A document management methodology based on similarity contents

Meziane, F and Rezgui, Y

http://dx.doi.org/10.1016/j.ins.2003.08.009

A Document Management Methodology Based on Similarity Contents ⋆

Farid Meziane *

School of Computing, Science and Engineering, Salford University, Salford M5 4WT, UK

Yacine Rezgui

Information Systems Institute, Salford University, Salford M5 4WT, UK

Abstract

The advent of the WWW and distributed information systems have made it possible to share documents between different users and organisations. However, this has created many problems related to the security, accessibility, right and most importantly the consistency of documents. It is important that the people involved in the documents management process have access to the most up-to-date version of documents, retrieve the correct documents and should be able to update the documents repository in such a way that his or her document are known to others. In this paper we propose a method for organising, storing and retrieving documents based on similarity contents. The method uses techniques based on information retrieval, document indexation and term extraction and indexing. This methodology is developed for the E-Cognos project which aims at developing tools for the management and sharing of documents in the construction domain.

Key words:
Document consistency, ontology, similarity content

1 Introduction

The main activity of most PC users is about creating, managing, deleting and retrieving electronic documents. Thanks to the existing file management systems, this organisation is performed using hierarchical structures whereby a document is stored and accessed at a specific location. For example, we would create a file "Lecture1.ppt" in the subdirectory "Lectures" which is itself a subdirectory of the "Object-Oriented Design" directory. In fact we are also associating some semantics to the created file. In this example, we have just created the first lecture of the "Object-Oriented Design" module. However, using strict hierarchical filing can make it hard for users to perform some operations that include [5]: *File documents:* documents can appear in only one place; *Manage documents:* locations in the hierarchy are used for organisational and management purposes; *Locate documents:* Document may be filed according to one criterion but retrieved according to another; *Share documents:* different structures for different people. The task becomes even more complex when dealing with various documents of one or more organisations particularly if the WWW is used as the place to exchange and organise these documents. Another major problem faced with shared documents is consistency whereby everybody interested in the document should be aware of any changes made to it.

Modern file management systems associate more information to user files. This information records for example the file's owener, its size, the date it is created and last accessed [5]. However, they have not properly addressed the previously mentioned issues. Some systems have attempted to solve some of these issues. The Presto system [6,5] aims at creating placeless documents and attempts to create a more natural and fluid forms of interaction with a document space. Their approach is based on document properties rather than document locations. They have defined documents properties as "the features of the documents that are meaningful to users such as categorisations, keywords and content-based features". However, the definition and association to documents of these features are left to the creator of the document. As stated by the authors, this can be a very subjective process. To remedy to this shortcoming the "documents properties are expressed relative to to user of the document, rather than the producer" [5]. Other properties identified as "active properties", including mainly functions such as summarisation and backup were also associated with the documents. DocMan [2] is a document

---

⋆ A Shorter version of this paper was presented at the 7th International Workshop on Application of Natural Language to Information Systems, Stockholm, Sweden, 2002

* Corresponding author
Email addresses: f.meziane@salford.ac.uk (Farid Meziane), y.rezgui@salford.ac.uk (Yacine Rezgui).

management system which supports cooperative preparation, exchange and distribution of documents. The system particularly stressed on the loss of work done simultaneously on a document and access restrictions. DocMan introduces the revision concept that prevents any loss of information caused by concurrent modifications by forbidding documents' revision to be overwritten. This is achieved by the creation of a new revision or version when the user modifies the document. Users are then informed about the different revision of the document. The Zelig System [4] was developed for managing multiple representation documents. It was claimed that different groups of users will favour different representations of documents. In the design of the Zelig system, a clear distinction between the conceptual level and presentation level was made. The conceptual level of the document is where the semantic of the document and its logical structure are represented. The presentation level is the way the document's semantic is conveyed to the user.

In this paper we present a methodology for managing and maintaining documents consistency using similarity content. This methodology is developed for the E-Cognos project which aims at developing tools for the management and sharing of documents in the construction domain. The approach is based on generic principles related to information retrieval and knowledge management. The aim of this project is to exploit these principles to develop an approach that will support consistency across large knowledge repositories maintained in a heterogeneous and distributed collaborative business environment. The methodology aims also at addressing most of the issues discussed in the previous paragraph. It mainly aims to:

- Identify a document through a set of document characteristics that are not defined by the document's producer but by a predefined set of properties and terms defined by the system's ontology. This will form the basis for both the classification (by the producer) and the retrieval of documents (by the user).
- Manage the updating process of documents by not only keeping track of all the changes but also by notifying users when new version of the documents are produced.
- deal with an heterogenous and large database of documents vital to the construction domain.

It is the aim of this methodology to automate all its steps and make the process transparent to the user. The approach is based on a solid theoretical foundation, and will be deployed in a real business environment. The remaining of the paper is organised as follows: in section 2 we present the motivation and the background behind the project. In section 3 we define the document logical representation and in section 4 we present the different types of document handled in this project. This is followed by section 5 where dimodels used for documents semantic characterization are presented. In section 6 we present

the generic model of the methodology used for poorly structured documents. Sections 7 and 8 presents variants of the methodology for documents with text formatting structure and highly structured documents.

2 Background and Motivation

The construction industrial processes are characterized nowadays by an intensive use of information technology. Decisions with the greater design and economic consequences are made in the early stages of a product's lifecycle. However, the integration of construction industry processes is becoming difficult due to new design considerations (new standards and regulations, energy consumption and material recycling requirements, etc.) and the continuous introduction of new techniques, materials and building elements, which result in the need of an increasing number of specialists in various domains. Numerous documents of diverse nature are involved in the construction domain. These documents are of two types: drawings and written documents. Drawings are the straightforward media to convey most of the information needed by construction companies and include a lot of information that can be hard to put into words. They are usually more formal and comprehensive than text information. Moreover, written documents are complementary to drawings, they are the traditional support of an engineering project description. Some of them such as building codes, examples of technical solutions, computation rules define the legal context of a project. Others like technical specifications documents or bill of quantities are generated by the engineering activities and often have a contractual importance.

