A Document Reuse Tool for Communities of Practice

Aida Boukottaya[1], Bernadette Charlier[1], Christine Vanoirbeek[2]

[1] Didactique Universitaire, University of Fribourg, Switzerland
[2] MEDIA research group, EPFL, Switzerland

{aida.boukottaya@epfl.ch, bernadette.charlier@unifr.ch, Christine.vanoirbeek@epfl.ch}

Abstract. With the rise of the Internet, virtual communities of practice are gaining importance as a mean of sharing and exchanging information. In such environments, information reuse is of major concern. In this paper, we outline the importance of enriching documents with structural and semantic information in order to facilitate their reuse. We propose a framework for document reuse based on an explicit representation of the logical structure as well as links to domain ontologies. Such explicit representation facilitates the understanding of the original documents and helps considerably in automating the reuse process. Document reuse automation is based on matching techniques that consider several criteria including semantic and logical similarities.

Keywords: Communities of practice**,** Document reuse, Self-describing documents, logical structure, semantics, Schema Matching.

1 Introduction

Communities of Practice (CoPs) are becoming more important as a mean of sharing information within and between organizations. A Community of Practice emerges from a common desire to work together; it can be defined as a network that identifies issues, shares approaches, methodologies, documents, experiences, and makes the results available to others [21]. With the rise of the Internet, virtual CoPs are gaining importance as a new model for virtual collaboration and learning. In virtual CoPs, the common space is provided by a suite of collaborative and communicative environments, ranging from simple mailers, forum, discussion lists, and audiovisual conferences to more advanced collaborative work environments that enable information and knowledge exchange and sharing.

In this context, the process of capturing and sharing a community's collective expertise is of major concern. In [6], author describes such process as a cyclic one composed by four basic steps: *find/create*, *organize*, *share*, and *use/reuse*. The "find/create" step concerns the creation of knowledge/information gained through research and/or industry experiences, publications, etc. The goal of the two next steps in the cycle, "organize" and "share", is to first filter and organise expertise (e.g.,

E. Tomadaki and P. Scott (Eds.): Innovative Approaches for Learning and Knowledge Sharing,
EC-TEL 2006 Workshops Proceedings, ISSN 1613-0073, p. 235-244, 2006.

creating different categories of knowledge related to specific purposes, linking such knowledge with available resources). Second, the expertise is shared for wide availability making use of the Internet and other techniques of information sharing such as conferences and collaborative work environments. The final phase of the cycle, "use/reuse," enables shared expertise to be used and reused in order to minimize information overload and maximize content usability which decreases considerably time, effort and cost. In this phase expertise is applied and reapplied to solve real-world problems. The results are then captured as part of learned lessons and new expertise is created which enables the cycle to begin again.

In this paper, we essentially focus on document reuse within CoPs. As in [15], we identified at least two kinds of document reuse: (1) by *replication*: from a single document, several presentations are produced; and (2) by *extraction*: portions of a document are taken from one document and moved to another (generally performed by means of the now popular "Cut&Paste" command).

Since documents reflect in general authors' vision and "understanding" of the Universe, document reuse process requires access to the intentions and interpretations underlying the original document. The capability of reuse suggests then the understanding of authors' representation of the Universe in term of concepts and semantic relationships among them. Such representations only exist "in the mind" of authors and usually are not apparent in the document itself. Moreover, when reuse requires crossing system and application boundaries, several problems arise due to the heterogeneities of such systems. One response to these problems is to structure documents by using Markup Languages such as XML [22]. The advent of structured documents on one hand leveraged a promising consensus on the encoding syntax for machine processable information and such resolves several issues, such as parsing and character encoding recognition. On the other hand, mark-up identifies meaningful parts of a document, and thus makes authors' intentions more explicit.

