# An Ontology-Driven Management of Change*

**Normen Müller**
International University Bremen
D-28717, Bremen, Germany
n.mueller@iu-bremen.de

## Abstract

Current document management systems (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 and effects of changes are largely neglected. To avoid inefficiencies, conflicts, and delays the support of modification management is indispensable.

Here I present the design of the LOCUTOR system that aims to provide management of change functionality for arbitrary XML documents ranging from *informal*, e.g. instruction or construction manuals, to *formal* documents.

## 1 Introduction

We live in the information age: Huge amounts of information are available at our fingertips and computers influence every aspect in life. In particular we have to deal with e-documents everywhere. *Document engineering*,

> is the computer science discipline that investigates systems for documents in any form and in all media. As with the relationship between software engineering and software, document engineering is concerned with principles, tools and processes that improve our ability to create, manage, and maintain documents [DocEng, 2006].

Of this broad field only small parts have 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.

*Research Proposal of a PhD thesis

### 1.1 A Running Example

To sharpen our intuition about the issues involved let us consider the following situation: Immanuel — a coauthor of a technical report $\mathcal{R}$ — is responsible for some sections therein. He starts writing with some fundamentals $\boxed{1}$ and then builds on that: $\boxed{2} \rightarrow \boxed{1} \leftarrow \boxed{3}$. To enable other authors and interested parties to review and reuse 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 $\mathcal{S}$ based on Immanuel's parts of $\mathcal{R}$ in a different order. After a while Immanuel's coauthor Michael checks out the current version of $\mathcal{R}$. He notices some discrepancies within $\boxed{1}$, modifies it to his satisfaction yielding $\boxed{1}$, and commits his revision back to the DMS.

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

**P1** How to decide whether the modifications of $\boxed{1}$ conflict with the unchanged $\boxed{2}$ and $\boxed{3}$? So do Michael or Immanuel also have to modify $\boxed{2}$ and $\boxed{3}$?

**P2** How to decide what sort of modifications Michael performed, i.e., did he modify the meaning, the layout, or did he just correct some typos?

**P3** How to decide whehter Andrea has to be notified so that she does not mis-represent the state of affairs?

**P4** How to decide whether Andrea actually does need the modified version of $\boxed{1}$?

Recapitulating the problem:

*Relations between and within the documents are not represented in current* DMS. (†)

i.e., copies of $\mathcal{R}$ do not display the fact that $\boxed{2}$ and $\boxed{3}$ depend on $\boxed{1}$ and copies of $\mathcal{S}$ do not display the fact that $\mathcal{S}$ uses $\mathcal{R}$ and $\boxed{1}$, $\boxed{2}$, and $\boxed{3}$ in particular.

Thus current DMS do not solve (**P1**) – (**P4**)! Immanuel would have to contact Michael to get detailed information of the applied modifications or he would have to completely re-read $\boxed{1}$ and verify on his own if the modifications are in conflict with $\boxed{2}$ or $\boxed{3}$. So this workflow becomes tedious and error-prone. In particular there is still the open question: Who informs Andrea? Neither Immanuel nor Michael are aware of the fact Andrea is setting up some slides partially based on their technical report. Thus, Andrea has to inform herself, i.e., continuously check the state of $\mathcal{R}$ and verify by herself if, regarding her slides, the applied modifications are significant.

To avoid these 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.

## 2 A Structured View of Documents

I use a *structured view of documents* to facilitate MoC, information reuse and consistency. In contrast to file- and line-based systems like the Subversion system [SVN, 2006], I consider documents as structured collections of information units. In this context I define w.l.o.g. a *document* as *a self-contained* XML-*based composition of information units*.

Probst et al. [Probst *et al.*, 1997] posits that to obtain meaning from a single *data* element, e.g. a formula or a quantity, we need another component: We need some *context* for its interpretation (see [Kohlhase and Kohlhase, 2005] for a deeper explanation). That is why "self-contained" is part of the definition.

The reason why I base the definition on XML formats is on the one hand that many standard formats are already available as XML and others can easily be defined via DTD, XML Schema or RelaxNG. On the other hand I want to foster open, structural document formats and leverage context indication in the form of content markup. Furthermore by using XML-based document formats some structural information like information units being a constituting part of another information unit is already straightforward given by the syntax.