The documents generated within the entire life cycle of a construction project, and especially during the design stage, need to be of quality in order to provide a reliable basis for contractors to perform their construction activities. Documents of quality are obtained by ensuring, during their production, a coherent and consistent structuring both on the logical and physical side. This structuring is relevant in the sense that the semantics of a document can be efficiently mastered and thus correctly described (absence of redundancies and inconsistencies).

Moreover, a document has not only to be self consistent but needs also to be consistent with the entire project documentary base as well as the construction standard and regulation base. Furthermore, many practitioners and researchers in the construction domain have recognised the limitations of current approaches to managing the knowledge relating to and arising from a project in a distributive collaborative environment. Among the reasons for these limitations are:

- Much knowledge, of necessity, resides in the minds of the individuals working within the domain.
- The intent behind decisions is often not recorded or documented. It requires complex processes to track and record the thousands of ad-hoc messages, phone calls, memos, and conversations that comprise much project-related information.
- Data is captured during a project and archived at the end of a project; this is necessary but not sufficient for knowledge systems. Knowledge is created by people actively reflecting on the events represented by the project data. The knowledge gained is often poorly organised and buried in details. Hence, it becomes difficult to compile and disseminate useful knowledge to other projects.
- People frequently move from one project to another, so it is difficult to track the people who were involved in a recorded decision and who understand the context of the decision making and its implementation.

Knowledge in the construction domain can be classified into the following three categories:

- *Domain knowledge:* this forms the overall information context. It includes administrative information such as zoning regulations and planning permission, standards, technical rules and product databases. This information is, in principle, available to all companies, and is partly stored in electronic databases.
- *Corporate knowledge:* this is company specific, and is the intellectual capital of the firm. It resides both formally in company records and informally through the skilled processes of the firm. It also comprises knowledge about the personal skills, and project experience of the employees and cross-organisational knowledge. The latter covers knowledge involved in business relationships with other partners, including clients, architects, engineering companies, and contractors.
- *Project knowledge:* this is the potential for usable knowledge and is the source of much of the knowledge identified above. It comprises both knowledge each company has about the project and the knowledge that is created by the interaction between firms. It is not held in a form that promotes reuse (e.g. solutions to technical problems, or in avoiding repeated mistakes), thus companies and partnerships are generally unable to capitalise on this potential for creating knowledge.

This overall context has often resulted in knowledge redundancy and inconsistencies, business process inefficiencies, and change control and regulatory compliance problems. Moreover, the introduction of new national regulations, or amendments made to existing ones, are often not handled effectively within organisations and projects.

3 Documents and Their Logical Representation

A document is a transitional and changing object defined within a precise stage of the project life cycle. Generally, a document is related to many elaborated documents of the project documentary database. A document has one or many authors. It is described by general attributes such as a Code, an Index, a Designation, a Date of creation and a list of its Authors. Ideally, a list of document versions also keeps track of any amendments made to the document during its lifecycle. An indexing system may be associated to the document. A document is submitted for approval according to a defined circuit of examiners representing diverse technical or legal entities. Each examiner issues a statement that enables the document to be approved, rejected or approved under reservation.

Also, documents have been traditionally represented using a set of key words. These key words or indices can either be manually defined by a user with a good knowledge of the semantics of the document, or extracted automatically from the text of the document using proven Information Retrieval (IR) techniques. A document has a logical and a physical structure, which are both used to convey in the best possible way its internal semantics. The physical structure of a document is described using a properly defined syntax supported by one or several software tools.

Each document should have ideally metadata attached to it. A possible solution for describing metadata is through RDF (Resource Description Framework - a development based on XML) that provides with a simple common model for describing metadata on the Web. It consists of a description of nodes and attached attribute/value pairs. Nodes represent any web resource, i.e. Uniform Resource Identifier (URI), which includes URL (Uniform Resource Locator). Attributes are properties of nodes and their values are text strings or other nodes.

4 A Document Type Taxonomy

Following the description of what a document is, as well as the leading metalanguage and language standards in this area, an attempt is made to classify documents based on their inherent nature and the structure they exhibit, taking into account the specificities of the construction sector. Three classes of documents have been identified, namely: poorly structured documents, documents with a clear physical structure, and highly structured documents.

4.1 Poorly Structured Documents

These are documents that are composed of text with no formal structure. These constitute the vast majority of the construction documentation. Documents are treated here simply as black-boxes. The set of operations associated with this category of documents include:

- Modifying the content of the document
- Deleting the document

4.2 Documents With a Text Formatting Structure

These are documents that are tagged using the HTML language, or at best the XML language but without reference to a Document Type Definition (DTD) [1]. A physical structure in the form of a hierarchical tree, or hypertext link of nodes can be easily generated from this representation. This structure offers a variety of possibilities in terms of text retrieval. These documents include direct references to other documents / document sections. The set of operations associated with this category of documents include:

- Insert a new document element such as heading, paragraph and section in a document.
- Deleting an existing heading in a document.
- Modify the contents of an existing heading

4.3 Highly Structured Documents

This categorizes documents that are instances of an XML-based meta-language. These documents have a semantic structure that can easily be used as a basis for text queries and retrieval. Ideally, we can envisage that all the documentation that is used and produced in the construction industry be an instance of a specific XML DTD over which users can exercise control over its internal semantics. These documents include naturally direct references to other documents/document sections. The set of operations associated with this category of documents include:

- Adding a new DTD element to the DTD language.

