be codified and shared among the actors in order to execute the overall process. The organizational information related to the documents' context of use usually refers to the definition of roles in the organizational schema, access control policies, and lifecycle of documents. In order to speed up the process of project proposal editing, this information should be formalized conveniently and made available to the involved actors. Moreover, it should be available after the process is completed in order to formalize the practice and to make this experience available to other colleagues for future activities, such as editing another project proposal.

REQUIREMENTS FOR DOCUMENT MANAGEMENT SYSTEMS

The main critical issues concerning unstructured document management thus can be summarized in terms of poor reuse of content and context of use of business information. These considerations provide hints for the definition of the following general requirements that DMS should fulfill in order to enable effective document management:

- **Standard compliance.** The adoption of international widely accepted standards promotes interoperability among heterogeneous information systems and data sources. Standard compliance includes technological as well as business standards. Examples of technological standards are the eXtensible Markup Language (XML) (Sall, 2002) and J2EE (Java 2 Enterprise Edition) (Sun Microsystems, 2003). With the term *business standard*, we refer to the specifications related to business information describing documents' context of use. Examples of business standards are business process definition languages, such as the XML Process Definition Language (XPDL) (Workflow Management Coalition, 2002), the Petri Net Markup Language (PNML) (Weber & Kindler, 2002), and access right policy languages, such as the eXtensible Access Control Markup Language (XACML) (OASIS, 2003).
- **Multi-platform compatibility and support** (Stickler, 2001). A DMS solution should be deployable on different platforms in order to avoid monolithic and vendor-dependent solutions. Moreover, support of several platforms enables interoperability among heterogeneous information systems and facilitates the integration with existing legacy systems in the organization (e.g., http server, mail server, etc.).
- **Metadata-based approach for the representation of document properties** (Karjalainen et al., 2000; Murphy, 1998; Päivärinta, 2001; Salminen et al., 2000). Metadata are data about data. They provide an explicit representation in a human- and machine-understandable format of document properties. Traditionally, metadata are used in order to represent descriptive properties (i.e., title, author, keywords, etc.) of information resources in order to support document classification, search, and retrieval (Gilliland-Swetland, 2000). An example of descriptive metadata is the Dublin Core (DC) specification (Dublin Core Metadata Initiative, 2003). Moreover, metadata could be conveniently used to describe the document's context of use. In this way, business information, such as document lifecycle and access policies, can be specified by abstracting it from implementation details. For instance, the previously mentioned business standards (e.g., XPDL, PNML, and XACML) provide a meta-language, making it possible to describe organizational processes by means of standardized labels (e.g., activity, task, actor).
- **A standard methodological approach.** It should be followed for the design and de-

velopment of a DMS rather than a tool-oriented approach (Stickler, 2001). In fact, a methodological approach based on standard models and methods for DMS design and development can facilitate the formalization of user requirements, fast prototyping, and deployment of a high-quality product, accomplishing the formalized requirements. Moreover, a standard methodological approach for the design of an effective DMS should conveniently promote the accomplishment of the previous requirements (i.e., standard compliance, multi-platform compatibility and support, metadata-based approach) by including proper design and development methods and techniques.

The fulfillment of these requirements can lead to an effective, easily maintainable, flexible, and cost-effective solution for document management.

Existing Document Management Systems

At present, several commercial and open-source DMSs are available. Among the commercial products, the most important solutions in terms of market diffusion in the domain of document management are Documentum, FatWire, IBM, Interwoven, and Microsoft[1] (Moore & Markham, 2002). Among the open-source products, Zope Content Management Framework, OpenCMS, Apache Lenya, and open-source solutions for digital libraries, such as DSpace, and Marian[2], deserve to be mentioned. A comparison of some of these products with respect to their provided features is provided by Moore and Markham (2002). For the purpose of this article, we will evaluate some of these products according to their compliance with the previously mentioned requirements for DMS: standard compliance, multi-platform compatibility and support, metadata-based approach, and use of a standard methodological approach. The analysis synthesized in Table 1 refers to two commercial products— FatWire Content Server and Documentum — and three open-source products — Apache Lenya, Dspace, and Marian.

The analysis of these products highlights that the compliance with technical standards and multiplatform compatibility are requirements commonly understood and addressed by means of wide adoption of industrial standards, such as XML and related standards (Sall, 2002), LDAP (Lightweight Directory Access Protocol) (Yeong et al., 1993), SOAP (Simple Object Access Protocol) (Mitra, 2003), J2EE, and Internet protocols such as HTTP (Hypertext Transfer Protocol) and FTP (File Transfer Protocol). On the other hand, compliance with business standards is partially accomplished. In fact, while descriptive metadata standards often are used in open source solutions (e.g., Dublin Core), metadata standards for lifecycle and access policy descriptions rarely are used. Moreover, only MARIAN is associated with an open and standard methodological approach for the design and development of the product tailored to the requirements of a specific organization (Gonçalves et al., 2004). The discussion on commercial products is limited by the lack of documentation about some requirements (business standard compliance and metadata-based approach). The overall remark of this analysis is that these products do not address completely the previously mentioned high-level requirements for DMSs.

Research Directions in Document Management

The research in document management as a discipline that encompasses social and organizational issues and user needs and potentially utilizes several technologies in an organizational context is still in its infancy. The work of Sprague (1995) paves the way for a systemic view of electronic document management, which should integrate three perspectives of analysis of document management: *technologies* for document management, *benefits* for the application areas for which documents are mission-critical, and *roles and responsibilities* of the organization's departments and func-

*Table 1. DMS's evaluation results (***= good; ** = sufficient; * = insufficient compliance level with requirements for document management systems; - = no publicly available information)*

Requirements / Products	Open standards compliance		Multi-platform support	Metadata-based approach		Open and standard methodological approach	
	Technical standards	Business standards		Content	Context of use	Data model	Method
FatWire	*** LDAP, XML, SOAP and Internet protocols	-	*** J2EE compliant	-	-	* Object model	-
Documentum	*** LDAP, XML, SOAP and Internet protocols	-	*** J2EE compliant	-	-	* Object - relational model	-
Lenya	*** LDAP, XML and Internet protocols	* No standards for lifecycle and access policy	*** J2EE compliant	*** Dublin Core compliant	* Dublin Core compliant	** Open data model	-
DSpace	** Internet protocols	* No standards for lifecycle and access policy	** It runs only on UNIX and Linux OSs	*** Dublin Core compliant	* Partial compliance with Dublin Core	** Open relational data model	-
MARIAN	*** Internet protocols and XML	* No standards for lifecycle and access policy	*** Thanks to Java code portability	*** It is compliant to USMARC and DC	* It is compliant to USMARC and Dublin Core	** Open data model	*** The study of a standard method is in progress

tions for which EDM will be strategic. Based on this seminal contribution, Päivärinta (2001) proposes a method for the requirements analysis for DMS design, based on the use of metadata and the genre theory of organizational communication (Yates & Orlikowski, 1992).

