School of Engineering and Science
Jacobs University Bremen

June 15, 2007

Towards an Ontology-Driven Management of Change

Exposé of my PhD research proposal

Normen Müller
n.mueller@jacobs-university.de

Abstract

My proposed PhD thesis aims at the development of an ontology-driven management of change to support the evolution, revision and adaption of collections of technical, but informal documents. The key features of the proposed system are ontological relations between (informal) documents, extended document states, classification of change relations, and a calculus for reasoning on classified changes.

I will implement a research prototype *locutor* and evaluate it on three case studies ranging from eLearning documents to lecture notes to legal contracts. I intent to pass the former one at Rice University, the host of ConneXions. I want to integrate *locutor* into Rhaptos, the underlying software system of ConneXions, aiming at to significantly decrease the effort for maintenance of consistent document collections.

Motivation

We live in the information age: Huge amounts of information are available at our fingertips and computers influence every aspect in our lives. In particular we have to deal with an increasing amount of e-documents in research as well as in industries. Therefore the research on *Document Engineering* is concerned with principles, tools and processes that improve our ability to create, manage, and maintain documents. However only a few aspects of this broad research field found their way into practice, e.g. *document management systems* (DMS).

Current DMS are designed to coordinate the collaborative creation and maintenance process of documents through the provision of a centralized repository. The focus is primarily on managing documents themselves. Relations between and within documents as well as effect of changes on these relations are largely neglected, although information reuse and distribution could seriously benefit from such a relation management. Therefore human reviewers are needed for *management of change* (MoC), i.e. to maintain consistency after modifications. A costly, tedious, and error-prone factor in document life-cycles that is often neglected to cut cost leading to sub-optimal and often disastrous results.

To avoid resulting inefficiencies, conflicts, and delays, and to emphasize the importance of common information spaces in decentralized working environments the integration of a system support into DMS to manage modifications as well as relations is indispensable.

To sharpen our intuition about the issues involved let us consider the following situation (Figure 1): Immanuel — an author of a technical report $\mathcal{R}$ — starts writing his report with a first paragraph [a] of section 1. Then he continues writing [b] and [c] which make use of the fundamentals given in [a]. To enable other authors and interested parties to review and reuse

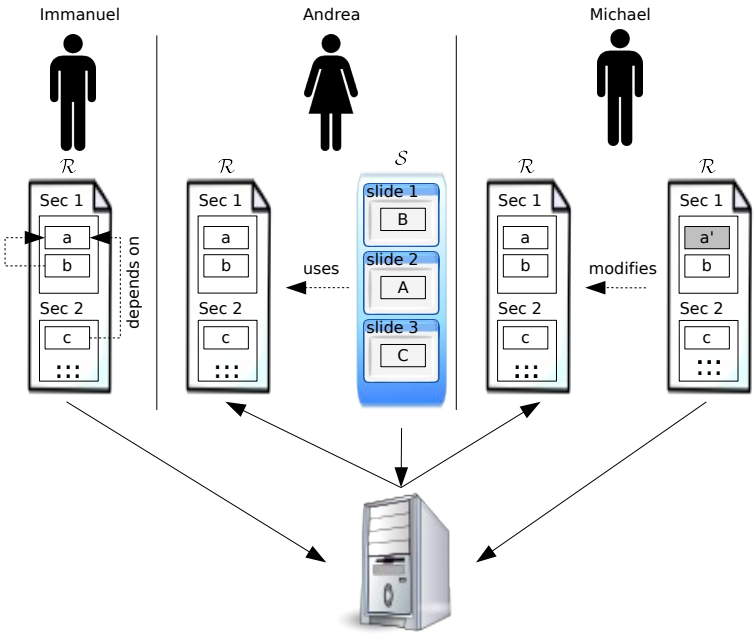

Figure 1: Actual state of DMS

his work he commits $\mathcal{R}$ to a shared DMS. Andrea — a division leader, reporting the work of her group to a client — accesses the DMS and obtains a working-copy of $\mathcal{R}$. She decides to set up some slides $\mathcal{S}$ to present the material in $\mathcal{R}$ using slides A, B and C that summarize a, b and c from $\mathcal{R}$, but orders them differently. Later, Immanuel's coauthor Michael checks out the current version of $\mathcal{R}$. He notices some discrepancies within a, modifies it to his satisfaction yielding a', and commits his revision back to the DMS.