[1] A DTD is a formal description in XML Declaration Syntax of a particular type of document. It sets out what names are to be used for the different types of element, where they may occur, and how they all fit together [7]

- Instantiating a new DTD element within a document.
- Deleting the instance of a DTD element within a document.
- Modifying or extending the content of a DTD element instance.

5 Models for Documents' Semantics Characterization

Index terms are traditionally used to characterize and describe the semantics of a document. This approach attempts to summarize a whole document with a set of terms that are relevant in the context of the document. While this approach has given some satisfactory results in the area of Information Retrieval (IR), it still has some limitations as it proceeds by oversimplifying the summarization process by relying on a subset of relevant terms that occur in a document, and uses these as a mean to convey the semantics of the document. This section will describe the existing IR models that exist, a taxonomy of which is given in [1]. There are three main classical models of IR: Boolean, Vector and Probabilistic. In the Boolean model documents are represented as a set of index terms. This model is said to be set theoretic [9]. In the Vector model documents are represented as vectors in a t-dimensional space. The model is therefore said to be algebraic. In the probabilistic model, the modelling of documents is based on probability theory. The model is therefore said to be probabilistic. Alternative models that extend some of these classical models have been developed recently. The Fuzzy and the Extended Boolean Model have been proposed as alternatives to the set theoretic model. The Generalized Vector, the Latent Semantic Indexing, and the Neural Network models have been proposed as alternatives to the Algebraic Model. The Inference Network, and the Belief Network models have been proposed as an alternative to the Probabilistic Model. It is also worth mentioning that models that reference the structure, as opposed to the text, of a document do exist. Two models have emerged in this area: the Non-Overlapping Lists model and the Proximal Node model.

5.1 The Boolean Model

The Boolean model is based on the set theory and Boolean algebra. Query expressions are expressed as a combination of Boolean expressions, including Boolean operators which have a clear semantics. It was adopted and had great success in bibliographic and library information systems. The main criticism of the Boolean model [19]lies in its binary evaluation system. A document can be either relevant or not to a given query. There is no inherent ability to rank the document in relation to its relevance to a given query. In other words, there is no notion of partial match to the query conditions. It is commonly

acknowledged today that index term weighting provides more satisfactory results in retrieval performance. More information on the Boolean model can be found in [1,20,19].

5.2 The Vector Model

The Vector model addresses the limitations of the Boolean model by providing an approach that supports document partial matching to a given query. This is achieved by assigning non-binary weights to index terms in documents and queries. These term key word weights are then used in a second stage to sort documents by their level of relevance to the initial query. More details and further description of the Vector model, which is today considered as the most popular IR model, can be found in [1,15,16].

5.3 The Probabilistic Model

This was introduced initially by Robertson and Sparck Jones[13] as a mean to address the Information Retrieval problem within a probabilistic context. It proceeds by refining recursively a guessed initial set of documents matching a user query by involving the user feedback to evaluate the relevance of the retained set. For each iteration, the user retains the documents that best match the query. The system uses then this information to refine the description of the ideal response set. As highlighted in [1], the main advantage of the probabilistic model is that documents are ranked in decreasing order of their probability of being relevant. The disadvantages include:

(1) to guess the initial separation of documents into relevant and non relevant sets
(2) the method does not take into account the frequency in which an index term appears within a document

A thorough description of the Probabilistic model can be found in [12].

5.4 Alternative Set Theoretic Models

Alternative set theoretical models include the fuzzy set model and the extended set model.

5.4.1 Fuzzy Set Model

Several models that make use of the Fuzzy Set theory have been proposed. The model from Ogawa, Morita and Kobayashi [11] deserves a particular attention in that a thesaurus is being used in conjunction with the Fuzzy Set theory to expand the set of index terms in a query and extend the retrieved document set.

5.4.2 Extended Boolean Model

The principle behind the extended Boolean model is to overcome the binary limitations of the Boolean model by extending the latter and enhancing it with partial matching and term weighting from the vector model. This model has been introduced by Salton, Fox and Wu [14]. More thorough description can be found in [14,1].

5.5 Alternative Algebraic Models

Alternative Algebraic models include the Generalized Vector Space Model, the Latent Semantic Indexing Model, and the Neural Network Model.

5.5.1 Generalized Vector Space Model

The Generalized Vector Space model assumes that two index term vectors might be non-orthogonal which means that there is a possibility for two index terms to be correlated. This term correlation is used as a basis for improving retrieval performance [22].

5.5.2 Latent Semantic Indexing Model

The principle behind the latent semantic indexing model is that ideas in a text are more related to the concepts that are conveyed within it as opposed to index terms. By using this approach, a document may be retrieved only by the virtue that it shares concepts with another document that is relevant to a given query. As indicated in [8], the intent behind the latent semantic indexing model is to map each document and query vector into a lower dimensional space which is associated with concepts. This is achieved by mapping the index term vector into the lower dimensional space [1].

5.5.3 Neural Network Model

The Neural Network model is based on research carried out in the area of Neural Networks. The principle behind ranking documents that are retrieved against a given query is to match the query index terms against the Document index terms. Since Neural Networks have been extensively used for pattern matching purposes, they have been used naturally as an alternative model for information retrieval [1]. Detailed description of this model can be found in [21].

5.6 Alternative Probabilistic Models

The use of probability theory for quantifying document relevance has always been a field of research in Information Retrieval sciences. Two examples of IR models based on probability theory are the Inference Network model and the Belief Network. Both models are based on the Bayesian Belief Networks that provides a formalism combining distinct sources of evidence, including past queries and past feedback cycles. This combination is used to improve retrieval performance of documents [18].