The use of metadata, together with markup languages and formal information models, is recognized widely as a basic mechanism for DMS design (Murphy, 1998; Murray, 1996; Salminen et al., 2000). A more implementation-oriented approach is discussed in Ginsburg (2000, 2001), providing practical guidelines for the design of DMS, such as the use of metadata, mechanisms of coordination between authors and readers, and ontology building. However, this contribution does not aim to define a methodological approach for design, development, and deployment of DMSs supported by standard modeling methods and techniques.

METADATA FOR DOCUMENT MANAGEMENT

Metadata Benefits in Document Management

As mentioned in previous sections, metadata allow the representation of information resource relevant properties in a human- and machine-understandable way. Especially in the case of unstructured document management, metadata are the mechanisms that enable the representation of document content and context of use properties in an explicit and formal way. Metadata can provide a solution to the opacity of unstructured documents (Ginsburg, 2001), allowing a machine-processable representation of document-relevant information.

At present, metadata are used in organizations to describe unstructured documents beyond those used for highly structured information (e.g., databases); for instance, the properties for word-processed documents, the metadata contained in the header records of e-mail messages, and directories of reusable software objects, and the indexes of digital image management and manual record retention systems (Murphy, 1998). Murphy (1998) states that metadata in organizations generally lack the centralized or controlled aspect that metadata have in application domains such as digital libraries or Web communities, and, consequently, they are not exploitable enough in information sharing, organizational learning, or knowledge management. As far as we know, this situation has not changed for many years. Moreover, while several standardization efforts exist for digital libraries, Web communities and other application domains, standard metadata models in document management are still lacking.

State of the Art

Several metadata sets have been proposed by research communities and/or industrial consortia in order to provide a standard way to describe and manage information resources. Most metadata standards belong to the following categories:

- **Description of information resources.** These standards provide metadata for the description and identification of information resources. Examples are Dublin Core (Dublin Core Metadata Initiative, 2003), which is a standard for library information items and also suitable for application to generic information objects; and MPEG-7 (Manjunathn et al., 2002) for multimedia content.
- **Specific functions of information management.** Some of the following metadata standards cover specific issues related to the management of information resources: XPDL and PNML standards for business process description, Extensible Access Control Markup Language (XACML) for the description of access right policies, Common Warehouse MetaModel, CWM (Object Management Group, 2001) for data warehouses, ISO 15489 (International Organization for Standardization, 2001) for record management, just to mention a few.
- **Specific application domains.** There are several standard propositions for specific application domains: USMARC for digital libraries (Crawford, 1989), IMS (IMS Global Learning Consortium, 2003) for e-learning, and IEC 82045 (International Electrotechnical Commission, 2001) for management of technical documents.

From this classification, no existing metadata specifications are focused purposely on document management. The single exception is represented by the IEC 82045 specification, which, however, is a restraint to technical documents.

Some existing standard specifications could be conveniently adapted and integrated in order to represent document properties in organizations (Päivärinta et al., 2002). For instance, Dublin Core (DC), one of the most widely adopted metadata standards for information item description, offers generic descriptive labels that could be used to describe content-related properties of documents in organizations. The DC element set provides 15 labels: *Coverage*, *Creator*, *Date*, *Description*, *Format*, *Language*, *Other Contributors*, *Publisher*, *Relation*, *Resource Identifier*, *Resource Type*, *Rights Management*, *Source*, *Subject*, *Title*. Most of them are meaningful in an organizational context. An exception is certainly provided by the label *Publisher*, which is tied to the author-title-publisher model that traditionally is applicable to documentation that is made publicly available (e.g., books, journals, etc.). This model is not always applicable to organizational documents, as the volume of internal use documents largely exceeds documents from external sources (Murphy, 1998). Obviously, a special case is provided by those kinds of organizations (e.g., publishing companies) that produce material that should be made available for public distribution. In general, a document is described more usefully through its states, such as *draft*, *authorized*, *signed*, declaring the evolution of the document during its lifecycle but not necessarily by condition of public distribution of the resource. The example of the document *project proposal*, illustrated in a previous paragraph, clearly highlights what was stated before. As already mentioned, a metadata model that is useful for document management should include the description of the organizational context (i.e., the lifecycle model of the documents) and its relation with organizational processes, roles, and responsibilities (Salminen et al., 2000).

Many standardization efforts exist in the domain of record management, such as the ISO 15849 international standard (International Organization for Standardization, 2001). A Record "is evidence of an activity or decision and demonstrates accountability" (Public Record Office, 2001, p.7). Records are documents (structured and unstructured), whose management requires a rigorous process (Emery, 2003). While a Document Management System is focused on knowledge sharing and collaboration capabilities that can be promoted by using a document repository, a Record Management System is focused more on maintaining a repository of evidence that can be used to document events related to statutory, regulatory, fiscal, or historical activities within an organization (Emery, 2003).

DOCUMENT MANAGEMENT AND SHARING MARKUP LANGUAGE

This article aims to propose a metadata model named Document Management and Sharing Markup Language (DMSML) that represents document properties that are relevant to document management and enable the design of Web-based DMSs in a way that promotes the reuse of content and context of use information, which is conveyed by unstructured documents for organizational purposes. In the following section, we describe the high-level requirements for DMSML metadata specification and the modeling approach that we have adopted.

DMSML Metadata Specification

Specification of metadata elements should guarantee that metadata are representative and relevant properties of documents. In order to decide which properties are representative and relevant for our needs, some high-level requirements are defined. Although these guidelines drove the specification process of DMSML, they are generally applicable to metadata specification in other application domains.

High-Level Requirements for DMSML Specification

The high-level requirements that we identified for the specification of DMSML are: generality, extensibility, and interoperability.

Generality. The metadata model should be applicable to document management for organizations in any application domain (e.g., public administration, construction, software, services businesses, etc.). General commonly sharable and widely adopted labels should be selected. An example of fulfillment of generality is the choice to label as *creator* the person who created a document, instead of *journalist* or *writer*, which carries the significance of a specific application domain.

Extensibility. Since DMSML cannot contain specific labels tailored to any application domain, it will allow the extension of its element set. Refinement of existing elements (e.g., substitution of the element *creator* with the element *journalist*) or the introduction of new elements should be allowed in order to deal with the requirements of specific application domains. Moreover, the extension of the metadata set should be driven by business requirements rather than technological choices.