In current DMS this is were the story ends and the problems start:

P1 Does the modified paragraph a' conflict with the unchanged b and c? Do Michael or Immanuel also have to modify b and c?

P2 What type of modifications has Michael performed, i.e., has he changed the meaning, the layout or has he simply corrected some typos?

P3 How is Andrea informed about the changes so that she does not misrepresent the state of affairs?

P4 Does Andrea need to modify slide A to account for the changes in a'? What happens to her document $\mathcal{S}$ if Immanuel adds another paragraph d to $\mathcal{R}$?

It may be illuminating to contrast what is needed to answer problems **P1** to **P4** above to current DMS that track changes in documents and support collaboration. Word processors like MS Word or OpenOffice contain simple change management functionality, which allow collaborators to record changes in the document and accept or reject changes. This supports a simple sequential collaboration model, where collaborators pass around the document like a token.

CVS[1] [CVS05] and Subversion [SVN06], free/open-source version control systems originally developed for collaboration on program sources, allow concurrent collaboration by tracking changes in a repository, and propagating them between local "working copies" of the document

[1] As Subversion is the next generation of CVS, I will concentrate on Subversion from hereon.

that can be edited by the collaborators concurrently: After a working copy has been changed, changes are committed into the repository and collaborators can update their local copies with them. SUBVERSION already offers basic solutions **S1** — **S3** for **P1** to **P3**:

S1 the notions of conflict derived from the syntactic differencing algorithms used in SUBVERSION yield a very basic form of the envisioned conflict management: two edits conflict, if they are in the same (text) line;

S2 change logging facility can be used in SUBVERSION to attach (informal) comments to changes; it is generally considered good style to comment on the gravity of changes in log messages;

S3 SUBVERSION functionality like the post-commit hooks allow one to notify the user when specific documents have changed.

I will call change management functionality at this level **weak change management** (WCM), in contrast to an envisioned **strong change management (SCM)** that I want to develop in the proposed project as an all-embracing full solution to the problems **P***i*. Let us now see why WCM is insufficient in our example situation.

S1 is insufficient, since the real danger in document management comes from **long-range** (i.e. semantic) **conflicts**, as our example shows: Michael's changes in a' can cause semantic conflicts with the unchanged b and c. With weak change management, Michael would have to check the complete document collection to determine whether his modifications are in conflict with b or c and whether and where he has to adapt these parts possibly yielding new versions b' and c', which could trigger similar changes. Without a precise notion of long-range conflicts, there is no way to determine which fragments of the document collection are affected by a change. Also notification becomes a problem: Who informs Andrea when? If she puts $\mathcal{R}$ on a watch list using techniques from **S3** she will get notified about any change in $\mathcal{R}$, even though most are irrelevant to her slides: e.g. if Michael has corrected a typo then this should be propagated to Andrea's slides. However, if Michael introduced a change in semantics then Immanuel might want to stick to his version and his versions of b and c still has to refer to the original version a.

We can state that the weaknesses of WCM approaches come from the lack of *explicitly represented relations between and within the documents* in WCM systems, i.e., copies of $\mathcal{R}$ do not "know" that b and c depend on a and copies of $\mathcal{S}$ do not "know" that $\mathcal{S}$ uses $\mathcal{R}$ and the individual slides refer to the paragraphs a, b, and c in particular.

To the best of my knowledge, current DMS (commercial ones as well as open ones) offer only weak change management facilities. Furthermore, they are usually tied to a particular document format. The SUBVERSION approach works on arbitrary document formats that allow differencing, patching, and merging of patches, e.g. ASCII text, LaTeX, and XML-based formats. Therefore I consider the SUBVERSION system as the base line against which I will evaluate the proposed methods and I will implement the proposed *locutor* [loc07] system as an extension on top of SUBVERSION.

The Approach

In the following I will give a brief survey of my approach to develop an ontology-driven management of change [Mül06] integrated into *informal* document engineering processes.

Structured View of Documents. There are a multitude of approaches to add structural markup for documents and document collections. Early representatives are the TeX/LaTeX format [Knu84, Lam94] for mathematic/scientific documents, which adds codes that describe the document structure as control sequences, which are interpreted by a formatting engine.