5.6.1 Inference Network Model

The Inference Network model takes an epistemological as opposed to frequentist view of the information retrieval problem [17]. It proceeds, as described in [1] by associating random variables with the index terms, the documents, and the user queries. A random variable associated with a user document denotes the event of observing that document. This document observation asserts a belief upon the random variables associated with its index terms. Both index terms and documents are represented as nodes in the network. Edges are drawn from a node describing a document to its term nodes to indicate that the observation of the document yields improved belief on its term nodes. The random number associated with the user query models the fact that the information request specified in the query has been met. This random number is also represented by a node in the network. The belief in the query node is then expressed as a function of the beliefs of the nodes associated with the query terms. A more description of this model can be found in [17,18].

5.6.2 Belief Network Model

The belief network generalizes the inference network model. It was introduced by Berthie *et al.* [3]. It is also based on an epistemological interpretation of probabilities. It differs from the inference network model in that it adopts

a clearly defined sample space. It therefore provides a separation between the document and query portions of the network. This has the advantage of facilitating the modelling of additional evidential sources, including past queries and past relevance information.

5.7 Structured Models

These refer to models that combine information on text content with information on the physical structure of the document. A comprehensive survey of structured models can be found in [1].

Based on this survey on document characterization models and the nature of the documents to be used in our system, we have chosen the vector model.

6 System Description

The general framework of the methodology, as shown in Figure 1, is for poorly structured documents. The methodology is composed of 7 steps and these are described in the following subsections. Step 0 is the entry point to the system. It can be the submission of a new document or the re-submission of a modified document. Both instances will go through the same process. A logical document is used for searching and other document related operations. A Physical document is only retrieved on users requests.

6.1 Document Cleansing Module

This step aims at reducing the document to a textual description by eliminating non-discriminating words. The resulting document contains mainly nouns and association of nouns that carry most of the documents semantics. A cleansed document reduces drastically text complexity allowing better performance in document retrieval and processing. This involves the following tasks:

- Lexical analysis of the text in order to treat digits, hyphens, punctuation marks, and the case of letters. This reduces the initial document into a subset of words that are potential candidates for index terms.
- Elimination of stopwords to filter out words with very low discrimination values for retrieval purposes.

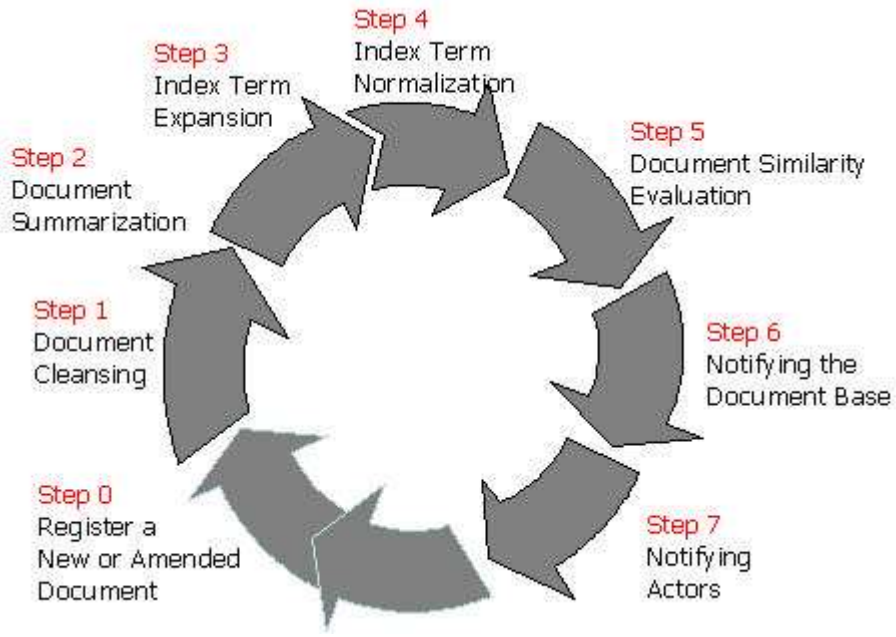

Fig. 1. System Overview for Poorly Structured Documents

- Stemming the remaining words with the objective of removing affixes (prefixes and suffixes) and preventing the retrieval of documents containing syntactic variations of query terms, e.g. use, using, used, usage, etc.
- Index terms selection whereby all index terms are reduced to their minimal number by only retaining nouns as nouns are expected to carry most of the semantics of the text [1].

6.2 Document Indexing

This step aims at providing a logical view of a document through summarization via a set of semantically relevant keywords. These are referred to, in this stage, as index terms. The purpose is to gradually move from a full text representation of the document to a higher-level representation. This module is composed of the following tasks:

6.2.1 Index Terms Extraction

In order to reduce the complexity of the text, as well as the resulting computational costs, the index terms to be retained are:

- All the nouns from the cleansed text. It is in fact argued that most, if not all, of the semantics of a text document is carried out by nouns as opposed to verbs, adjectives, and adverbs [1].

- Noun groups (non-elementary index terms) co-occurring with a null syntactic distance (number of words between the two nouns is null).

6.2.2 Extracting the Structure of the Document

This stage will only be possible if the document has been produced using a document formatting language, including RTF, SGML, HTML and XML. Each node of the resulting hierarchical structure will have an identifier that will be used to track nodes, their parents and children. A node might contain other elements including references within or outside the scope of the document, figures, tables, formulas, etc.

6.2.3 Establishing the Inverted File Structure of the Text

The purpose of an inverted file structure is to track the position of each index term occurrence in the text. The positions of index term occurrences can be tracked either on a word or character basis, or on a physical position basis (by pointing to node identifiers for example). The latter technique, referred to as Block Addressing, can only be used where a physical structure of a document is available. It presents the advantage of reducing space storage requirements. Documents with no clearly defined physical structure will make use of word addressing. Two activities are involved in this stage:

- Determining the raw frequency of each index term in the text; This is referred to as intra-clustering similarity in the Vector Model [1,15,16]. This aims at determining the number of times a term is mentioned in a document, as well as the location in the document of the term occurrence.
- Determining the number of documents in which each index term appears. This aims at counting the number of documents of the Document Knowledge Base (Project Knowledge Base and / or Corporate Knowledge base) in which the term appears.