Interoperability. Interoperability is defined as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged" (Institute of Electrical and Electronics Engineers, 1990). Two applications that are both DMSML-aware can exchange just the metadata or the metadata with the related document and be able to access and meaningfully process them.

DMSML is not a standard proposition, and interoperability based on DMSML-awareness may be limited, as long as the metadata set is not widely adopted. Furthermore, as mentioned previously, some specific issues that are encompassed in the discipline of document management (e.g., information resource description, access management, and business process) are addressed by existing standards. The adoption of some of these standards in the DMSML specification is an effective way to promote interoperability and to create a comprehensive framework of document management metadata, which takes advantages of existing contributions.

Metadata Modeling

The use of proper formal representation techniques enables the unambiguous understanding of the concepts conveyed by the metadata elements in heterogeneous communities and promotes the metadata exchange across heterogeneous systems (Duval et al, 2002; Murray, 1996). Most metadata standards lack an accounting for underlying data modeling principles, thus not providing a clear exposition of entities and relationships represented in metadata specification (Lagoze & Hunter, 2001). As suggested by several sources (Duval et al., 2002; Melnik & Decker, 2000), it is important to express in a formal and abstract way concepts and relationships embedded in the specification (i.e., the meaning or semantics) and to share strategies and rules for metadata encoding and implementation (syntax) for computer-supported serialization, exchange, access, and manipulation of the metadata set.

Meaning and syntax should be kept as separate as possible in order to allow agreement and adoption of the metadata set independently from technological and implementation choices, which can vary over time.

In order to provide a rigorous formalization of the DMSML metadata model, we refer to two layers of data modeling, usually adopted in the traditional database design approach (Elmasri & Navathe, 2003).

1. **Conceptual layer.** This layer provides an abstract representation of concepts and relations among concepts, independent from implementation details and often by means of standard graphical notations. Conceptual models enable people with low technical expertise to understand the meaning of data and to manipulate the data model and participate in the extension of DMSML for the specific purposes of their own organization; it thus encourages the transfer of business knowledge detained by the organization members to the DMSML metadata set. Conceptual UML class diagrams (Booch et al.,

1998) are used in order to represent concepts and relationships underlying the DMSML metadata model.
2. **Logical layer.** The logical layer translates domain-related concepts and relationships in data constructs, which are expressed in a rigorous and standard logical data modeling paradigm. Logical models are used by database designers to translate concepts into database constructs. The traditional paradigms for logical data modeling are the relational and object-oriented models (Elmasri & Navathe, 2003). Our approach aims to benefit from the use of XML, which at present is the standard for data serialization and exchange. XML defines a generic syntax used to mark up data in a text document, using simple and flexible tags. The grammar of XML documents can be defined by means of the XML schema language, as proposed by the W3C in its March 2001 XML Schema Recommendation (Sall, 2002). An XML document that respects the grammar rules of its XML schema is called a valid instance of that XML schema. In our case, DMSML is an XML schema that defines rules in order to represent document metadata and their values in an XML document.

DMSML BUILDING BLOCKS

After analyzing the role of documents in organizations, the content as well as the context of use are relevant properties of documents. DMSML is made of three building blocks: content-related properties are included in the Descriptive Information model, while context of use properties are expressed in the Collaboration and Lifecycle models:

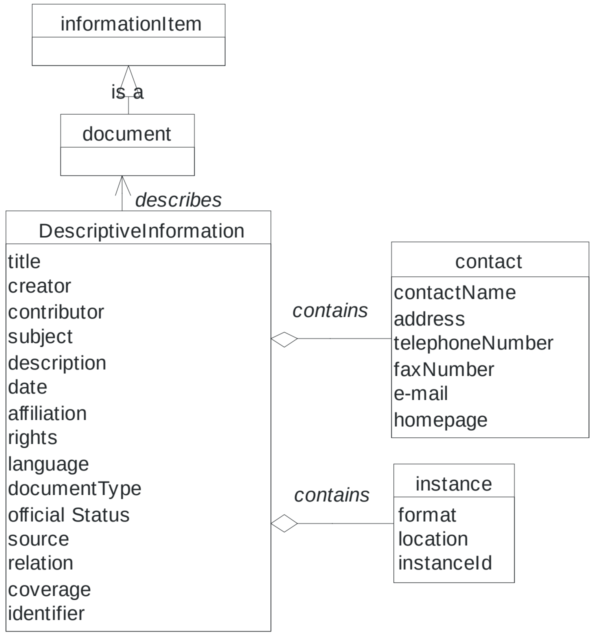
1. **Descriptive Information model.** Contains the set of metadata that describes and identifies the document, such as title, author, and subject.
2. **Collaboration model.** Formalizes how human resources are structured (the organizational model) and how access to information resources is regulated (the access right policy) on the basis of organizational roles or responsibilities of individuals.
3. **Lifecycle model.** Specifies the lifecycle of the document. The document lifecycle usually consists of the following stages: creation, review, publication, access, archive, and deletion. A specific lifecycle may not implement all these stages or implement others. This depends on the characteristics of the document types in use in the organizations. For instance, a project proposal lifecycle is different from the lifecycle of an official communication, which should be conveyed to the personnel of the organization at the final stage of its life cycle (the document passes through the states of draft, revised, final, and published).

DMSML also includes other concepts that represent the entities relevant for document management, such as the *Document*, which is the atomic unit of information that we consider; the *Folder*, which is a collection of documents; and the *Workspace*, which models the working environment and contains documents and folders. Each entity type may be described by specific labels of the DMSML model. In the following paragraphs we will show the conceptual views of the Descriptive Information, Collaboration, and Lifecycle models (Figures 1, 2, and 3), together with an example of metadata specification for our case study (a Project Proposal). Due to limitation of space, we provide an extract of the logical layer for the Descriptive Information model (Figure 2). For more details, the DMSML model is fully presented in Paganelli (2004).