This combination of content markup and information units makes it a document by my definition.

The following sections describe how I propose to identify data elements in the notion of information units and how to define non-grammatical relations between them. Based on that I present a two-layered view of documents which I will finally expand to a *two-layered two-dimensional view*.

### 2.1 Informations Units and Ontological Relations

For *formal* documents like specifications or programs the relations between information units, e.g. routine/subroutine, are quite clear and various structuring operations have been proposed for modularization. Main motivations for modularization have been the sharing of subspecifications within one specification, the reuse of specifications, and the structuring of proof obligations. Furthermore the structure of specifications can also be exploited when the effects of changes are analyzed [Mossakowski *et al.*, 2006]. Therefore some initial research has been conducted on methods and tools [Autexier *et al.*, 2002; Mossakowski, 2005] managing the consistency and change of formal documents. However, all these systems base their MoC on the inherent underlying (formal) mathematical structure of the documents.

For handling *informal* documents the situation is completely different. The grammatical and non-grammatical relations between and within the documents are rather clear to humans, but how to make these machine understandable?

Therefore I propose to use knowledge representation (KR) methods, in particular on the notion of a *system ontology* [Krieg-Brückner *et al.*, 2004b][1]. This is an ontology describing the data model of a system or the representation language the system and its applications are based on independently of their respective syntactical realization. Thus I am not bound to any specific document format but yet able to capture semantic interrelations, e.g. *illustrates*, *refines* or *depends-on*, even between (fragments of) informal documents.

For representing ontologies various artificial languages and notations have been proposed. I use Description Logics (DL), a family of knowledge representation languages that can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. A KR system based on DL provides facilities to set up knowledge bases, to reason about their content, and to manipulate them. A knowledge base (KB) comprises two components, the TBox and the ABox. The TBox introduces the terminology, i.e., the vocabulary of an application domain, while the ABox contains assertions about named individuals in terms of this vocabulary. As statements in the TBox and in the ABox can be identified with formulae in first-order logic (FOL)[2] the description language has a model-theoretic semantics — that is an "*account of meaning in which sentences are interpreted in terms of a model of, or abstract formal structure representing, an actual or possible state of the world*" [Matthews, 1997]. Thus a KB is equivalent to a set of axioms in first-order logic and like any other set of axioms, it contains implicit knowledge that can be made explicit through inferences[3].

So in order to maintain consistency within and between documents after modifications, i.e., to reason on changes, I represent a system ontology inside the Locutor system as the TBox, while the ABox is dynamically synthesized out of the documents and the information units in particular[4].

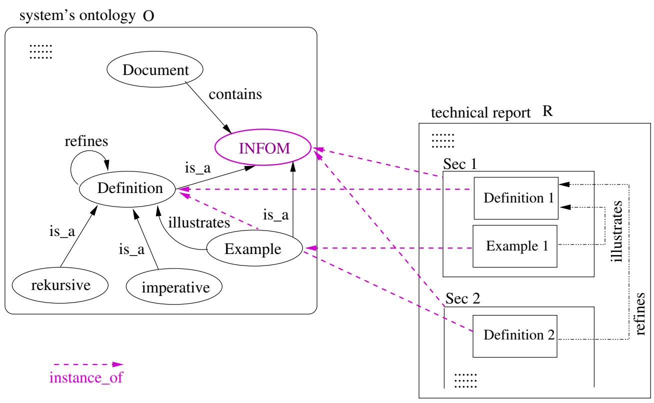


Figure 1: INFOMs and Ontological Relations

To identify information units, I predefine the concept of information units to be part of any (user-defined) system ontology (Figure 1). A concrete elaboration of the term "information unit" is a further part of the research I want to undertake. For the purpose of this article one can pragmatically think of information units as "*tangible/visual text fragments potentially adequate for reuse*" constituting

[1]Called *system's ontology* there.

[2]Note: DL is a decidable fragment of FOL!

[3]Technical term in the DL world: reasoning.