6.2.4 Calculating the Index Term Weight for the Document

The purpose here is to quantify the degree of importance in terms of semantics the index term has over the document. The following formula from the Vector Model is used:

$$w_{i,j} = f_{i,j} \times idf_i \tag{1}$$

where

$w_{i,j}$ represents the quantified weight that a term k_i has over the document d_j.

$f_{i,j}$ represents the normalised frequency of a term k_i in a document d_j and is calculated using equation 2:

$$f_{i,j} = \frac{freq_{i,j}}{max_l freq_{l,j}} \tag{2}$$

Where

$freq_{i,j}$ represents the number of times the term k_i is mentioned in document d_j

$max_l freq_{l,j}$ computes the maximum over all terms which are mentioned in the text of document d_j

idf_i represents the inverse of the frequency of a term k_i among the documents in the entire knowledge base, and is expressed as shown by equation 3:

$$idf_i = \log \frac{N}{n_i} \tag{3}$$

Where

N is the total number of documents in the knowledge base and n_i is the number of documents in which the term k_i appears.

6.3 Index Terms Expansion and Normalization Using a Construction Thesaurus and Ontology

This step aims at normalizing the index terms obtained from the previous stage by using either direct ontology concept mapping wherever possible, or indirect ontology mapping by using a thesaurus as described in Figure 2. If no direct mapping exist between the initial index term and the list of ontology concepts then the thesaurus is used to provide synonyms for each term. The synonyms are used for indirect mapping. It is important to emphasise that the ontology is the structure that is used to convey semantics and maintain knowledge consistency across the project, corporate and domain layers. As such, the concepts of the ontology are the unique reference for the E-Cognos platform.

The E-Cognos platform provides a set of knowledge management services, including an ontology-related service. This implements a dedicated API (Application Programming Interface) that supports ontology creation and maintenance (including concept creation). The ontology service is available through a "Web Service Model" implementation, further details can be found in [10].

To illustrate the concepts mapping, let suppose that the following sentence is to be summarized: "The separation element between the kitchen and dinning area is made of fire resistant bricks." Now, the key index terms to be extracted from the sentence are: "Separation Element, Kitchen, Dinning Area, Fire Resistant Bricks". Let consider the first index term, "Separation Element". It is highly possible that a given Construction Ontology would not contain such concept. An indirect mapping would have to be established via a dedicated Construction Thesaurus. The latter would be used to find concepts that are semantically close to the one of "Separation Element". The thesaurus would return a set of concepts such as Wall, which have an occurrence in the Construction Ontology, and which will therefore be used to establish an indirect mapping between the "Separation Element" index term and the Ontology via the concept of Wall. Let take the second index term "Wall". Let suppose that the term exists as such in the Construction Ontology. We have then a direct mapping between the index term and the Ontology.

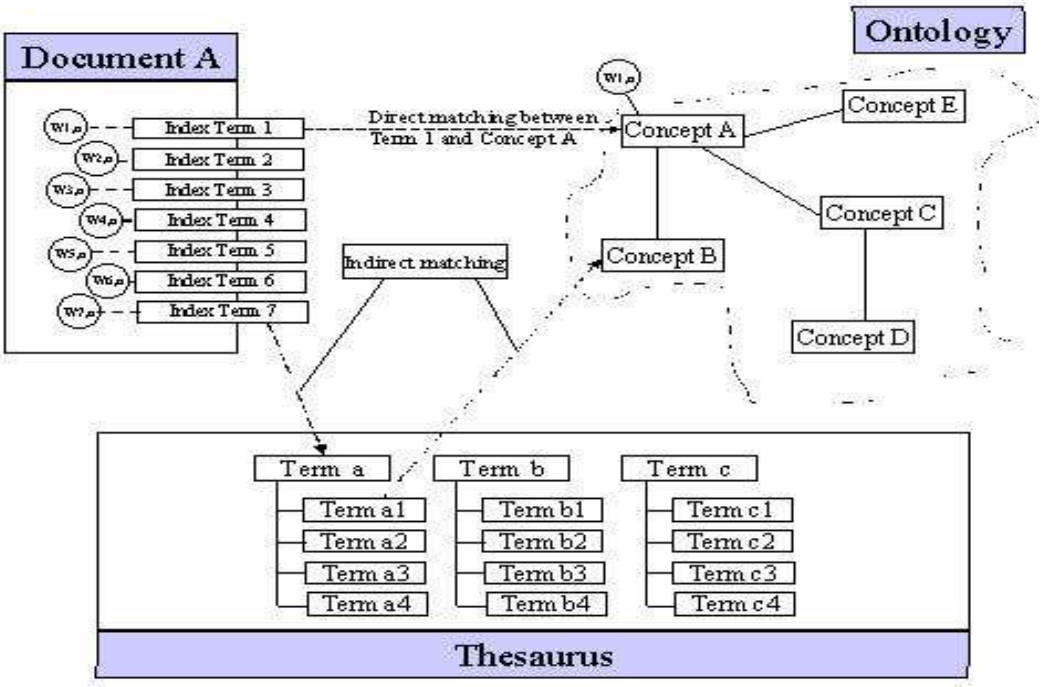

Fig. 2. Index Terms Mapping Against the Ontology

6.3.1 Ontology Concept Expansion Based on Concept to Concept Relationship

It is proposed in this methodology that the retained concepts be expanded based on their ontological direct relationships. We distinguish three main types of relationships:

- Generalisation/Specialisation Relationships(e.g. Wall can be specialized into Separation Wall, Structural Wall and Loadbearing Separation Wall)

- Composition/Aggregation Relationship (e.g. Door is an aggregation of of a Frame, a Handle :. etc)
- Concept association with varying semantics (e.g. a Beam supports a Slab and a Beis supported by a Column)

6.3.2 Ontology Concepts Weighting

The ontology concepts resulting from a direct document index term mapping, indirect index term mapping, or ontology concept expansion need re-weighting. The following approach is proposed:

(1) In case of a direct mapping the weight of the document index term is applied as such to the ontology concept.
(2) In case of indirect mapping or concept expansion, it is proposed that a correlation factor be applied to the initial document index term weighting. The correlation factor is obtained by the cosine of the angle between the Index Term Vector and the Ontology Concept Vector. This is based on a technique used in Query Expansion Based on Similarity Thesaurus [23].