In this paper, we essentially address the second kind of reuse (extraction). We consider documents as an effective mean for storing explicit knowledge, and study the additional benefits of using structure and explicit representation of metadata and semantic information. This work is carried out in the framework of PALETTE project[1] aiming to provide communities of practice with a set of services concerning data production, exchange and reuse; reification of explicit and tacit knowledge about practices and advanced collaboration.

The outline of the rest of the paper is the following: Section 2 describes a motivating scenario based on the observation of LEARN-NETT community. Section 3 gives an overview of the benefits of structuring documents. Section 4 proposes a multi-layered

[1] The work presented in this paper is carried out in the framework of a collaboration between the EPFL Center for Global Computing and the University of Fribourg and funded within the FP 6 IP project PALETTE (FP6-028038): http://palette.ercim.org/

model for documents that is built using annotation facilities. Section 5 gives the conceptual framework for the proposed reuse tool.

2 Motivating Scenario

To elucidate the need for document reuse, we present a simple use case using observations we made to LEARN-NETT[2] community. LEARN-NETT is a virtual campus aiming at conceiving and trying methodologies for training teachers (also called students) based on a learning-oriented approach [8]. Students produce either group documents (reports, etc) or individual documents (dissertations, individual reflections). Tutors in LEARN-NETT community have a central role in the organization and the regulation of the students' groups. More exactly, they help students to express their needs, animate the work of the group, provide resources, regulate exchanges, and give quick feedback. For this, tutors rely on a pedagogical guide and a set of references and resources. Tutors are supported in their activities by a project coordinator. The coordinator participates in the elaboration of pedagogical guides and tools for tutors. He also produces a weekly report summarizing the project progress.

Produced documents reflect actors' experience and expertise. In this context, reusing such expertise is of major concern. For instance, a student group aiming to solve a real-world problem could reuse the expertise of previous groups. Instead of producing reflections, reports from scratch, we could maintain a material pool consisting of definitions, theorems and their proofs, exercises, book chapters, dissertations, reflections and examinations. When a student is producing a new document, he (or she) could reuse this existing material which reduces considerably time and effort. Students' researches (e.g., dissertations and scientific papers) could also be reused for designing tools and pedagogical guides for tutors. The major problem to address while reusing such documents is their heterogeneities. Heterogeneity arises in general from the fact that each author creates its own documents according to specific requirements and goals.

Based on these observations, we essentially distinguish two categories of heterogeneities: *organisational (structural) heterogeneity* and *semantic heterogeneity*. Organisational heterogeneities [12], [13] and semantic heterogeneities [20], [16], [10] have been well documented in the literature with a consensus of what each encompasses. In most cases, the distinction between the two can be characterized by differences in organisation (*how are the data in the document is organised?*) and interpretation (*what do the data mean?*). This distinction however is not always clear, since the organization of data often conveys semantic information. Semantic heterogeneity refers to domain level incompatibility. Examples include the attribution of different names for semantically equivalent concepts and the attribution of the same name for semantically different concepts. Organisational heterogeneity arises

[2] http://tecfa.unige.ch/proj/learnett/

when semantically similar entities are modelled using different descriptions. As an example, we can consider the organization of pedagogical units (using an ascending or a descending approach). An ascending approach presents to students concrete cases and tends to generalize them in order to extract a theory. This theorization supposes a good understanding of the real facts. In such a strategy, bricks representing examples of a concept are presented before bricks describing the theory of the same concept. Contrary to the ascending strategy, the descending one consists in presenting at first the theory, and then when this one is supposed to be understood, examples are presented in order to assimilate better the theory. The goals of the two strategies are the same, but the organisation of pedagogical units differs. Reusing documents suggests the capability to resolve such heterogeneities.