[4]Note, there will be *a* ABox for each document, so that *the* ABox constitutes the union of all respective documents.

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[5]. To distinguish between *grammatical* and *non-grammatical*[6] relations, I call the latter *ontological* relations and subsume both by the term *structural* relations.

To clarify the terms INFOM and *ontological* relations let us recall our running example (cf. section 1.1). We presume one of the authors of the technical article $\mathcal{R}$ has established a system ontology $\mathcal{O}$ declaring all concepts and relations of the domain of interest $\mathcal{R}$ is related to, e.g., an ontology describing the concepts of a customer requirements specification. Now, Immanuel does have the ability to “tag” his fragments of $\mathcal{R}$ with concepts of $\mathcal{O}$ (Figure 1). Thus, he is able to explicitly identify information units: 1 is an individual of the concept “definition” Def, 2 is an individual of the concept “example” Ex *illustrating* the first Def, and 3 is also an individual of the concept “definition” Def but *refining* the first one. Note, regarding the pragmatic definition of information units, Immanuel is also able to “tag” grouping elements within $\mathcal{R}$, e.g. sections and paragraphs, by concepts of $\mathcal{O}$.

Thus, by making information units and relations between them explicit, we solved the former problem (†)[7].

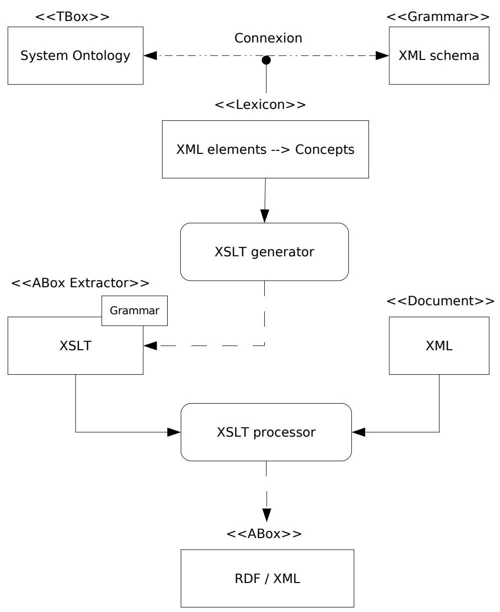


Figure 2: Connexion between concepts and XML elements

The open question is how to design the connexion between a system ontology and respective XML documents, i.e., how to dynamically synthesize an ABOX out of a document. In order to leave the documents untouched, I suggest a *stand-off markup*. Markup is said to be stand-off, or external, when the markup data are placed outside of the text it is meant to tag. The markup therefore points to, rather than wraps, the relevant data content. Therefore I will develop a meta-language to set up a *lexicon* describing the connexion between concepts of a system ontology and XML elements. An XSLT generator then builds up — based on such a lexicon and the respective grammar — an XSL transformation, say an ABOX EXTRACTOR. Finally a ABOX of a respective document is synthesized by an XSLT processor and encoded in RDF/XML. Such a generated RDF/XML document constitutes the stand-off markup. The reason why I propose to use XSLT is that this language is in particular designed to map XML to XML and here to map a XML document markup to RDF/XML respectively.

## 2.2 Narrative and Content Layer

Following [Verbert and Duval, 2004] and [Kohlhase, 2006] I separate documents into two layers: A *narrative* and a *content* layer both of which consist of INFOMs and are composed via relations. The pictorial representation of the two layers is given in Figure 3.

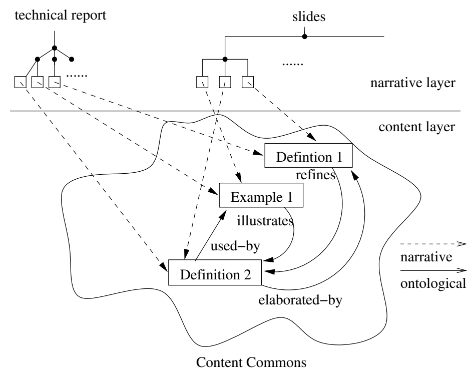


Figure 3: Narrative and Content Layer

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[8]. 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” [CNX, 2006]. Thus we clearly separate the conceptual level from the discourse presentation level.

