Conceptual Modeling of Electronic Content and Documents in ECM Systems Design: Results from a Modeling Project at Hoval

Alexander Simons, Jan vom Brocke, Stefan Fleischer and Jörg Becker

Abstract The implementation of enterprise content management (ECM) software requires careful analysis of an organization's content and document assets, and conceptual information models can provide substantial input for ECM systems design. In particular, content models can support the documentation of both organizational and technological conditions and can illuminate software-related requirements. Therefore, a conceptual modeling language for electronic content and documents has to meet several conditions: It should facilitate description of how content can be reused in different documents, the creators and users of content, and the software systems involved. In addition, given the vast number of digital assets created and used in today's organizations, such a language has to safeguard a clear and consistent representation while also being ready for efficient adaptation and maintenance. With the help of the general criteria of conceptual modeling proposed by Becker et al. (e.g., correctness, relevance, clarity), this chapter identifies these and related requirements and argues that they are not sufficiently met by existing modeling approaches. As a response, we propose a novel modeling language that we developed and evaluated during the course of a modeling project at Hoval, to be used in describing electronic content and documents.

A. Simons (✉) · J. vom Brocke
Institute of Information Systems, University of Liechtenstein, Fuerst-Franz-Josef-Strasse 21, 9490 Vaduz, Liechtenstein
e-mail: alexander.simons@uni.li

J. vom Brocke
e-mail: jan.vom.brocke@uni.li

S. Fleischer · J. Becker
European Research Center for Information Systems, University of Muenster, Leonardo-Campus 3, 48149 Muenster, Germany
e-mail: stefan.fleischer@ercis.uni-muenster.de

J. Becker
e-mail: becker@ercis.uni-muenster.de

J. vom Brocke and A. Simons (eds.), *Enterprise Content Management in Information Systems Research*, Progress in IS, DOI: 10.1007/978-3-642-39715-8_14,

Introduction

In the era of digitization, companies are confronted with an avalanche of information, and managing the tremendous amount and variety of electronic and other content requires, among other things, maintaining its timeliness and consistency (vom Brocke et al. 2010, p. 3). For example, product descriptions are often present in several types of materials, including instruction manuals, technical specifications, sales catalogues and presentations, and marketing brochures and flyers (compare Rockley et al. 2003, pp. 4–6). Particularly in the early stages of the product lifecycle, such content is subject to frequent change, which can have important economic effects, as updating all of the relevant documents may be difficult and time-consuming at an enterprise-wide level. Because many of these documents make promises about products and services, outdated and/or inconsistent documentation can lead to problems with customers and can even have legal consequences (vom Brocke et al. 2011b, p. 970).

Numerous approaches to and software solutions for dealing with these or similar challenges have been discussed in research and practice. Document management systems, which primarily serve to store and retrieve files, are of only limited help to companies. More promising are solutions for content management that allow content to be handled independently of its structure and presentation (Boiko 2002, pp. 135–137; Clark 2007, pp. 44–45) so textual or graphical content from different containers, such as Web pages and documents, can be reused efficiently (O'Callaghan and Smits 2005, pp. 1272–1274), increasing efficiency in the process of creating documents and keeping them up-to-date (e.g., when content is revised, extended, or translated) (Rockley et al. 2003, pp. 24–26). Since the turn of the millennium, the academic discipline of IS has addressed the issue of company-wide management of electronic content and documents under the umbrella term "ECM" (Päivärinta and Munkvold 2005; Smith and McKeen 2003; Tyrväinen et al. 2006; vom Brocke et al. 2011a), an integrated concept that supports the management of all possible forms of information across their entire lifecycle (Smith and McKeen 2003, p. 648; vom Brocke et al. 2011b, pp. 967–968).

ECM systems offer a number of benefits to companies, such as meeting retention requirements, improving information quality, capturing and disseminating knowledge, and supporting collaboration within and between organizations (Päivärinta and Munkvold 2005, pp. 2–3). At the same time, however, organizations that adopt ECM face challenges in selecting the right software, implementing efficient workflows, defining and maintaining metadata and corporate taxonomies, and training and motivating the staff involved (Munkvold et al. 2006, pp. 75–84). The process of implementing ECM in a company begins with identifying and analyzing the company's existing documents and content, a process commonly referred to as a "content audit" (Rockley et al. 2003, pp. 104–105). Having an in-depth understanding of a company's business documents and content facilitates the best possible selection and customization of ECM systems (O'Callaghan and Smits 2005, p. 1275).

The results of a content audit can be documented in the form of conceptual models that can then serve as starting point for ECM systems implementation. Conceptual content models help the creators and users of content by describing and illustrating the software systems involved and how content is reused in different documents. As such, content models may facilitate communication between the ECM project team and the users who will work with the system, revealing the demand for support on the system's side in particular (Kung and Solvberg 1986; as cited in Wand and Weber 2002, p. 363). Thus, content models can provide a roadmap to the company's increasingly complex content and system landscapes that often grow over many years.

Therefore, a modeling language has to meet several requirements, which are identified in this paper with the help of the general modeling principles from Becker et al. (1995), including correctness, relevance, clarity, comparability, and efficiency (pp. 437–439). The requirements we identify are not sufficiently met by traditional approaches, so a novel modeling language, which was developed and evaluated during the course of a modeling project at Hoval, is proposed.

The research project described in this chapter followed the design science paradigm (Hevner 2007; Hevner et al. 2004; March and Smith 1995), in particular the design research methodology from Peffers et al. (2008, p. 54). In the next section, we provide a background for the chapter, and on that basis, identify the requirements a modeling language for electronic content and documents must meet. Then we present the proposed modeling language based on a meta-model and summarize the results from the modeling project at Hoval. The results of the project confirm the utility of the modeling language but also indicate the need for future research, which is outlined in the last section.