Descriptive Information Model

The Descriptive Information model includes labels that provide descriptions of information resources. It mostly includes static properties, which generally are used for search and indexing purposes (e.g., Title, Creator, Date, Description, Document Type, Subject, Contact, Affiliation), as described in Figure 1. These labels are general enough to be applied to any

Figure 1. DMSML — Descriptive Information model, conceptual layer



document type and any organization. A subset of these labels also can be used to describe the workspace and folder entities (e.g., Title, Creator, and Description). DMSML can be extended in order to address the requirements of a specific organization by including descriptive elements that are tailored according to the specificities of a document type, organization, or organizational unit. For instance, a scientific paper can be characterized by further elements, such as Abstract, Journal and Publisher, while a document of type *Contract* should be associated with descriptive elements such as Contract Type, Customer and Product. We chose to integrate the Dublin Core (DC) (Dublin Core Metadata Initiative, 2003) metadata set in the Descriptive Information model, as it provides generic description labels, and it is a widely adopted standard. In addition to the DC elements, DMSML provides labels that are specific to the organizational environment; for instance, *Contact* (e.g., mail, telephone number, etc.) and *Affiliation* (e.g., project, department, organizational unit, etc.).

This conceptual view is mapped into the XML schema modeling primitives (see Figure 2) to provide the logical view of the model. Recalling the example of the project proposal mentioned previously, Figure 3 shows an instance of descriptive DMSML metadata for a specific project proposal document.

Collaboration Model

Document sharing can be considered a specific instance of collaborative activities. During the stages of document lifecycle, the participants collaborate for the accomplishment of organizational purposes. In this context, the DMSML Collaboration model is defined in terms of access policies assigned to the members of the organizational model, as shown in Figure 4. Access policies define the access rights to resources (Figure 4a). There are two dimensions related to permission assignment: a permission can be assigned to a specific employee or user (e.g., by means of a specific attribute, such as the name or an identifier), or the policy can be specified in terms of organizational entities and roles rather than to spe-

Figure 2. DMSML — Part of Descriptive Information model, logical layer

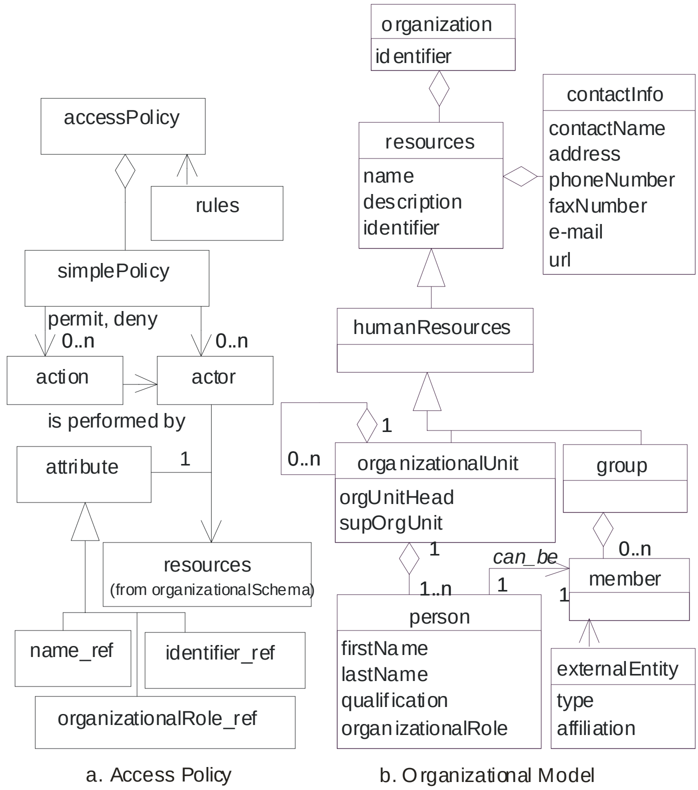
```
<?xml version="1.0" encoding="UTF-8"?>
  …
    <xs:element name="document" type="DocumentType"/>
      <xs:complexType name="DocumentType">
        <xs:sequence>
          <xs:element name="description" type="DescriptionType"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="DocumentDescriptionType">
        <xs:sequence>
            <xs:element ref="dc:title"/>
            <xs:element ref="dc:creator"/>
            <xs:element ref="dc:contributor"/>
            <xs:element ref="dc:subject"/>
            <xs:element ref="dc:description"/>
            <xs:element ref="dc:date"/>
            <xs:element ref="dc:rights"/>
            <xs:element ref="dc:language"/>
            <xs:element name="documentType" type="xs:string"/>
            <xs:element name="affiliation" type="xs:string"/>
            <xs:element name="contact" type="contactType"/>
            <xs:element name="identifier" type="xs:string"/>
                                    .....
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

Compliance with the Dublin Core (DC) metadata standard

Figure 3. DMSML specification for a project proposal — Descriptive Information model

```
<document>
        <title> Mobile Commerce Project Proposal </title>
        <creator> John Smith </creator>
        <contributor> Mark Johnson </contributor>
        <subject> wireless technologies, e-commerce </subject>
        <description> Proposal for Mobile Commerce services
                                  architecture</description>
        <date> 10/07/2004 </date>
        <rights> confidential</rights>
        <language> english</language>
        <documentType> Project Proposal</documentType>
        <affiliation> R&D Department </affiliation>
        <identifier> 0012223456</identifier>
        <contact>
        <contactName>John Smith</contactName>
        <address> my address  </address>
        <telephoneNumber> my telephone number </telephoneNumber>
        <faxNumber> my fax number </faxNumber>
        <e-mail> john.smith@mycompany.com </e-mail>
        <homepage> www.mycompany.com/johnsmith/ <homepage>
        </contact>
        .....
</document>
```

Figure 4. DMSML — Collaboration model, conceptual layer



cific participants (e.g., by means of the organizational role, attribute), according to the Role-Based Access Control (RBAC) strategy (Sandhu et al., 1996). We adopted the eXtensible Access Control Markup Language (XACML) (OASIS, 2003), allowing the description of access policies to information resources in an extensible and standard way.

Organizational models then map roles and organizational functions and units to individuals or groups (Figure 4b). DMSML includes an organization model that specifies the organizational units, individuals and related organizational roles. In order to satisfy changing requirements (e.g., the setup of a short-term project), it also may be extended to groups or external entities that are not institutional members of the organizational model but may be defined ad hoc for specific purposes and have a short life. The DMSML organizational model also can be connected with already existing Directory Services (e.g., LDAP) (Yeong et al., 1993) in order to synchronize common information.

For instance, an access policy can state that a project proposal can be accepted by a project manager of a specific organizational unit (Figure 5a). The organizational model makes it possible to associate these parameters (project manager, organizational unit name) to individuals' names or identifiers (Figure 5b). On the basis of this information, a software application can decide if an end user has the permission to accept the project proposal.

Lifecycle Model

The document lifecycle is a process specified in terms of a sequence of tasks performed by some actors, as shown in Figure 6. The execution of a task usually is triggered by

Figure 5. DMSML specification for a project proposal — Collaboration model

compliance with XACML