3 Structured document reuse

3.1 Why structuring documents?

Structured document refers to a document conforming to a pre-defined grammar or schema that describes the permissible document components and their logical organization [1]. XML is the mark-up language for presenting information as structured documents. The document structure (described in a DTD or more recently using an XML Schema [23]) can be utilized to facilitate several issues such as document authoring, document publishing, document querying and browsing, etc. Based on structure, it is easy to achieve replication. Different layout formats such as HTML (for Web sites), PDF (Printed documentation), WML (for wireless devices) could be generated automatically. However, dealing with structured documents has also some drawbacks. Reusing structured documents (by extraction) raises a number of fundamental problems to transform or to adapt their intrinsic structure. Structure transformation process is known to be extremely laborious and error-prone. It is typically attained by writing manually translators (often encoded on a case-by-case basis using specific transformation languages such as XSLT [24]). This is generally achieved trough three main steps: understanding the source and target schemas, discovering schemas' mapping by means of inter-schema correspondences, and translating mapping result into an appropriate sequence of operations in a given transformation language [14].

3.2 Schema matching

A serious obstacle for translating directly between two structured documents is that a mapping between both schemas needs to be carefully specified by a human expert. Manual mapping is known to be a time consuming and error-prone process. One response to this problem is *schema matching*. Schema matching is the task of semi-automatically finding correspondences between two heterogeneous schemas. Several applications relying on schema matching have arisen and have been widely studied by

the database, AI communities and more recently document engineering community [18], [7], [17].

Mapping two schemas is a very challenging problem. Solutions to this problem have produced two types of matchers: *structural matchers* and *semantic matchers*. Structural matchers typically map two schemas according to their syntactic clues. Examples of such clues include element names, types, and common logical structure. See our previous work [4] for more examples of syntactic matchers. However, such clues are often unreliable and incomplete. For example the same labels may be used for schema elements having totally different meanings. In such conditions, the main challenge is not to only determine existing relations between schema elements, but also making sure that the matching process does not discover incorrect mappings. Moreover using only structural matching, semantic mismatches are largely undressed. In contrast, semantic matchers rely on explicit knowledge generally stored within a domain ontology[3] in order to improve mapping accuracy. Although these approaches use semantics, its use is limited to taxonomic knowledge to determine, for example, that the term used in one schema generalizes or specializes a term in the other schema. As a result, structural mismatches are not addressed although the structure of a document often conveys semantic information and traduces the designer point of view. We believe that both the logical structure of the document and additional semantic information relating to a domain of interest, are important for both identifying reusable document fragments and adapt them according to user needs.

4 Re-thinking document structure

In open and evolving environments, such as the ones used by communities of practice, the number of shared and exchanged documents is increasingly growing. As noticed in the motivating scenario (section 2), exchanged documents are of various formats. Examples include totally unstructured (documents containing raw text expressed in natural language), semi-structured[4], text documents (containing structural information such as chapter, section, sub-sections, etc), and highly structured documents based on predefined schema. In this context, one of the huge challenges we face that is the automation of such documents' content reuse. This difficulty is due to the lack of explicit structure and knowledge.

To address this problem, we propose a "*self-explaining*" document model. A document is considered to be self-explaining if it contains an explicit representation of its logical structure and semantics. As in [9], we conceive this model as a multi-layered model. The layout layer (or physical layer) reflects document format and publishing characteristics. It answers the question: "*how has to appear the document on a given publishing support?*" It is either embedded within the document in terms of typographic characteristics (Courier, Times, red, etc), or expressed outside the

[3] An ontology is a shared conceptualization of knowledge in a particular domain.

[4] Semi-structured documents are documents where the structure is often irregular, partial, unknown, or implicit.

document by means of style sheets (e.g., CSS Style sheets for Web documents). The logical layer represents an organization in term of structure (Chapter, paragraph, title, etc). It is expressed generally in terms of logical elements and can be either implicit in the document or explicitly expressed using schema languages. The meta-information layer includes two types of information: (1) meta-data describing the intrinsic properties of a document (e.g., title, authors, etc) and are generally expressed in languages such as RDF [19]; (2) domain vocabulary and taxonomies (expressed using ontologies and/or thesauri) relating document content to a specific domain of interest.