Background

ECM remains an elusive concept that lacks a theoretically sound foundation (vom Brocke et al. 2011a, pp. 478–480). However, for the purposes of this chapter, ECM is understood as a modern and integrated approach to digital information management (Päivärinta and Munkvold 2005, p. 1). The integrative nature of ECM has at least three parts (vom Brocke et al. 2011b, pp. 967–968): First, ECM refers to the management of all of an organization's information assets, regardless of type, format, granularity, or source. Second, ECM includes both technological (e.g., software, hardware, and standards) and managerial (e.g., strategies, methods, and processes) capabilities. Third, ECM covers the management of information over its entire lifecycle. As such, the concept of ECM includes "the strategies, tools, processes and skills an organization needs to manage all its information assets (regardless of type) over their lifecycle" (Smith and McKeen 2003, p. 648).

Developing and implementing such a comprehensive approach to managing content and documents is often a highly complex and time-consuming endeavor (White 2002, p. 22). In a longitudinal case study, Munkvold et al. (2006) identified a number of challenges companies face when implementing ECM systems (pp. 75–84). As these challenges are both technological and managerial in nature, Tyrväinen et al. (2006) characterize ECM as a research topic relevant to the IS discipline (p. 628).

Tyrväinen et al. (2006) present a framework for ECM research (pp. 628–631), which is used here to explain the salient role content audits and models play in ECM implementation. The framework distinguishes four perspectives to which questions relevant to IS research can be assigned: *processes, content, technologies,* and the *enterprise context* (Tyrväinen et al. 2006, p. 628). The four perspectives should be viewed from an integrated perspective, not separately. For example, developing and implementing efficient *processes* that support different phases of the *content* lifecycle requires careful selection and alignment of ECM *technologies* and consideration of the legal aspects relevant to content management in the *enterprise context* (e.g., long-term retention).

While Tyrväinen et al. (2006) designed their framework to stimulate and guide future research in the field of ECM (p. 627), the framework's core ideas can also be transferred to the practical implementation of ECM systems. The content perspective represents the core of the framework, as, "in any piece of ECM research, the content perspective is involved in some way" (Tyrväinen et al. 2006, p. 628). At the outset of any ECM initiative, organizational documents and their content should be closely examined, as they substantially affect the managerial and technological requirements of the three remaining dimensions of processes, technologies, and the enterprise context. Only after a thorough analysis of the content is it possible to determine the relevant legal, economic, and social factors at the level of the enterprise, how existing processes have to be redesigned, and the technologies needed to meet all these requirements (vom Brocke et al. 2011a, pp. 483–484). However, the analysis of content is often a challenge for companies to address, as multiple factors must usually be considered from the content perspective. O'Callaghan and Smits (2005) list a number of questions that should be answered in a content audit including questions related to the types of content that are present in the organization, who is responsible for it, who uses the content, how and where the content is reused and repurposed, which content is to be retained and in what form, and which systems are to be used for the creation and processing of content (p. 1275).

Given the enormous amount and variety of electronic content in today's enterprises, the answers to these questions may be elusive. Unlike structured data, which is usually well-documented, semi-structured or unstructured information is often described to only a limited extent. While document overviews may be present in the form of, for example, tables, these overviews usually answer the central questions only partially and/or they refer only to specific departments, business functions, or processes. Documentation of the results of a content audit is also aggravated, as documents, content, an organization's information needs, and content-related processes are dynamic. Therefore, an approach that allows organizations to acquire, analyze, report, and maintain the information they need to understand the creation and use of content at an enterprise-wide scale is needed. Conceptual information models can serve this purpose.

Content Modeling Requirements

A modeling language for electronic documents and content has to meet several requirements. Any of a number of general approaches to evaluating conceptual models can be used to identify the requirements of a modeling language for electronic documents and content. Many of these approaches refer to data modeling (Frank 1997, pp. 97–98), particularly entity relationship models. For example, Moody and Shanks (1994) propose six criteria for assessing the quality of a data model: simplicity, completeness, flexibility, integration, understandability, and implementability (p. 101). Kesh (1995) presents a framework for evaluating entity relationship models, taking into consideration the structure (suitability, soundness, consistency, conciseness) and contents (completeness, cohesiveness, validity) of data models (pp. 681–685). Genero et al. (2000) examine the maintainability of entity relationship models, differentiating between understandability, legibility, simplicity, analyzability, modifiability, stability, and testability (p. 514).

Similar to Frank (1997), who distinguishes criteria inherent to the model (e.g., completeness and non-redundancy) and discusses the relationship of a model with the observer, with reality, and with the modeling purpose (pp. 98–99), Becker et al. (1995) present a general approach to evaluating conceptual models ("Grundsätze ordnungsmäßiger Modellierung") using the criteria of correctness, relevance, efficiency, clarity, and comparability[1] (pp. 437–439; own translation). Our requirements for a modeling language for electronic documents and content are derived from Becker et al.'s (1995) five general criteria (pp. 437–439).

Correctness. Becker et al. (1995) distinguish between the syntactical and semantic correctness of a conceptual model (p. 437). While the former refers to the consistent use of the model elements and notation rules defined in the meta-model, semantic correctness is related to the model's structural and behavioral compliance with the object system it represents (Becker et al. 1995, pp. 437–438). Accordingly, a modeling language for electronic documents and content has to support modeling on both syntactically and semantically correct levels. On the semantic level, the language should particularly allow the illustration of reuse of content in different documents, the correct assignment of user roles and rights, and the consideration of the software systems involved to be described. For its part, developing a syntactically correct content model requires a meta-model that describes all of the relevant model elements and the possible relationships among them. The modeling language should also provide options that help prevent the language defects (e.g., the use of synonyms) that can occur when the same content

[1] With *systematic structure*, Becker et al. (1995) propose another quality criterion of conceptual modeling, acknowledging that information models are typically put up for different views that must be integrated (e.g., data, process, and functional views) (p. 439). Although this criterion appears also to be relevant in the context of content modeling (e.g., documents are typically part of process models), it is outside the scope of this chapter.

or document objects are part of more than one sub-model (e.g., for certain business functions and areas).