```
<PolicySet
  xmlns="urn:oasis:names:tc:xacml:1.0:policy".../>
<Policy PolicyId="doc_revision_Policy" ...>
  <Description> Project Proposal policy
                              </Description>
          ...
     <Action>
       <ActionMatch ...>
         <AttributeValue ...>accept</AttributeValue>
           <ActionAttributeDesignator .../>
         </ActionMatch>
       </Action>
       <Action>
         <ActionMatch ...>
         <AttributeValue ...>refuse</AttributeValue>
           <ActionAttributeDesignator .../>
           </ActionMatch>
       </Action>
     </Actions>
     </Target>
     <Rule RuleId="doc_revision_Rule"
                              Effect="Permit">
       <Subject>
         <SubjectMatch ...>
           <AttributeValue >Project Manager
                 </AttributeValue>
           <SubjectAttributeDesignator
               AttributeId="orgRole" … />
          </SubjectMatch>
       </Subject>
       ….
</Policy>
```

a.Access Policy

```
<organization>
  <name>My company</name>
  <resources>
          ....
     <person>
       <identifier>OP1</identifier>
       <personName>
         <firstName>Paul</firstName>
         <surname>Smith</surname>
       </personName>
       <contactInfo>
         <Name> Paul Smith</Name>
         <address> my address </address>
         <phone> +39 349 85553</phone>
          <fax> +39 055 4473324</fax>
          <e-mail>smith@mail.com</e-mail>
          <url>www.mycompany.com</url>
       </contactInfo>
       <qualification>Project mng.
                              </qualification>
       <OrgUnit>R&D</OrgUnit>
       <orgRole>ProjectManager</orgRole>
     </person>
          .....
   </resources>
</organization>
```

b.Organizational Model

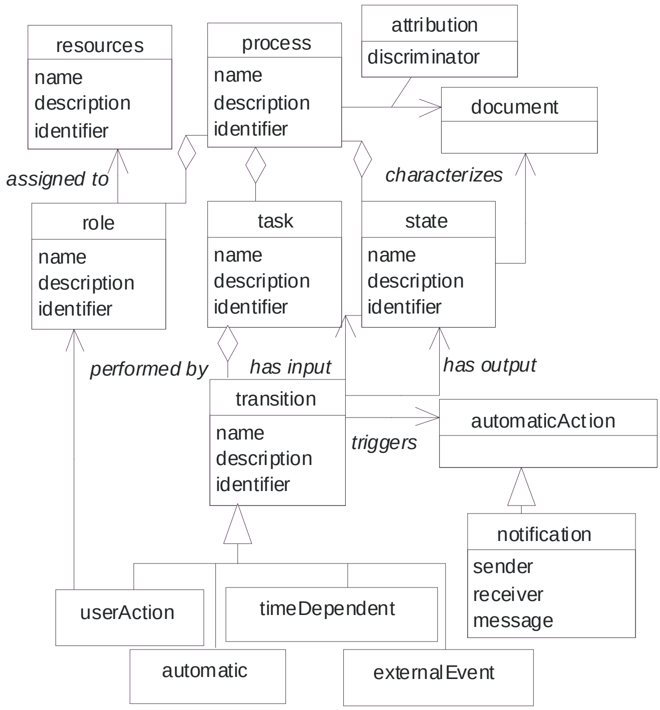
a transition condition, which can be automatic, time-dependent (e.g., a deadline) or caused by a user action or by an external event, and it is associated with an evolution of the document state (e.g., from draft to under_review, to final or under_revision). The DMSML model easily can be mapped to the Petri Net Markup Language (Weber & Kindler, 2002), which is a standard proposition for an XML-based format of Petri Nets, by means of XSL Transformations (Sall, 2002). DMSML thus can benefit from the use of the Petri Net process model, which has been proposed as a conceptual standard for the modeling and analysis of processes, thanks to its formal semantics, graphic notation, expressiveness, and abundance of analysis techniques.

For the sake of clarity, Figure 7a shows the lifecycle description for a project proposal document in the graphical Petri Net notation. Circles represent the states of documents (or *places* in the Petri Net language), and rectangles represent the transitions from one state to another. The lifecycle of the document is built upon the concatenation of these states and transitions. Figure 7b shows a part of the XML document that describes the project proposal lifecycle in the DMSML language.

DMSML BENEFITS FOR DMS DESIGN AND DEVELOPMENT

DMSML represents a comprehensive metadata model that encompasses content- and context-of-use document properties. DMSML

Figure 6. DMSML — Lifecycle model, conceptual layer



has been designed with the traditional metadata-design principles of generality, extensibility, and interoperability in mind and aims to address document management requirements with which, to our knowledge, existing metadata standards do not deal. As a matter of fact, DMSML contains labels that are applicable to any document type. Thanks to the standard extension mechanisms provided by XML and XML schema, the DMSML specification can be extended in order to deal with specific requirements. Interoperability is promoted by compliance with the following existing standards: Dublin Core, XACML, and PNML.

DMSML is a language that enables one to define open data and process models for the design, development, and operation of a DMS by means of a metadata-based approach. More specifically, DMSML enables the specification of functional requirements and their representation in an XML document for the design and deployment of a document management system. The DMSML provides XML elements to describe the configuration of the workspace or the folder structure in order to create or import a document resource classification schema, to specify the lifecycle and the access policies assigned to single documents, or on a document type basis. While generally most of these specifications are embedded in proprietary workflow engines or collaborative applications, DMSML provides a comprehensive framework that enables one to configure a DMS based on a declarative approach, minimizing the need for new code when the configuration needs to be modified. Thanks to its modeling principles,

Figure 7. DMSML specification for a project proposal — Lifecycle model

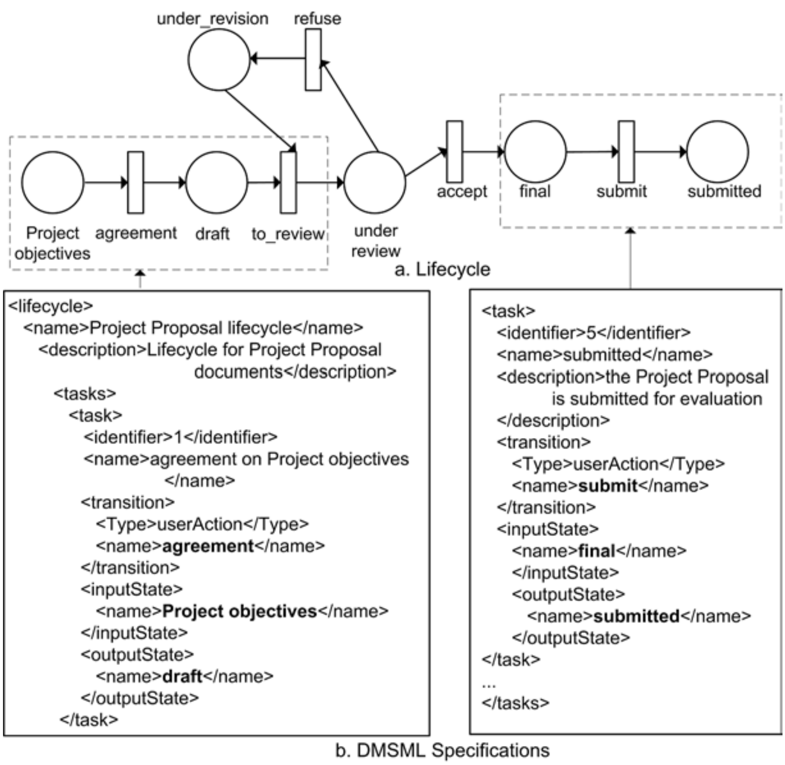
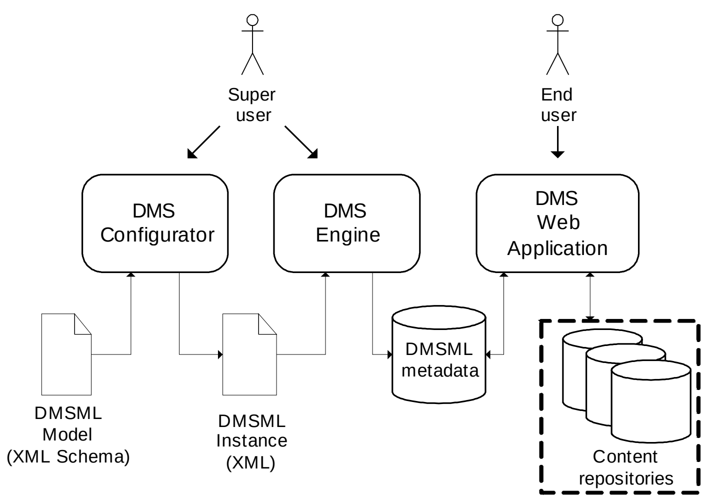


Figure 8. DMSML framework prototype architecture



DMSML facilitates the understanding of the metadata meaning and enables communication among heterogeneous communities. In such a way, business knowledge can be exploited effectively for the extension of DMSML metadata in a specific case. In order to validate the DMSML-driven method for DMS design and deployment and to provide some tools to support the DMSML methodological approach, we are developing a DMSML framework prototype.

DMSML FRAMEWORK PROTOTYPE

The DMSML framework prototype provides the user with automated support for adaptation and use of the metadata set and the design, deployment, and maintenance of a document management system. The DMSML framework consists of three parts: a DMS Configurator, a DMS Engine, and a DMS Web Application, as shown in Figure 8.

DMS Configurator