SGML [Gol90] is similar to TEX/LaTeX in spirit, but tries to give the markup scheme a more declarative semantics (as opposed to the purely procedural – and rather baroque – semantics of TEX) to make it simpler to reason about (and thus reuse) documents. In the past few years the XML format has established itself as a general basis for markup languages, so I will concentrate on XML-based document formats here. I propose to base MoC, information reuse, and consistency on a *structured view of documents*. In this context I regard *documents* as *self-contained structured compositions of information units*. For the purpose of this proposal one can pragmatically think of information units as "*tangible/visual text fragments potentially adequate for reuse*" constituting the content of documents. To distinguish the term "information unit" between common speech and the ontological concept, I will call from now on the ontological concept INFOM.

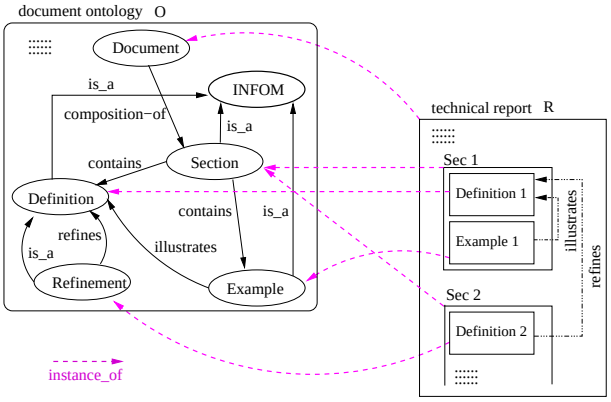

Figure 2: A document ontology $\mathcal{O}$

Following [KBM06] I will describe a document format and its way to structure documents (i.e. the relations between the INFOMs) in terms of a *document ontology* (Figure 2). This is an ontology that formalizes document structure (e.g. section, paragraph) rather than the document contents and is used to classify the type of documents. This provides a notion of consistency and invariants that allows one to propagate effects of individual changes to entire documents. Conversely, the ontology will provide means to localize effects of changes by introducing a notion for semantic dependencies between document parts.

Two-Layered Two-Dimensional View of Documents. Following the OMDOC [Koh06] approach[2] I will separate documents into two layers (Figure 3) both under version control: A *narrative* and a *content* layer both of which consist of INFOMs and are composed via relations. The presentational order of information units in documents is represented on the narrative layer whereas the information units themselves and the ontological relations between them are placed in the content layer. The connection between the narrative and the content layer is represented via *narrative relations* (analogous to symbolic links in UNIX). The information units and the ontological relations build up the "content commons" [CNX07]. I will use the term NARCON for the graph representations of document collections consisting of a narrative layer and a content layer.

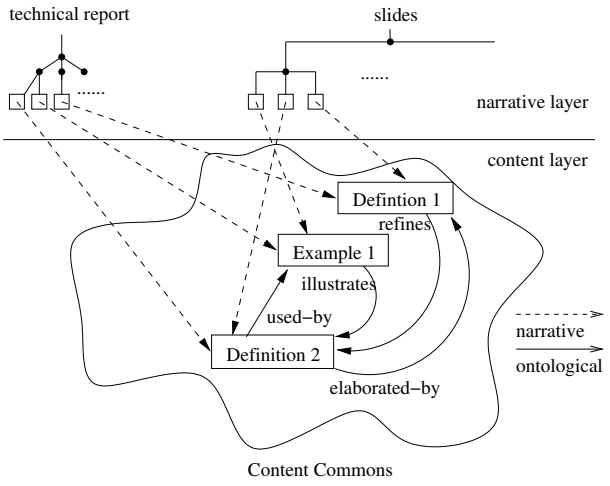

Figure 3: Narrative and Content Layer

Following the initial work in the MMiSS [MMi] project, I will also model the concept of *variants*. This expands the application area not only "in-the-breadth" but also "in-the-depth". Thus, by extending the well-known concept of *versions* and *revisions* by the concept of variants, the life-cycle of documents will no longer be only along a horizontal time line but also along a vertical line of variants. On the document level I call the concept of versions, revisions, and variants *document states*.

[2]The OMDOC group does do not claim to have invented this concept, it is part of the XML folklore and can already be found e.g. in [VD04]. But the OMDOC format probably implements this idea in the cleanest way.