Relevance. Relevance generally refers to a model's goal orientation; model elements are relevant if their exclusion reduces the overall benefit of the model (Becker et al. 1995, p. 438). For example, a model's relevance is closely related to its level of abstraction; that is, models that feature higher degrees of completeness or exactness than others do are likely to cover increased amounts of irrelevant information (Becker et al. 1995, p. 438). Because of the enormous number and variety of documents that must be handled in today's enterprises, the level of abstraction is an important criterion in content modeling. In addition, a modeling language for electronic documents and content must meet the requirements of multiple model users with diverse goals. The potentially relevant content modeling goals include both organizational goals (e.g., in the case of reorganization projects) and technology-related goals (e.g., in the case of implementation projects). Therefore, model developers must be given considerable freedom in the process of content modeling.

Efficiency. Whereas relevance generally refers to the scope of a model (i.e., the results view), efficiency refers to the effort that must be undertaken in the course of the modeling process (i.e., the process view) (Becker et al. 1995, p. 438). The efficiency of content modeling is particularly determined by the frequency with which the developed models must change. Documents and content, as well as information needs and software systems, are constantly changing in today's organizations, so a modeling language for electronic documents and content should ensure efficiency with regard to model development and with regard to model adaptation. The support from appropriate modeling tools may substantially reduce the effort required for model development and adaptation.

Clarity. A language's clarity, which refers to a model's structure and readability, depends largely on the subjective perception of the model user (Becker et al. 1995, p. 438), so it is particularly determined by the graphical notation of a modeling language. Model clarity is also closely related to the principle of correctness because higher degrees of model completeness or exactness are likely to result in reduced model clarity (Becker et al. 1995, pp. 438–439). Therefore, the assignment of attributes to model elements (e.g., metadata for content storage and retrieval), for example, should not too severely impair the model's readability. Furthermore, the clarity of a content model is important to the ability to consider organizational and technological content management requirements equally. Whereas content creators and users typically know more about organizational conditions than they do about technological implications, the adopters of ECM systems are more likely than content creators to be aware of available and required software features. To facilitate communication between the ECM project team and the users, a content model should be equally comprehensible for both groups (i.e., business and IT). The development of sub-models, each of which has its own scope, may be able to take into account the heterogeneity of modeling goals and model users, which would contribute to the relevance of content modeling. Clarity is also an issue when the modeling language is to be extended (e.g., by new model elements).

Comparability. Comparability, like correctness, has both syntactical and semantic dimensions. While the syntactical aspect refers to the compatibility of models developed with different modeling methods and languages, the semantic aspect refers to the comparability of different models at the content level (e.g., as-is model vs. to-be model) (Becker et al. 1995, p. 439). While the syntactical dimension is out of the scope of this chapter, the notion of semantic comparability is especially important in content modeling to support the consolidation and integration of content models developed for multiple company divisions and functional areas.

Discussion of Existing Modeling Approaches

The development and evaluation of conceptual modeling languages and methods is a core topic of design-oriented IS research (Fill et al. 2007, p. 419). During the course of our modeling project, we evaluated some of the approaches for information and data modeling that have been discussed in the community with regard to their applicability for enterprise-wide content modeling. This evaluation was based on the modeling requirements summarized above using questions such as whether all element types relevant to content modeling are provided by the languages, how much time and effort would be needed to develop and adapt the content models, whether there is suitable tool support for the adaptation of the modeling languages, whether the clarity, structure, and readability of content models developed with these languages is sufficient, and whether there is sufficient freedom in the process of content modeling.

The results of the review suggest that existing approaches to information and data modeling have only a limited applicability to the modeling of content and documents at an enterprise-wide level. While many approaches are likely to result in the development of both syntactically and semantically correct content models, they do not sufficiently meet the other requirements of conceptual content modeling, particularly the ones regarding clarity and efficiency. As of today, there are but a few specific approaches to modeling electronic documents and content at an enterprise-wide level. For example, Rockley et al. (2003) propose a table-based approach to content modeling that aims at documenting the reuse of content (pp. 159–182) by distinguishing among semantic data (e.g., subject, date, or contact), basic data (e.g., title or container), and architecture-related data (e.g., type of reuse) (Rockley et al. 2003, p. 175). Existing approaches to content modeling focus on specific areas of information management (e.g., training material or software guides) or are developed for certain industries (e.g., telecommunication or pharmaceutical), while more generic approaches with enterprise-wide scopes are rare (Rockley et al. 2003, p. 177). As a response, the next section presents a modeling language for electronic documents and content developed in consideration for the requirements and criteria presented above.

Modeling Language

The purpose of conceptually modeling content can be support for organizational goals (e.g., information lifecycle management) or technical goals (e.g., ECM systems selection and customization). Examples of content modeling goals include creating transparency in information management, documenting the reuse of content, detecting shortcomings in the process of document creation, and eliminating media disruptions. These goals must be clearly defined at the outset of any content modeling initiative because they largely determine the components a content model later requires, particularly the elements that are used in the modeling process. For example, while some implementation projects may demand detailed descriptions of technical requirements (e.g., system functionality, metadata, user rights and roles), such requirements may be of minor relevance in reorganization projects. The modeling language presented in this section, *Enterprise Content Modeling Language* (*ECML*), is explained with regard to two general modeling goals: conceptually describing the reuse of content in different documents and describing the creation and use of content by different users and systems over the content's lifecycle.