The first objective of our work is to make structured, semi-structured and un-structured documents self-explaining. For structured documents, the problem is quite easy since the layout structure and the logical structure are already separated. The problem is more complicated for semi-structured and unstructured documents. One solution to this problem is to offer annotation facilities. Annotation refers to new information such as comments, semantics and new structures placed over existing documents. The goal is to progressively *facilitate* and *motivate* authoring of structurally and semantically tagged document content.

4.1 Manual annotation Vs automated structure/semantics extraction

With the advent of structured documents, several researches and industrial efforts have been dedicated to the analysis of raw or semi-structured documents in order to structure or re-structure them. In [11], authors proposed the MarkItUp system designed to recognize the structure of untagged electronic documents; their approach is based on learning by example to gradually build recognition grammars. Authors in [2] used a constraint propagation method to extract logical structure of library references. Work described in [3] proposed an approach based on the use of a transformation language to interactively restructure HTML documents.

Research in information extraction and automatic metadata extraction generally rely on the existing of many documents (sharing the same format) with similar structure and semantics, which is very difficult and inapplicable to communities of practice where a variety of documents are produced with very differing format, structure and semantics. In this context, we advocate the use of manual annotations. The main difficulty is enabling and motivating non-technical users to structure and semantically enrich their documents.

4.2 Requirements for annotation tool

One of the fundamental problems we face when designing an annotation tool for a communities of practice, is to incite their members to take the effort to produce structured documents and then semantically link document elements to available

domain ontologies[5]. To answer this problem, we fix a set of requirements for the annotation tool we aim to develop:
(1) Ease of use: the proposed annotation tool should be easy to use; this could be achieved by providing authors with a convenient graphical interface that abstracts languages syntax (XML Schema, RDF, Ontology description languages). Moreover, authors should be provided by a set of predefined schemas (deduced from the analysis of CoPs activities) as well as domain ontologies in order to assist him/her to annotate document content easily. However, authors should also have the freedom to modify and/or add specific elements to predefined schemas in order to answer their own need.

(2) Annotation result representation and evolution: Annotation result should be presented in a graphical manner in order to help the user in the validation of the produced result. Moreover, in a CoP evolving environment, documents can easily evolve; the annotation result should be then adapted without redoing the whole annotation process. One solution is to structure annotations. Structuring annotation result greatly increases its reusability, especially when documents evolve.

(3) Motivating annotations: authors will be motivated to annotate their document content only if they experience the added value taken from this exercise. The idea is to provide CoP's members with a set of services that consume structured and semantically enriched documents and produce useful results. Document reuse tool is one of these services. In the context of PALETTE project, several services based on structured documents will be provided (information discovery based on annotations, publishing services, etc).

5 Document reuse tool: Conceptual Framework

The proposed information reuse tool consists of a set of Web services. Web services are defined as loosely coupled, reusable software components that refer to programmatic interfaces used in the Word Wide Web for application-to-application communication. A main characteristic of Web services is that they are self-describing, which means that they contain all necessary information advertising their functionalities. Web services are particularly interesting for virtual communities, as they allow non-technical community members to combine them in new value-adding services. Based on our previous work on structured document reuse [4] [5], we propose a conceptual framework (Figure 1) that encompasses the whole document reuse process. The framework consists of four basic set of services:

Document restructuring services: include (1) annotation service which has to manage links between original documents, predefined schemas and ontologies; (2) the structuring of annotation result. Document restructuring services use ontologies provided by domain knowledge management services. They also interact with evolution services to manage annotations' changes; and with validation services to

[5] A working team within the PALETTE project is focusing on developing evolving ontologies for CoPs

validate annotation results. To do all these tasks, document restructuring services rely on set of user interfaces. These services are currently under development in the context of PALETTE project. A set of tests and an evaluation process are planned with the help of several CoPs.