The DMS Configurator will be used by a sort of super-user (e.g., a system administrator), because its usage corresponds to the DMS specification and installation phase. This installation process is crucial for the definition of an instantiation of a DMSML-based document containing all information relevant to the proper document management of the specific organization. To this extent, the DMS Configurator acts through a wizard that will guide the user through the configuration of a DMS Web application according to organization requirements. The configurator application provides a sequence of interfaces that progressively guides the user through the definition of the workspace, the organizational schema, and the structure of folders. The end user can specify a set of lifecycle templates and access policies to be assigned to documents or document types.

The DMSML instance thus created — DMSML specifications — containing business and organizational information, such as organizational schemas and access policies, then will be processed by the DMS Engine.

DMS Engine

The DMS Engine is a Web-based application that enables the user accessing a standard Web browser to deploy a DMS that is customized according to the specifications encoded in the DMSML instance document. Again, the end user of this component of the prototype is a technical person or a system administrator.

Based on the features of the DMSML model, the DMS Engine enables a completely declarative approach for the design and deployment of an ad hoc document management system based on the specifications expressed in the DMSML instance. As shown in Figure 8, the declarative specifications in DMSML then are fed into the DMS Engine to produce tailored document management systems. These are built upon a collection of configurable components (Java classes) that provide the infrastructure for the new DMS. This infrastructure includes core libraries providing document management features.

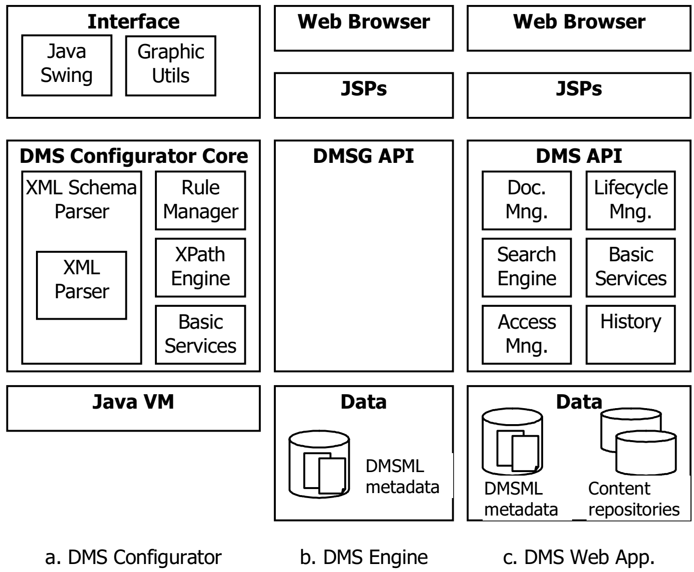
DMS Web Application

The DMS Web Application provides basic document management features that can be accessed by a generic end user (a member of the organization) through a standard Web browser. The DMS Web Application can be automatically configured and deployed, thanks to the facilities offered by the DMS Configurator and the DMS Engine applications.

The functions provided by the DMS Web Application include:

- Facilities for navigation, document upload, version control, and so forth. The interface provides end users with some information about the documents organized in the workspace folders, not only in terms of descriptive properties (e.g., title, creator, etc.) but also of lifecycle and sharing constraints.
- Document lifecycle management.
- Access control (ruling the execution of permitted actions according to the organizational access policies).

Figure 9. DMSML framework prototype: a. DMS configurator; b. DMS engine; c. DMS Web application



- Search functions enabling a metadata-based and full-text document search.
- Log files recording.

Based on the following formal foundations of DMSML, the DMS Web Application aims to provide a solution for document management that is capable of addressing the aforementioned high-level requirements:

- **Open standards Compliance.** The application is based on technical standards (e.g., J2EE and XML) and on business metadata standards as well (e.g., Dublin Core, XACML, and PNML).
- **Multiplatform compatibility.** As it is based on J2EE standard specifications.
- **Metadata-based approach.** Thanks to its extensive use of DMSML metadata.
- **Open and standard methodological approach.** For DMS design and development, thanks to the instrumental support provided by the DMSML framework prototype.

PROTOTYPE IMPLEMENTATION DETAILS

The DMS Configurator is a Java application. Its architecture consists of an *Interface*, which uses the JavaSwing Graphic Toolkit and other graphic utilities (e.g., images, etc.), and the *DMS Configurator Core* built on top of the *Java Virtual Machine*, as shown in Figure 9a. The DMS Configurator Core is composed of five main components:

- An XML Schema Parser, which verifies the validity of the XML document against the DMSML model. The XML Schema Parser also contains an XML parser.
- JDOM API, used to create, access and manipulate XML documents.
- An XPath Engine, enabling one to validate XPath expressions.
- A Rule Manager, which interprets and enforces the rules associated with the user actions.
- A set of Basic Services, such as logging and data storage facilities.

DMS Engine and DMS Web Application are both Web applications designed according to the standard J2EE specifications. They are characterized by a multi-tier architecture consisting of a Client, an Application Logic (composed of an Interaction and a Business Logic side), and a Data tier, as shown in Figures 9b and 9c, respectively. The Client side is a standard Web browser, and the Interaction side is realized by means of JSPs (Java Server Pages).

Regarding the DMS Engine, the Business Logic contains a template of a DMS Web Application and a set of APIs (DMSEngine API). The DMSEngine APIs are a set of Java classes that customize the template according to the DMSML specifications. The Business Logic of the DMS Web Application is composed of a set of DMS APIs implemented by Java classes, which provide basic functions for the management of workspaces, folders, and documents. The DMS APIs consist of several components:

- A Document Management component providing facilities for navigation, document upload, version control, and so forth.
- A Lifecycle Management component that enforces the evolution of the document across the lifecycle steps.
- An Access Manager, which should guarantee that users execute permitted actions according to the organizational access policies.
- A Search Engine, enabling a metadata-based and a full-text document search.
- A History component that records log files.
- Basic Services, such as monitoring and connection to database services.

CONCLUSION AND FUTURE WORK