ECML features ten basic types of model elements, which are described by means of an entity relationship meta-model in Fig. 1 (gray boxes). The model elements, which are hierarchically structured, can be further specified according to their attributes. The figure also shows the elements' graphical notations (which are connected with the element types by dotted lines). "Content Types," describe content at a general level; they can, as a function of granularity, be self-contained (e.g., an image) or composed of other Content Types (e.g., a product description that contains an image). In contrast, "Content Assets" represent specific content (e.g., an image of a certain product), so Content Assets are always assigned to at least one Content Type. As it is the case for Content Types, Content Assets may

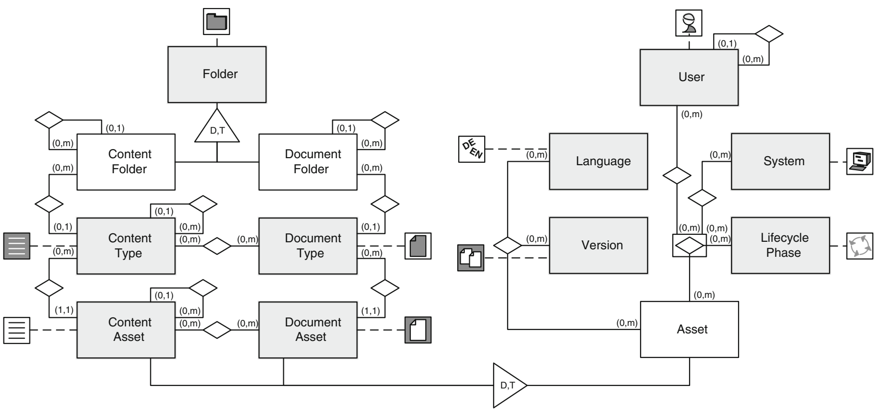

Fig. 1 Entity relationship meta-model for *ECML*

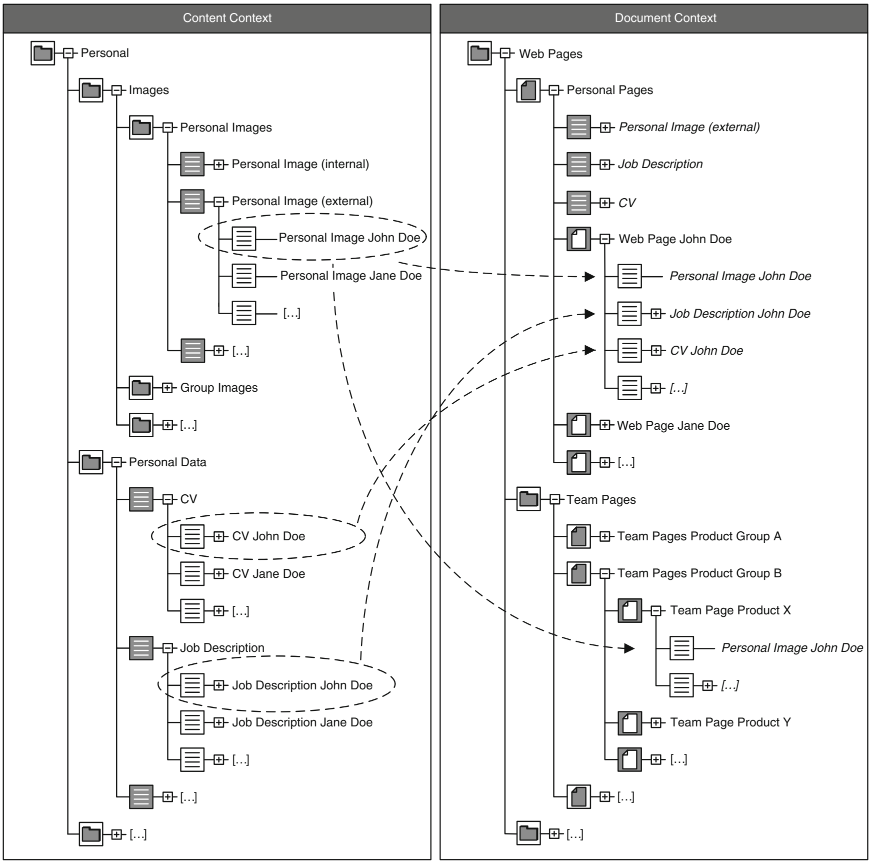

Fig. 2 Example of an *ECML* model (1)

vary in terms of their granularity. In contrast, the granularity can be disregarded for "Document Types" and "Document Assets," which represent complete information products, although they can otherwise be understood analogous to Content Types and Content Assets. Accordingly, Document Types represent general documents (e.g., sales catalogues), while a Document Asset represents a specific occurrence of a Document Type (e.g., a sales catalogue of a certain year). In addition, Document Types and Document Assets may be composed of a number of Content Types and Content Assets (e.g., a sales catalogue that contains multiple product descriptions and images). Both element types can be organized in "Folders."