Furthermore, to each index term is associated a Vector expressed as follows:

$$\overrightarrow{k}_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,N})$$

Where $w_{i,j}$ is a weight associated to the [index term, document] pair and is expressed as shown in equation 4:

$$w_{i,j} = \frac{(0.5 + 0.5\frac{f_{i,j}}{max_j(f_{i,j})})itf_j}{\sqrt{\sum_{l=1}^{N}(0.5 + 0.5\frac{f_{i,l}}{max_j(f_{i,l})})^2 itf_j^2}} \tag{4}$$

Where

t is the number of terms in the knowledge base, N is the number of documents in the knowledge base, $f_{i.j}$ is the frequency of occurrence of the term k_i in the document d_j, t_j is the number of index terms in the document d_j and itf_j the inverse term frequency for document d_j and expressed as follows:

$$itf_j = \log \frac{t}{t_j} \tag{5}$$

Therefore the correlation factor is expressed as shown on equation 6:

$$sim(k_i, k_j) = \frac{\overrightarrow{k_i} \bullet \overrightarrow{k_j}}{|\overrightarrow{k_i}| \times |\overrightarrow{k_j}|} = \frac{\sum_{q=1}^{t} w_{q,i} \times w_{q,j}}{\sqrt{\sum_{q=1}^{t} w_{q,i}^2} \times \sqrt{\sum_{q=1}^{t} w_{q,j}^2}} \tag{6}$$

The new weighting factor of the newly adopted ontology concept will be:

$$w_{j,d} = w_{i,d} \times sim(k_i, k_j)$$

6.4 Document Similarity Evaluation

The purpose of this step is to compare the similarity between a newly uploaded/processed document with the remaining documents (or knowledge items) stored in the various knowledge repositories.

6.4.1 Document Similarity Calculation Against all the Document Set

The purpose here is to provide a function that evaluates the similarity between two documents. We adopt the following approach:

Let t be the number of index terms in the system and k_i a generic index term. $k = \{k_1, k_2, \ldots, k_i\}$ is the set of all index terms. A weight $w_{i,j} > 0$ is associated with each index term k_i of a document d_j . If an index term does not appear in the document text then $w_{i,j} = 0$. Therefore, with a document is associated an index term vector:

$$\overrightarrow{d}_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$

Based on the document index term vector above, two documents d_i and d_j are represented as t-dimensional vectors. The approach adopted by the Vector Model to evaluate the similarity between a query and a document by measuring the correlation between their index term vectors is used. Furthermore, the similarity between two given documents will be measured by the correlation between their index term vectors. This correlation can be quantified by the cosine between these two vectors as shown in equation 7. $sim(d_i, d_j)$ varies between 0 and 1. An example of an illustration of the matrix is given in Table 1.

$$sim(d_i, d_j) = \cos(d_i, d_j) = \frac{\overrightarrow{d_i} \bullet \overrightarrow{d_j}}{|\overrightarrow{d_i}| \times |\overrightarrow{d_j}|} = \frac{\sum_{q=1}^{t} w_{q,i} \times w_{q,j}}{\sqrt{\sum_{q=1}^{t} w_{q,i}^2} \times \sqrt{\sum_{q=1}^{t} w_{q,j}^2}} \qquad (7)$$

6.4.2 Establishing Document Clusters Based on the Similarity Table

The purpose here is to propose clusters of documents based on their degree of similarity. These clusters can directly be generated from the Document Similarity Matrix proposed in the previous section.

Table 1
An example of a Document Similarity Matrix

	d1	d2	d3	d4	d5	d6
d1	1	sim(d2,d1)	sim(d3,d1)	sim(d4,d1)	sim(d5,d1)	sim(d6,d1)
d2	sim(d1,d2)	1	sim(d3,d2)	sim(d4,d2)	sim(d5,d2)	sim(d6,d2)
d3	sim(d1,d3)	sim(d2,d3)	1	sim(d4,d3)	sim(d5,d3)	sim(d6,d3)
d4	sim(d1,d4)	sim(d2,d4)	sim(d3,d4)	1	sim(d5,d4)	sim(d6,d4)
d5	sim(d1,d5)	sim(d2,d5)	sim(d3,d5)	sim(d4,d5)	1	sim(d6,d5)
d6	sim(d1,d6)	sim(d2,d6)	sim(d3,d6)	sim(d4,d6)	sim(d5,d6)	1

6.5 Notifying the Constituents of the Document Base

The purpose of this step is to notify relevant documents of the knowledge base, based on the nearest cluster(s), the potential risk of inconsistency that might exist as a result of a new event (upload of a new document, amendment to an existing document, etc.).

6.6 Notifying Relevant Actors

The purpose of this stage is for each potentially inconsistent document to notify actors who have subscribed an interest into the document (including authors) about this last event, and its potential degree of inconsistency. This notification will be materialised by the sending of an XML-based description of the meta-data of the newly created or amended document to all the concerned actors.