Computation of Structural Differences. I propose to base my computation of structural differences on the insights of XML-`diff` tools and the initial work of [EK04]. According to this I will extend the `diff`–algorithms and unification-based techniques, proposed there, to operate on NarCons resulting in a $\mathcal{M}$`Diff`–algorithm, i.e. a model based `diff`–algorithm comprising an equality theory on NarCons. Therewith *`locutor`* will be able to identify syntactically different Infoms to be semantically equal and thus to minimize the number of Infoms affected when changing Infoms *(Equality Theory)* and to frame the syntactical representation of Infoms and thus to help to locate changes of Infoms relative to the internal structure *(Syntactical Structure)*.

Reasoning on Changes. In the first step, to compute the *long-range effect of changes* the *`locutor`* system will enable authors to *classify* computed structural differences. Therefore I propose a MoC ontology comprising a *taxonomy of change relations*. The connection between a document ontology and MoC ontology will be modeled in a so-called *system ontology*. It is one of the central intuitions behind this proposal that SCM techniques can be based on information that can be expressed in system ontologies. I claim that the *`locutor`* system only needs the system ontology part of a fully formal domain semantics. Thus system ontologies will be the central means for extending the SCM methods to the structured, two-layered and two-dimensional document setting.

In the second step, the *`locutor`* system will *reason on classified structural differences* utilizing inference rules consolidated in a *change relation calculus* based on a system ontology.

Prototype System. I will implement the MoC approach in the *`locutor`* prototype system. This implementation will progress in parallel with theory development and serves as a continual reality check to evaluate the concepts. For the latter, I will undertake three case studies ranging over differing domains, representation formats and base systems (cf. section "Case Study: ConneXions").

Objectives

The objectives of my thesis work are:

O1 Modeling system ontologies to be open to any (specific) application area.

O2 Capturing of ontological relations between information units to enable management of change "information_unit-by-information_unit" rather than "line-by-line".

O3 Computation of effects of changes subject to classified change relations, i.e., identification of semantic long-range conflicts.

O4 Identifying exactly *when*, *where*, *why*, and *by what* updates corrupt documents w.r.t. structural and ontological relations.

O5 Extension of document states by a second dimension, i.e., to consider not only different versions and revisions of information units — the first dimension — but also different variants.

O6 Integration of management of change into arbitrary DMS without requiring adaptations to document engineering processes, i.e., authors are not required to adapt their editing practices (cf. section "Case Study: ConneXions").

Summary, I hope to seriously facilitate information consistency, reuse, and thus information distribution by implementing a management of change regarding the complex relations between document states.

References

[CNX07] Connexions. Project homepage at `http://www.cnx.org`, seen February 2007.

[CVS05] Concurrent Versions System: The open standard for Version Control. Web site at `http://www.cvshome.org`, seen August 2005.

[EK04] Frederick Eberhardt and Michael Kohlhase. A Document-Sensitive XML-CVS Client. unpublished KWARC blue notes, 2004.

[Gol90] C. F. Goldfarb. *The SGML Handbook*. Oxford University Press, 1990.

[KBM06] Bernd Krieg-Brückner and Achim Mahnke. Semantic Interrelation and Change Management. In OMDoc – *An open markup format for mathematical documents [Version 1.2]* [Koh06], chapter 26.6, pages 274–277.

[Knu84] Donald E. Knuth. *The TEXbook*. Addison Wesley, 1984.

[Koh06] Michael Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in LNAI. Springer Verlag, 2006.

[Lam94] Leslie Lamport. *LaTeX: A Document Preparation System, 2/e*. Addison Wesley, 1994.

[loc07] *locutor*: An Ontology-Based Management of Change, seen June 2007. system homepage at `http://www.kwarc.info/projects/locutor/`.

[MMi] MMISS: Multimedia in Safe and Secure Systems. Web site at `www.mmiss.de`.

[Mül06] Normen Müller. Towards an Ontology-Driven Management of Change – Research proposal for a Ph.D. thesis. `http://kwarc.info/nmueller/papers/resprop.pdf`, 2006.

[SVN06] The Subversion Project. Web site at `http://subversion.tigris.org/`, seen August 2006.

[VD04] Katrien Verbert and Erik Duval. Towards a Global Component Architecture for Learning Objects: A Comparative Analysis of Learning Object Content Models. In *Proceedings of the EDMEDIA 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 202–208, 2004.