The element types described so far can be applied to describe the reuse of content. To do so, two general modeling contexts need to be distinguished: "Content Context" and "Document Context." (See Fig. 2 for a fictitious, simplified example.) In the example above, the Content Context covers various pieces of personnel-related content that can be grouped into »Images« and »Personal Data«. A distinction is made between »Personal Images« and »Group Images« both of which may be used for either internal or external purposes. For example, »Personal Image (external)«, is a self-contained Content Type that subsumes under it a number of specific personal images (i.e., Content Assets, such as »Personal Image John Doe« or »Personal Image Jane Doe«). »Personal Data« is a folder under which »CVs« and »Job Descriptions« are organized and which are modeled as composed Content Types (displayed by the + symbols). Both Content Types and Content Assets are reused in the Document Context, which, in the example, describes the content of »Web Pages«, including »Personal Pages« and »Team Pages«. The content model illustrates that a Document Type determines of which Content Types an information product is generally composed (e.g., »Personal Page« = »Personal Image (external)« + »Job Description« + »CV«) and specifies the concrete Document Assets that instantiate it (e.g., »Web Page John Doe« = »Personal Image John Doe« + »Job Description John Doe« + »CV John Doe«). However, Document Assets do not necessarily have to contain a model element for each Content Type of which the represented Document Type is composed (e.g., not all »Personal Pages« must contain a concrete »Job Description«). This is also true, of course, for composed Content Assets. In the example, the reuse of content (e.g., »Personal Image John Doe«) in various documents (e.g., »Web Page John Doe« and »Team Page Product X«) is illustrated by dashed arrows and by Content Assets written in italics.

However, content models cannot be used only to document the reuse of content in organizations but can also help clarify which documents and content are used by which individuals ("Users") and with the help of which software products ("Systems"). Such information can, for instance, support the selection and customization of ECM systems. As documents and content are created and used by means of a variety of systems over the documents' and contents' life spans (e.g., a marketing brochure may be created by means of a graphics software, then converted into a pdf file, and finally published on the Internet), *ECML* also distinguishes among several "Lifecycle Phases." The combination of Lifecycle Phases, Systems, and Users with specific Document Assets then describes who is responsible for the creation, editing, and publication of a certain document and what software is used in the process. Figure 3 provides another simple example, extending the Content Context from Fig. 2 by the model elements mentioned.

Finally, *ECML* also allows languages and versions to be assigned to certain documents and content. The assignment of "Versions" and "Languages" to certain Document Assets and Content Assets, respectively, then represents a real-world document or content asset (e.g., version 2.0 of the German user manual for a certain product). Alternatively, attributes could be assigned to Content Assets and Document Assets for this purpose. Examples of common attributes that may also represent the metadata required for storage and retrieval, are »ID«, »Notes«, »Status« (e.g., "in progress", "under review", and "in translation"), »Creation Date« and »Editing Date«, »Creator«, »Editor«, and »Owner«.

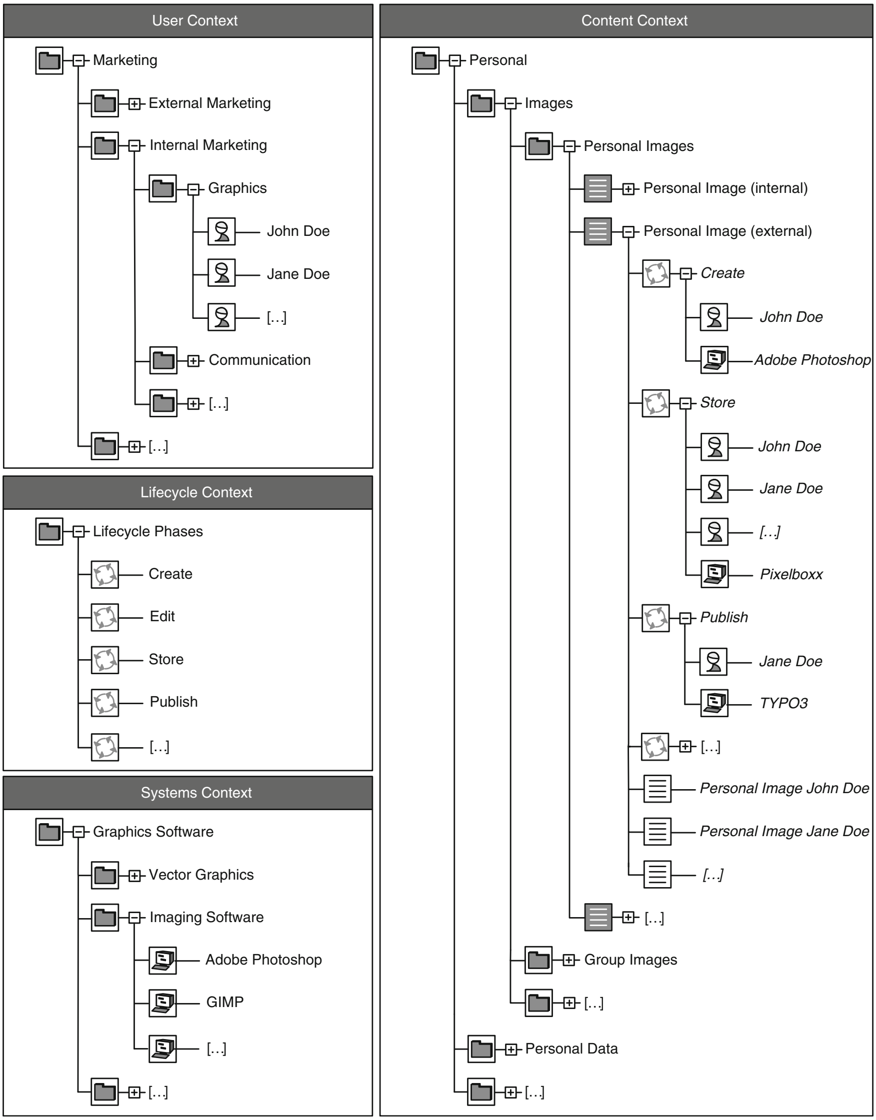

Fig. 3 Example of an *ECML* model (2)

Which model elements are relevant depends largely on the content modeling goals. As many goals can be pursued, *ECML* may require adaptation (e.g., in the form of additional element types and modeling rules). The next section provides an application example of *ECML* from practice.