7 Maintaining Document Consistency Based on Document Explicit Relationships

This case applies to documents that have a clear physical structure and make use of hypertext navigational and cross-referencing links. Hypertext allows non-sequential browsing and editing of text. It can be represented as a network of nodes that are correlated by direct links in a graph structure. Each node is associated with a block of text that can represent a paragraph, chapter, section, or even a web page. Two related nodes are connected one to the other by a direct link, which correlates the text associated with these two nodes. This is explicitly described in the text by a special tag, or a highlighted portion

of the text. Figure 3 describes the application of the method for this type of documents. The proposed rules to apply are as follows:

Rule 1: if a node is amended then the node in question as well as the recursive parents should be flagged as potentially inconsistent. For instance, if paragraph P4 of document B in Figure 3 is amended then chapter b4 of document B(Containment Relationship) and chapter C3 of document A (Referencing Relationship) are potentially inconsistent, and should be flagged as such.

Rule 2: if a node is amended then the external nodes that are referencing it might be potentially inconsistent. These external nodes should be flagged as potentially inconsistent as they do reference an amended node. We consider that it is up to the author to look after the consistency of the internal references of the node being modified.

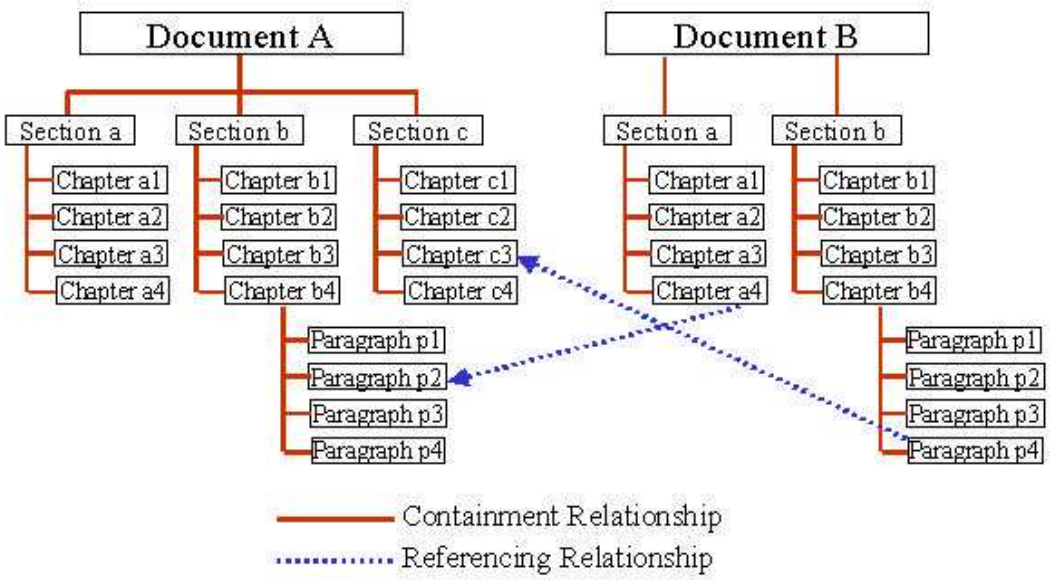

Fig. 3. Relationship Types in a Structured Hypertext Document

8 Maintaining Consistency of Highly Structured Documents

Highly structured documents are best represented by XML DTD compliant documents. XML documents allow human and machine-readable semantics mark-up. XML allows users to define new tags and impose data validation on them. This raises the problem of having unified and standardised definitions of tags used across documents. In that respect, it is highly recommendable to use a standardised DTD for authoring XML documents. This is already an area of intense activity (AECXML, bcXML, etc.). It is recommended in this approach that the elements of a given XML DTD be interpreted semantically

by indexing them, by the author of the DTD, to the concepts of the ontology. Concepts that highly describe the semantics of the contents of the instance of the DTD element are selected and retained by the DTD author(s) as this is a knowledge intensive activity. Therefore, each DTD element will be associated and indexed to a set of ontology concepts, as described in Figure 4.

In the same way, each ontology concept will be associated with a set of indexing DTD elements. Let us take an example:

DTD_ Element_ A1 has ontology indexes: *(Ont_ Con_ 1,Ont_ Con_ 3, Ont_ Con_ 4)*

DTD_ Element_ A2 has ontology indexes: *(Ont_ Con_ 2, Ont_ Con_ 4, Ont_ Con_ 6)*

DTD_ Element_ A3 has ontology indexes: *(Ont_ Con_ 4, Ont_ Con_ 6, Ont_ Con_ 8)*

If an instance of a *DTD_ Element* is amended, then the ontology concepts that index this element will be used to characterize this amendment. A simple but not very effective approach is to flag all documents that index the same ontology concepts of a document/or DTD Element instance that has been amended as potentially inconsistent. Using this approach, instances of *DTD_ Element_ A2* and *DTD_ Element_ A3* will be flagged as potentially inconsistent following an amendment to *DTD_ Element_ A1*.

A further step, which makes use of a more sophisticated approach, will attempt to retain only the instances from the flagged DTD elements that contain or reference the same ontology concept instance. This implies that the E-Cognos platform maintains instances of ontology concepts throughout the system.