This article has highlighted some critical issues concerning unstructured document management in organizations, giving sound arguments to increase the awareness of what managers should account for when considering the possibility of adopting a document management system in order to fulfill their organizational needs. In particular, we defended the idea that metadata are the basis for novel strategies and solutions for effective document management. However, since existing metadata specifications are too generic or focused on application domains other than document management, we have proposed in this article a metadata model for the management of unstructured documents in organizations, enabling the design and deployment of DMSs.

The DMSML metadata includes descriptive, collaboration- and process-related properties of unstructured documents. The DMSML modeling approach presented here enables the representation of the metadata set in a way that promotes human and machine understanding by separating and properly representing metadata semantics and syntax.

We also described the DMSML Framework Prototype. The prototype includes a Web-based DMS (based on the DMSML Descriptive Information, Collaboration, and Lifecycle models), a DMS Configurator, and a DMS Engine, enabling end users to specify business requirements to properly configure and customize the DMS for the needs of a specific organization by exploiting the features of the DMSML model.

At the moment, testing activities of the prototype in real application scenarios are undergoing. The evaluation activity aims to verify the capability of the DMSML-based solution, based on the DMSML model and the DMSML Framework Prototype, to address critical factors related to the management of unstructured documents in an organization. We expect that the results of the evaluation provide a measure of the impact of a DMSML-based approach in an application scenario and suggestions for the refinement of the DMSML labels. The testing phase aims also at evaluating the degree of usability and user satisfaction provided by the DMSML Framework Prototype. The evaluation should focus conveniently on those organizations that already have adopted process optimization and re-engineering strategies (i.e., organizations with quality certifications). In this case, the DMSML-based DMS configuration could support predefined organizational processes and document lifecycles. At present, first stages of evaluation are being carried out in an Italian SME (Small Medium Enterprise).

During the evaluation phase, the cost of using metadata also will be analyzed. As illustrated throughout this work, the use of metadata has several benefits, but its cost has to be evaluated in terms of resources needed for metadata creation and management (Duval et al., 2002; Gilliland-Swetland, 2000). Metadata creation and insertion in the system can require a manual or automatic procedure, or a hybrid approach, according to the type of metadata (i.e., *creator* and *date* values can be automatically inserted by the system). The optimal strategy should be found in order to obtain the best compromise of accuracy and efficiency.

At the current stage of the prototypal development of the DMS Web Application, the end user should manually insert most descriptive metadata. Further work is needed in order to investigate strategies and mechanisms in order to automate this stage as much as possible (i.e., using content analysis tools for the automatic extraction of metadata from texts).

REFERENCES

Booch, G., Jacobson, I., & Rumbaugh, J. (1998). *Unified modeling language user's guide*. Boston, MA: Addison-Wesley.

Crawford, W. (1989). *MARC for library use: Understanding USMARC***.** Boston, MA: G.K. Hall.

Dublin Core Metadata Initiative. (2003). Dublin core metadata element set, Version 1.1: Reference description. Retrieved May 20, 2003, from http://www.dublincore.org

Duval, E., Hodgins, H., Sutton, S., & Weibel, S. (2002). Metadata principles and practicalities. *D-Lib Magazine*, *8*(4). Retrieved June 20, 2003, from http://www.dlib.org/dlib/april02/weibel/04weibel.html

Elmasri, R., & Navathe, S. B. (2003). *Fundamentals of database systems*. Addison Wesley.

Emery, P. (2003). *Documents and records management: Understanding the differences and embracing integration*. Retrieved September 16, 2004, from http://www.zylab.com

Gilliland-Swetland, A. J. (2000). *Setting the stage. Introduction to metadata: pathways to digital information*. Retrieved March 18, 2003, from http://www.getty.edu

Ginsburg, M. (2000). Intranet document management systems as knowledge ecologies. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS-33)*. IEEE Computer Society Press.

Ginsburg, M. (2001). Openness: The key to effective intranet document management. In *Proceedings of International Symposium on Information Systems and Engineering ISE'2001*. Las Vegas, Nevada.

Gonçalves, M. A., Fox, E. A., Watson, L. T., & Kipp, N. A. (2004). Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries. *ACM Transactions on Information Systems*, *22*(2), 270-312.

IMS Global Learning Consortium. (2003). IMS learning design v1.0 final specification. Retrieved December 15, 2003, from http://www.imsglobal.org/learningdesign/index.cfm

Institute of Electrical and Electronics Engineers. (1990). *IEEE standard computer dictionary: A compilation of IEEE standard computer glossaries*. New York: Institute of Electrical and Electronics Engineers.

International Electrotechnical Commission. (2001). *Document management. Part 1: Principles and methods* (IEC 82045-1:2001). Retrieved July 15, 2004, from http://www.iso.org

International Organization for Standardization. (2001). *Information and documentation. Records management. Part 1: General* (ISO 15489-1). Retrieved October 22, 2003, from http://www.iso.ch

Karjalainen, A., Päivärinta, T., Tyrväiinen, P., & Rajala, J. (2000). Genre-based metadata for enterprise document management. In *Proceedings of the 33rd Hawaii's Conference* *on System Sciences, HICSS* (pp. 3013-3023). Los Alamitos CA: IEEE Computer Society.