Application Example and Evaluation

Project Overview

ECML was developed and evaluated during the course of a modeling project at Hoval. Hoval, a manufacturer of heating and ventilation systems, operates in more than fifty countries and employs about 1,200 people (http://www.hoval.com). Hoval operates five production sites in five countries: Liechtenstein (Vaduz), Austria (Marchtrenk), the UK (Lincoln), Slovakia (Istebné), and China (Beijing) (Hoval n.d.). Hoval has subsidiaries in thirteen countries, and the company's headquarter is in Vaduz, Liechtenstein. Hoval is a highly innovative company, and the generation and processing of knowledge and information has always been a central issue. Facing an increasing number of digital documents and an increasing amount of content to be dealt with in their business processes, Hoval started developing an ECM strategy at the beginning of 2009. The objectives of the project were to document semi-structured and unstructured business information, to increase information quality and process efficiency, to make document management more reproducible, and to meet information needs at the level of the individual employee. At a basic level, ECM strategy development at Hoval can be divided into seven phases: goal definition, delineation of tasks, as-is analysis, development of possible solutions, scheduling, implementation, and monitoring and evaluation. The results outlined in the following paragraphs are mainly related to the third phase, the as-is analysis, in which the researchers had the chance to participate as part of the project group.

Identification and analysis of existing documents, content, information needs, and systems was recognized early as a central issue of ECM strategy development at Hoval. While the project group had planned to document the results of this analysis in the form of tables, it became clear that this approach would be inefficient and would produce unstructured and confusing results. Therefore, conceptual modeling was considered as a way to increase the efficiency of the analysis and the clarity of the results. The primary goal of the modeling project was to document the reuse of content in documents at a company-wide level.

The results presented here were established and evaluated based on several sources of data. For example, we participated in four workshops (180–240 min each) and four discussion sessions with the project lead (60 min each) and conducted twelve semi-structured interviews (60–90 min each), each digitally recorded and fully transcribed to allow for detailed analysis, with decision-makers from several departments. The workshops and interviews primarily informed the identification and analysis of documents, content, systems, users, and potential for improvement. They were also used to identify the requirements for content modeling at Hoval, against which the developed modeling language was also evaluated. Our sources of data further included corporate documents (e.g., minutes and presentations from the workshops, existing document overviews).

In sum, twelve content models for the company's departments were developed and integrated into a consolidated model. The models were designed by means of *H2-Toolset*, a meta-modeling software for the specification of hierarchical models developed at the European Research Center for Information Systems (ERCIS). One of these models is presented in the following section, after which evaluation results are presented.

Application Example

The content model explained here was developed to document the reuse of content in Hoval's spare parts catalogue. The main data source for the design of this model was a web-based platform named *Internet Data Access*. As this catalogue is publicly available, no user rights and roles were modeled. Likewise, the modeling of systems and lifecycle phases were out of the scope of this project.

The screenshot displayed in Fig. 4 shows the Content Context and the Document Context in which the product-related content and documents are described and grouped in the »Products« folder. Various Content Types and Content Assets are distinguished in the Content Context, such as the »Front Pages« of »Operating Instructions«, which contain the respective »Product Names« (1) and »Images« (2). »Front Pages« represent composed Content Types, whereas »Product Name« and »Image« both represent self-contained Content Types. For example, Content Assets are the »Product Names« and »Images« for »Oil Condensing Boiler MultiJet® (8–25)« and for »Oil Condensing Boiler UltraOil® (35, 50)«. The described content is reused in the Document Context. As the screenshot shows, »Operating Instructions«, as a Document Type, are generally composed of various Content Types (e.g., »Safety Information«, »Customer Service«, »Functional Principle«, »Boiler System Control«, »Imprint«) (3). In turn, Content Assets (e.g., »Safety Information Oil/Gas Boiler«, »Hoval Customer Service«) are reused in specific Document Assets (e.g., the »Operating Instructions« for »Oil Condensing Boiler MultiJet® (8–25)«) (4). The screenshot also shows the possibility of describing different versions (e.g., »00«, »01«) and languages (e.g., »DE«, »EN«) (5) and of assigning attributes to model elements (e.g., »Number«, »Notes«, »Status«) (6).

In the course of the project, the functionality of *H2-Toolset* was extended by two plug-ins, the use of which is also illustrated in the example. *Document Launcher* is a plug-in that allows model elements to be connected to real-world documents (i.e., to open documents directly from within *H2-Toolset*), and *Reuse Visualizer* allows inquiries about the reuse of content (i.e., to determine in which documents certain content assets are included). In the example, *Reuse Visualizer* indicates which Document Assets need to be updated if, for instance, the product name »Oil Condensing Boiler MultiJet® (8–25)« is changed. (In the example, such a change would affect the following documents: »Op. Instr. 2-MultiJet (8–25) TT23S«, »Op. Instr. 2-MultiJet (8–25) M1.3«, and »Op. Instr. 2/3/4-MultiJet (8–25)«) (7). Using *Document Launcher* makes it possible to get direct access to