Matching Services: In order to reuse structured documents, we need to establish a set of similarities between the reused fragments and the document where fragments will be reused. To do this, we adopt a multi-criteria matching process. Each criterion is represented by a Web service. These services are extensible. As new criterion become available to resolve the schema matching problem, a new Web service is created. Examples of developed services include: (1) *Semantic similarity service*: measures the similarities between entities based on the meaning inferred from their names and their links to domain ontologies; (2) *Constraint similarity service*: relates schemas elements based on their respective constraints (specified in the logical layer). Such constraints include the use of Datatypes and integrity constraints; (3) *Structural similarity service*: relates schemas entities based on the similarity of the structural context in which they appear (defined by their ancestors and descendents in the logical model). The idea behind our proposed solution is to represent each element's context as a path and to then rely on a path resemblance measure to compare such contexts. To achieve this, we relax the strong matching notion frequently used in solving query answering problem. To compute path resemblance measure, we further use algorithms from dynamic programming. These services are finalized and details about related theory and algorithms can be found in [4], [5].

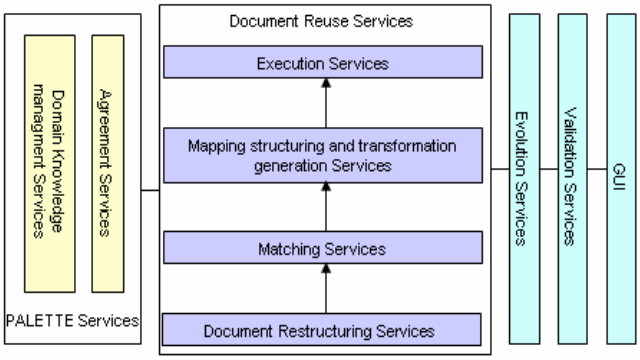

Fig. 2. Conceptual Framework for Document Reuse tool

Mapping structuring and transformation generation services: The main goal of these services is to combine all the above similarity measures and produce a mapping result that clearly defines source and target mapped entities, required transformation operations, and conditions under which the mapping can be executed. These services rely on validation services using graphical representation of the mapping result enabling the user to both valid mapping result and to add further constraints in a transparent manner.

Execution Services: These services generate automatically the appropriate transformation scripts based on the above mapping structure.

Additional services run along the entire reuse process, interacting with the former four modules. Domain knowledge management services are services that define lexical and domain-specific ontologies for CoPs. Agreement services are responsible for establishing a consensus on predefined schemas and/or ontologies. These two services are currently under development by other partners in the PALETTE project. Evolution services are responsible in keeping both annotation results and mappings in synchrony with documents changes.

6 Summary

Communities of practice are social networks of relationships that provide information, knowledge, and a space where people interact for mutual benefit. This paper studies document content reuse problem within CoPs. Faced with the diversity of documents formats, content and goals, a critical step in document reuse is to make such documents self-explaining. The main idea is that by enriching original documents with an explicit logical structure as well as linking content to available ontologies, we can assist authors in the reuse process. This is done by proposing a set of services able to determine similarities between original documents and reused fragments. We proposed a conceptual framework describing such services and their interactions. Currently, we are instantiating the framework in the context of several Cops participating to PALETTE project. In the future, the main task will be dedicated to the evaluation and enhancement of the proposed framework based on CoPs feedback.