Lagoze, C., & Hunter, G. (2001). The abc ontology and model. In *Proceedings of the International Conference on Dublin Core and Metadata Applications* (pp. 160-176). Retrieved July 15, 2003, from http://www.nii.ac.jp/dc2001/proceedings/abst-26.htm.

Manjunathn, B. S., Salembier, P., & Sikora, T. (2002). *Introduction to MPEG-7 multimedia content description interface*. New York: John Wiley & Sons.

Melnik, S., Decker, S. (2000). A layered approach to information modeling and interoperability on the Web. In *Proceedings of the ECDL'00 Workshop on the Semantic Web*, Lisbon, Portugal.

Mitra, N. (2003). *SOAP Version 1.2 Part 0: Primer* (W3C Recommendation). Retrieved October 2003, from http://www.w3.org/TR/soap12-part0/

Moore, C., & Markham, R. (2002). *Enterprise content management: A comprehensive approach for managing unstructured content*. Giga Information Group, Inc. Retrieved February 2003, from http://www.msiinet.com/html/pdfs/essecm3.pdf

Murphy, L. D. (1998). Digital document metadata in organizations: Roles, analytical approaches, and future research directions. In *Proceedings of the 31st Hawaii International Conference on System Sciences* (Digital Documents) (pp. 267-276).

Murray, P. C. (1996). Information, knowledge and document management technology. *KM Metazine*. Retrieved July 10, 2003, from http://www.ktic.com/topic6/12-INFKM.HTM

OASIS. (2003). *Extensible access control markup language (XACML), v. 1.0*. Retrieved August 10, 2003, from http://www.oasis-open.org

Object Management Group. (2001). *Common warehouse metamodel specification, v 1.0*. Retrieved March 15, 2003, from http://www.omg.org/cgi-bin/doc?formal/01-10-01

Paganelli, F. (2004). *A metadata model for unstructured document management in organizations*. Doctoral dissertation, University of Florence, Department of Electronics and Telecommunications, Florence, Italy.

Päivärinta, T. (2001). *A genre-based approach to developing electronic document management in the organization*. Doctoral dissertation, University of Jyvaskyla, Finland.

Päivärinta, T., Tyrväiinen, P., & Ylimäki, T. (2002). Defining organizational document metadata: A case beyond standards. In *Proceedings of the Xth European Conference on Information Systems (ECIS)*.

Public Record Office. (2001). *E-government policy framework for electronic records management*. Retrieved September 2004, from http://www. pro.gov.uk/recordsmanagement/

Sall, K. B. (2002). *XML family of specifications: A practical guide*. Boston, MA: Addison-Wesley Professional.

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

Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. *IEEE Computer, 29*(2), 38-47.

Sprague, R. H., Jr. (1995). Electronic document management: Challenges and opportunities for information systems managers. *MIS Quarterly, 19*(1), 29-50. Retrieved September 20, 2003, from http://www.cba.hawaii.edu/sprague/MISQ/MISQfina.htm

Stickler, P. (2001). Metia — A generalized metadata driven framework for the management and distribution of electronic media. In *Proceedings of the Dublin Core Conference 2001* (pp. 235-241).

Sun Microsystems. (2003). *J2EE 1.4 specification: Final release*. Retrieved November 2003, from http://java.sun.com/docs/index.html

the451. (2002). *Unstructured data management: The elephant in the corner*. Retrieved March 20, 2003, from http://www.the451.com

Weber, M., & Kindler, E. (2002). The Petri Net markup language. Advances in Petri Nets. *LNCS series*. Springer Verlag.

Workflow Management Coalition (2002). *Workflow process definition interface: XML process definition language, v. 1.0*. Retrieved March 16, 2003, from http://www.wfmc.org

Yates, J., & Orlikowski, W. J. (1992). Genres of organizational communication: A structurational approach to studying communication and media. *Academy of Management Review, 17*(2), 299-326.

Yeong, W., Howes, T., & Kille, S. (1993). *X.500 lightweight directory access protocol* (IETF RFC 1487). Retrieved October 15, 2002, from http://www.ietf.org/rfc/rfc1487.txt

ENDNOTES

1. www.documentum.com, www.fatwire.com, www.ibm.com, www.interwoven.com, www.microsoft.com
2. www.zope.org, www.opencms.org, cocoon.apache.org/lenya, www.dspace.org, www.dlib.vt.edu

Federica Paganelli is research scientist at the Electronics and Telecommunications Department, University of Florence. She holds a PhD in telematics and information society from the University of Florence. Her research interests include organizational information management, information system architecture, technologies, and design methodologies. She has a special interest in theories, methods and tools for metadata management.

Maria Chiara Pettenati is assistant professor at the Telematics Laboratory of the Electronics and Telecommunications Department of the University of Florence since late 2004. Until 2004 she held a post-doctoral research position in the same laboratory. In 2000 she received the PhD grade in telematics and information society granted from the University of Florence. From 1997 to 1999, she has been working at the Computer Science Department (DI - LITH Laboratory) of the (Swiss Federal Institute of Technology) EPFL. Her main research interests concern network architectures for trust-enabling and personalized services delivery, e-learning, and e-knowledge applications.

Dino Giuli is full professor in telecommunications at the Department of Electronics and Telecommunications of the University of Florence where he was director of the Department until 2004. Since 1996 he is promoter and scientific co-ordinator of the PhD program in telematics and information society. His research activities have been principally oriented in the domain of telematics and environmental monitoring systems. Dr. Giuli is author of more than 150 scientific publications, mainly published on scientific journals and international conference proceedings. He is an AEI member and IEEE senior member.