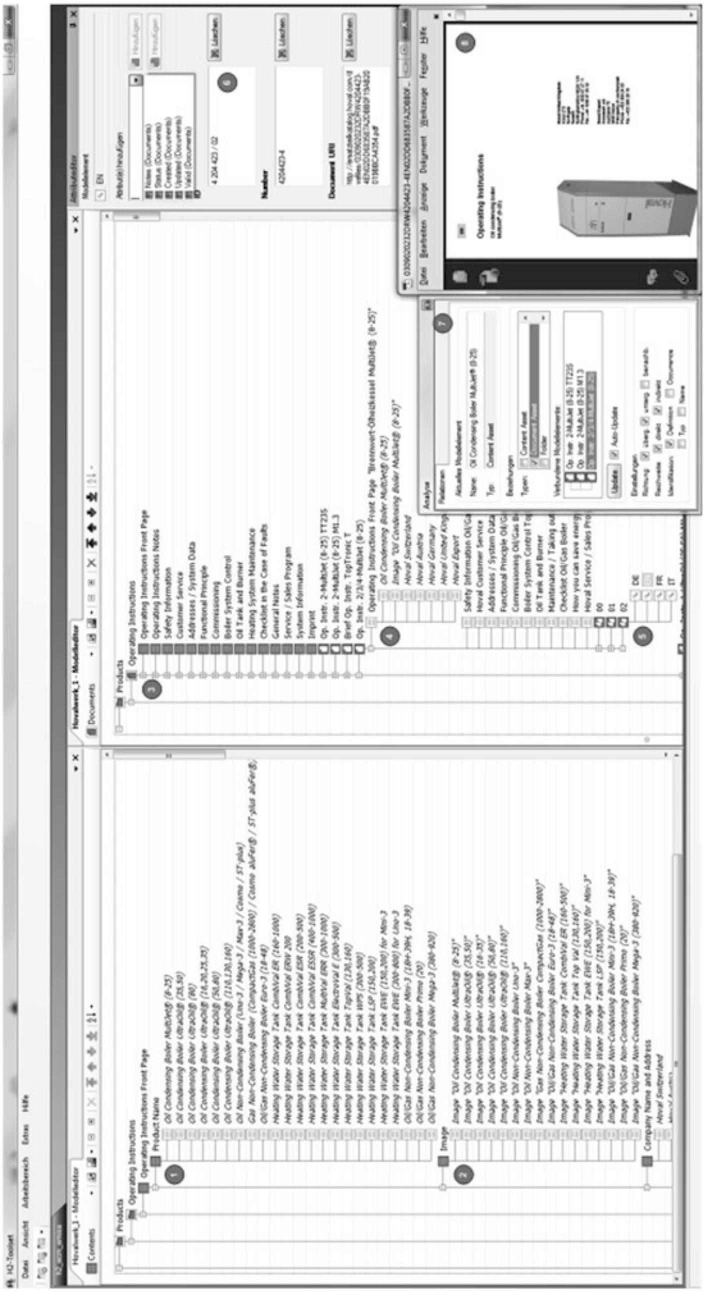

Fig. 4 Application example of *ECML*

documents in the model (e.g., the operating instructions for »Oil Condensing Boiler MultiJet® (8–25)«) simply by double-clicking on them (8). Therefore, a »Document URI« has to be assigned to the respective model element (in this case, a link to a Hoval Web page).

Evaluation Results

At Hoval, *ECML* was principally suited to model semi-structured and unstructured information. While the evaluation results confirm the practicability and effectiveness of the modeling language, they show also that further research is needed. This need is summarized using the requirements for content modeling from the modeling principles in Becker et al. (1995), which also formed the basis of the qualitative evaluation.

Correctness. The content model was assessed as being semantically correct; that is, its structure was determined to be compliant with the object system represented. As *ECML* was implemented by means of *H2-Toolset* (which automatically verifies compliance with the defined notation rules), the syntactical correctness of the model was ensured. While this approach to developing and maintaining the model was considered useful, a number of areas for further improvement were identified that would increase the semantic correctness of the model. For example, one recommendation was to implement a glossary to support model users during data entry and content search. Another recommendation referred to automated validation of the model with regard to its completeness, as model users may not always be able to specify all of the information required. Here, it was suggested that a testing mechanism be implemented that automatically informs the persons in charge of the model's maintenance about missing content or content that needs to be revised. Finally, options were discussed for more specific modeling of systems (e.g., release versions) and the business processes in which documents and content are created and used.

Relevance. The relevance of the content model was appropriate, and abstraction from users and systems in the content model was reasonable, although the level of detail was an issue. Content modeling took place on both a general level of content and document types and on the level of concrete instances, that is, content and document assets. However, since content and document instances are frequently modified, some model elements may be dated after just a short time. Therefore, the effort required to maintain the model could be substantial, which could lead to lower acceptance levels among its users. Stronger focus on the "type" level in content modeling was determined to be able to counteract this danger, but what level of detail ensures efficient maintenance remains to be examined.

Efficiency. The profitability of developing and maintaining the model was assessed as being good. Automatic reuse and updating of model elements, which is supported by *H2-Toolset* with *Object Definitions* and *Object Occurrences*, was considered particularly beneficial, and the usability of the *H2-Toolset* itself was assessed as adequate. However, the project partner identified some areas for improvement, such as the implementation of standardized dialogs that would support users in the process of entering data and maintaining content (e.g., the definition of mandatory content), and the consideration of inheritance principles (e.g., the reuse of metadata). While the two plug-ins created the possibility of using *H2-Toolset* for searching and maintaining documents and content (similar to using a very simple document management system), implementation of more efficient search mechanisms (e.g., based on a glossary) was also suggested.

Clarity. The project partner assessed the clarity of the model as being high, with particular emphasis on the hierarchical approach chosen as being adequate for the modeling of content. The symbols used fostered readability of the model, and the number of model element types was appropriate. A number of recommendations—such as implementation of where-used lists to document the reuse of content (which has been realized only prototypically by means of the *Reuse Visualizer* plug-in) and use of examples and notes in the model (which has been realized only rudimentarily by providing the ability to assign attributes)—were made to increase the model's clarity. The provision of general modeling guidelines was also an issue.