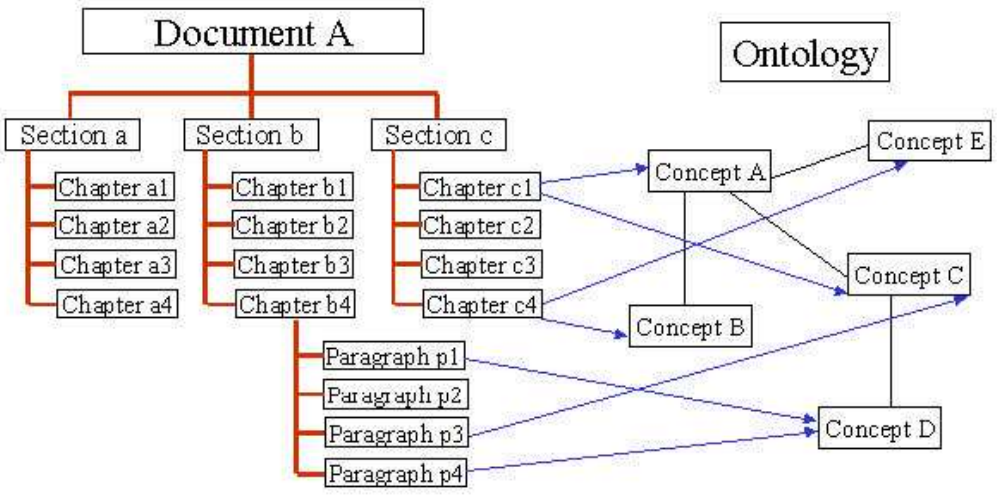

Fig. 4. XML Elements Indexing to the Construction Ontology

9 Conclusion

The work presented in this paper is the description of a methodology for maintaining document consistency across the knowledge repositories of the construction domain. The methodology uses generic principles related to information retrieval and knowledge management that can be incorporated into an approach that supports consistency across large knowledge repositories maintained in a heterogeneous and distributed collaborative business environment. E-Cognos aims at exploiting those principles to develop such an approach based on a solid theoretical foundation, and to deploy it in a real business environment in the context of the project partners. The methodology will be used in the construction domain. However, the model is generic and should be applicable to any other domain. Few changes might be necessary to take into account the nature of the document of the new domain and the use of another ontology which structure may influence some processes of the proposed model. A web-based implementation will be used for the E-Cognos project and this methodology will be implemented using Java and related technologies. It is also worth mentioning that the proposed methods assume that all documents have been authored using a common natural language. Moreover, the multilingual aspect of documents has not been addressed at the moment but will be addressed at a later stage.

Acknowledgements

The authors as well as the E-Cognos Consortium would like to acknowledge the financial support of the European Commission under the IST programme.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley, 1999.

[2] A. Bäcker and U. Busbach. DocMan: A document management system for cooperation support. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS-29)*, pages 82–91, 1996.

[3] A. Berthie, B. Ribeiro-Neto, and R. Muntz. A belief network model for ir. In *Proceedings of the 19th Annual International Conference ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–260, 1996.

[4] A. Celentano, S. Pozzi, and D. Toppeta. A multiple presentation document management system. In *Proceedings of the 10th Annual Conference on Systems Documentation*, pages 63–71, 1992.

[5] P. Dourish, W. K. Edwards, A. LaMarca, J. Lamping, K. Peterson, M. Salisbury D. B. Terry, and J. Thornton. Extending document management systems with user-specific active properties. *ACM Transaction on Information Systems*, 18(2):140–170, 2000.

[6] P. Dourish, W.K. Edwards, A. LaMarca, and M. Salisbury. Presto: An experimental architecture for fluid interactive document space. *ACM Transactions on Computer-Human Interaction*, 6(2):133–161, 1999.

[7] P. Flynn. The XML FAQ, version 3.01. WWW, `http://www.ucc.ie:8080/cocoon/xmlfaq#doctype`, 2003.

[8] G.W. Furnas, S. Deerwester, S.T. Dumais, T.K. Landauer, R.A. Harshman, L.A. Streeter, and K.E. Lochbaum. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the 11th Annual International Conference on Research and Development in Information Retrieval(ACM SIGIR)*, pages 465–480, 1988.

[9] V. Gudivada, V. Raghavan, W. Grosky, and R. Kasanagottu. Information retrieval on the world wide web. *IEEE Internet Computing*, pages 58–68, October–November 1997.

[10] C. Lima, B. Fies, A. Zarli, M. Bourdeau, M. Wetherill, and Y. Rezgui. Towards an IFC-based ontology for the building and construction industry: the e-cognos approach. In *proceedings of the eSM@RT European Conference on Information and Communication Technology Advances and Innovation in the Knowledge Society*, pages 254–264.

[11] Y. Ogawa, T. Morita, and K. Kobayashi. A fuzzy document retrieval system using the key word connection matrix and a learning method. *Fuzzy Sets and Systems*, (39):163–179, 1991.

[12] C.J. Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[13] S. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society of Information Sciences*, 27(3):129–145, 1976.

[14] G. Salton, E. Fox, and H. Wu. Extended boolean information retrieval. *CACM*, 26(11):1022–1036, 1983.

[15] G. Salton and M. Lesk. Computer evaluation of indexing and text processing. *Journal of the ACM*, 15(1):8–36, 1968.

[16] G. Salton and C. Yang. On the specification of term values in automatic indexing. *Journal of Documentation*, (29):351–372, 1973.

[17] H. Turtle and W.B. Croft. Inference networks for document retrieval. In *Proceedings of the 13th Annual International Conference ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1–24, 1990.

[18] H. Turtle and W.B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222, 1991.

[19] J. Verhoeff, W. Goffmann, and J. Belzer. Inefficiency of the use of boolean functions for information retrieval systems. *CACM*, 4(12):557–558, 1961.

[20] S. Wartick. *Boolean Operations*, pages 264–292. 1992.

[21] R. Wilkinson and P. Hingston. Using the cosine measure in a neural network for document retrieval. In *Proceedings of the 14th Annual International Conference ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 202–210, 1991.

[22] S.K.M. Wong, W. Ziarko, and P.C.N. Wong. Generalized vector space model in information retrieval. In *Proceedings of the 8th Annual International Conference ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25, 1985.

[23] Q. Yonggang and H. Frei. Concept based query expansion. In *Proceedings of the 16th Annual International Conference ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 160–169, 1993.