References

1. Abiteboul, S., Buneman, P., Suciu, D.: Data on the Web: From Relations to Semistructured Data and XML. Morgan Kaufmann Publishers, 2000.
2. Belaïd, A., Chenevoy, Y.: Constraint Propagation vs Syntactical Analysis for the Logical Structure Recognition of Library References. Lecture Notes in Computer Science 1339, BSDIA'97, Springer, pages 153-164, Curitiba, Brazil, November 2-5, 1997.
3. Bonhomme, S., Roisin, C.: Interactively Restructuring HTML Documents. Computer Network and ISDN Systems, vol. 28, num. 7-11, pages 1075-1084, May 1996.
4. Boukottaya, A.: Schema matching for structured document transformations, PHD thesis, October 2004.
5. Boukottaya, A., Vanoirbeek, C.: Schema matching for transforming structured documents. ACM Symposium on Document Engineering 2005: 101-110
6. Burk, M., Knowledge Management: Everyone Benefits by Sharing Information, http://www.tfhrc.gov/pubrds/novdec99/km.htm, 1999
7. Cali, A., Calvanese, D., Giacomo, G., Lenzerini, M.: On the expressive power of data integration systems. In Proceedings of 21st International Conference on Conceptual Modeling (ER 2002), pages 338-350, Tampere, Finland, 2002.
8. Charlier, B., DAELE, A., Docq, F., Hecquet, G., Lebrun, M., Denis, B., Peeters, R., De Lievre, B., Deschryver, N., Lusalusa, S., Peraya, D. : Learn-Nett: une expérience

d'apprentissage collaboratif à distance, Actes de la 1e biennale des chercheurs en sciences de l'éducation, Bruxelles, mai 2000.
9. Christophides, V.: Electronic Document Management Systems". Available at http://www.ics.forth.gr/~christop/ , 1998.
10. Garcia-Solaco, M., Saltor, F., Castellanos, M.: Semantic heterogeneity in multidatabase systems. In Bukhres, O.A and Elmagarmid, A.K., editors, Object Oriented Multidatabase Systems: A Solution for Advanced Applications, chapter 5, pages 129-202. Prentice-Hall, 1996.
11. Fankhauser, P., Xu, Y.: Markitup! an incremental approach to document structure recognition. Electronic Publishing, December 1993.
12. Kim, C., Seo, J.: Classifying schematic and data heterogeneity in multidatabase systems, IEEE Computer, 24 (12): 12-18, 1991.
13. Krishnamurthy, R., Litwin, W., Kent, W.: Language features for interoperability of databases with schematic discrepancies. In Proceedings of the ACM SIGMOD Conference, pages 40-49, 1991.
14. Kuikka, E.: Transformation of Structured Documents. Processing of Structured Documents Using a Syntax-directed Approach. PH.D. thesis, Computer Science and Applied Mathematics, University of Kuopio, 1996.
15. Levy, D. M., Document reuse and document systems. Electronic publishing, vol. 6(4), pages 339-348, December, 1993.
16. Naiman, C.F., Ouskel, A.M.: A classification of semantic conflicts in heterogeneous database systems. Journal of Organizational Computing, 5(2): 167-193, 1995.
17. Popa, L., Velegrakis, Y., Miller, R. J., Hernandez, M. A., Fagin, R.: Translating Web Data. In Proceedings VLDB' 02, pages 598-609, 2002.
18. Rahm, E., Bernstein, P.A.: On matching schema automatically. Microsoft Research Publications, 2001. Available at http://www.research.microsoft.com/pubs.
19. RDF Vocabulary Description Language 1.0 (RDF Schema), W3C Recommendation, 2004. Available at http://www.w3.org/TR/2004/REC-rdf-schema-20040210/
20. Sheth, A., Kashyap, V.: So far (schematically) yet so near (semantically). In Hsiao, D, K ., Neuhold, E.J., and Sacks-Davis, R., editors. In Proceedings of the IFIP WG36. Database Semantics Conference on Interoperable Database Systems (DS-5), pages 283-312,. Lorne, Victoria, Australis, North Holland, 1992.
21. WENGER, E.: Communities of Practice: Learning, Meaning and Identity, Cambridge University Press, 1998.
22. XML Extensible Markup Language, W3C Recommendation, 1998. Available at http://www.w3.org/TR/REC-XML
23. XML Schema Part 0: Primer, W3C Recommendation, 2001. Available at http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/
24. XSL Transformations (XSLT) 1.0, W3C Recommendation, 1999. Available at http://www.w3.org/TR/1999/REC-xslt-19991116