Comparability of models, the fifth modeling criterion considered, was not assessed in the course of the project. Nevertheless, it can be assumed that the development and maintenance of a glossary in particular would substantially increase model comparability (e.g., to avoid the use of synonyms).

Summary and Outlook

This chapter presented *ECML*, a modeling language for electronic documents and content that can support the implementation of ECM in organizations. Based on a review of the academic literature in the field, we explained the requirements for a modeling language for electronic documents and content. We then explained the *ECML* model elements based on a meta-model and summarized results from a modeling project at Hoval.

While the results of the modeling project confirm the practical relevance and effectiveness of the proposed language, they show also that additional research is needed. Future efforts should investigate the relevance of the *ECML* model elements, the level of detail of modeling, the possibility of assigning attributes to content, and the efficiency of maintaining *ECML* models. An *ECML* modeling method has already been developed and is to be evaluated in upcoming studies and projects. In addition, because we assessed only the applicability of some modeling approaches in this project, future research should investigate the suitability of further modeling languages and methods for describing the creation and use of content at an enterprise-wide level.

Acknowledgments This chapter is an extension and revision of a paper originally presented at the 14th Americas Conference on Information Systems (AMCIS 2008) in Toronto, Canada (vom Brocke et al. 2008). The authors thank the anonymous interviewees who assisted them in this research and the review team from AMCIS 2008 for their constructive comments on earlier versions of the paper.

References

Becker, J., Rosemann, M., & Schütte, R. (1995). Grundsätze ordnungsmäßiger Modellierung. *Wirtschaftsinformatik*, *37*(5), 435–445.

Boiko, B. (2002). *Content management bible*. New York: Wiley.

Clark, D. (2007). Content management and the separation of presentation and content. *Technical Communication Quarterly, 17*(1), 35–60.

Fill, H.-G., Gericke, A., Karagiannis, D., & Winter, R. (2007). Modellierung für Integrated Enterprise Balancing. *Wirtschaftsinformatik*, *49*(6), 419–429.

Frank, U. (1997). Möglichkeiten und Grenzen einer objektorientierten Modellierungslehre. In *Fachkongress Smalltalk und Java in Industrie und Ausbildung* (pp. 96–102). Erfurt.

Genero, M., Jiménez, L., & Piattini, M. (2000). Measuring the quality of entity relationship diagrams. In *Proceedings of the 19th International Conference on Conceptual Modeling* (pp. 513–526). Salt Lake City, UT.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems, 19*(2), 87–92.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *Management Information Systems Quarterly, 28*(1), 75–105.

Hoval. (n.d.). *Von inneren Werten*. Retrieved February 28, 2010, from http://www.waltherpartner.ch/bilder/Kunden/hoval_image.pdf

Kesh, S. (1995). Evaluating the quality of entity relationship models. *Information and Software Technology, 37*(12), 681–689.

Kung, C. H., & Solvberg, A. (1986). Activity modeling and behaviour modeling. In T. W. Olle, H. G. Sol, & A. A. Verrijn-Stuart (Eds.), *Information system design methodologies: Improving the practice* (pp. 145–171). Amsterdam: Elsevier.

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

Moody, D. L., & Shanks, G. G. (1994). What makes a good data model? Evaluating the quality of entity relationship models. In P. Loucopoulos (Ed.), *Entity-relationship approach—ER'94: Business modelling and re-engineering* (Lecture notes in computer science 881, pp. 94–111). Berlin/Heidelberg: Springer.

Munkvold, B. E., Päivärinta, T., Hodne, A. K., & Stangeland, E. (2006). Contemporary issues of enterprise content management: The case of statoil. *Scandinavian Journal of Information Systems, 18*(2), 69–100.

O'Callaghan, R., & Smits, M. (2005). A strategy development process for enterprise content management. In *Proceedings of the 13th European Conference on Information Systems* (pp. 1271–1282). Regensburg.

Päivärinta, T., & Munkvold, B. E. (2005). Enterprise content management: An integrated perspective on information management. In *Proceedings of the 38th Hawaii International Conference on System Sciences*. Big Island, HI.

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.

Rockley, A., Kostur, P., & Manning, S. (2003). *Managing enterprise content: A unified content strategy*. Indianapolis: New Riders.

Smith, H. A., & McKeen, J. D. (2003). Developments in practice VIII: Enterprise content management. *Communications of the Association for Information Systems, 11*(1), 647–659.

Tyrväinen, P., Päivärinta, T., Salminen, A., & Iivari, J. (2006). Guest editorial: Characterizing the evolving research on enterprise content management. *European Journal of Information Systems, 15*(6), 627–634.

vom Brocke, J., Becker, J., Simons, A., & Fleischer, S. (2008). Towards the specification of digital content: The enterprise content modeling language (ECML). In *Proceedings of the 14th Americas Conference on Information Systems (AMCIS 2008)* (paper 403). Toronto.

vom Brocke, J., Seidel, S., & Simons, A. (2010). Bridging the gap between enterprise content management and creativity: A research framework. In *Proceedings of the 43rd Hawaii International Conference on System Sciences*. Kauai, HI.

vom Brocke, J., Simons, A., & Cleven, A. (2011a). Towards a business process-oriented approach to enterprise content management: The ECM-Blueprinting framework. *Information Systems and e-Business Management*, *9*(4), 475–496.

vom Brocke, J., Simons, A., Herbst, A., Derungs, R., & Novotny, S. (2011b). The business drivers behind ECM initiatives: A process perspective. *Business Process Management Journal*, *17*(6), 965–985.

Wand, Y., & Weber, R. (2002). Research commentary: Information systems and conceptual modeling–A research agenda. *Information Systems Research, 13*(4), 363–376.

White, M. (2002). Content management: From vendor selection to successful rollout. *Online, 26*(6), 20–24.