Figure 4 consolidates the classes of relations we defined so far. Structural relations $\mathcal{SR}$ subsume grammatical $\mathcal{GR}$ and ontological relations $\mathcal{OR}$. As to the fact a system ontology describes the data model behind the representation format the grammatical relations have to be

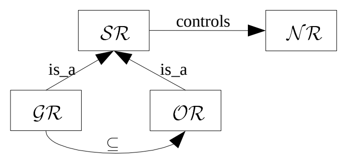


Figure 4: Taxonomy of Relations

[5]A little word-play on “atom”. I use the word “atom” in terms of not being further divisible.

[6]Relations defined within a system ontology.

[7]Being aware of the facts that documents and their generation/use are part of socio-technical processes and bridging the knowledge gap between mental models and knowledge representation is a problem the AI community tries to solve for half a century now, I aspire to achieve further insights through my case studies (cf. 4) to set up automated annotation tools to keep authors handcrafted annotation to a minimum.

[8]How far information units could also emerge on the narrative layer is a further research I want to undertake.

a subset of the ontological relations. Narrative relations $\mathcal{NR}$ are controlled by structural relations, i.e., the order of referenced INFOMs is verified. For example, without a previous definition the usage of a technical term within a technical report does not make sense.

To clarify the significance of such a layered view of documents, let us go back to our running example. For simplicity we assume the initially identified information units are derived from the technical report $\mathcal{R}$. Thus Andrea — the author of the slides — does not have to copy these information units but rather just "links"[9] to them. Only the new order of the old information units within the new information product is stored on the narrative layer and narrative relations refer to the respective information units already stored on the content layer.

Note, by assembling information units and respective structural relations we build up the foundations for a interdisciplinary information pool, i.e., pooling of information units related to various domains of interest. Therefore in further research I will also investigate how to compose documents of heterogeneous[10] INFOMs to provide information harvesting at a highest level.

So up to now we have reached a *two-layered view of documents* but have neglected the *ontological* relations between the identified information units so far! Only by using this additional information we will be able to establish a consistent and expressive management of change, i.e., we will be able to handle dependencies between information units and compute the effects of changes on theses dependencies (cf. section 3). Therefore let us look back on the situation in our scenario where Michael is modifying information unit 1, say the first Def. Now he is aware of the interrelations between the different parts of $\mathcal{R}$, in particular LOCUTOR will notify him about the fact that 2 and 3 depend on 1. Furthermore, by recognizing the narrative relations, LOCUTOR can also notify Andrea about the modifications (**P3**). We will discuss how to solve (**P1**), (**P2**) and (**P4**) in section 3.

### 2.3 The Concept of Variants

Following initial work in the MMiSS [Krieg-Brückner *et al.*, 2004a] project, in my approach I am also aware of the concept of *variants* [Mahnke and Krieg-Brückner, 2004]. 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*. I will model the concept of variants by expanding the (default) set of ontological relations by a further one called *variant-of*.

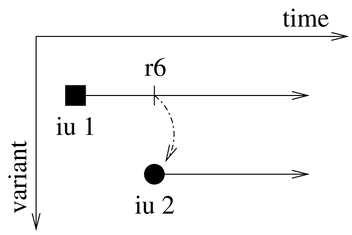

Figure 5: The Variant Dimension

To demonstrate the dimension of variants in a more "dimensional" way Figure 5 depicts another possible scenario: After modifying any information unit $iu_1$ several times (up to revision number $r_6$[11]) another user or the initial user herself decides to develop a variant of $iu_1$. To keep it simple one can imagine $iu_2$ to be a "language-variant" of $iu_1$, e.g. $iu_1$ is written in English and $iu_2$ in German. By an user annotating information unit $iu_2$ to be a variant of information unit $iu_1$ we will be able to build up a complete management of variants, i.e., the states and changes of the original information unit, the variants, and all relations between any of them will be managed as well.

To sharpen the notion of the term *variant* in our running example let us go back to Andrea. Remember she wanted to set up some slides $\mathcal{S}$ regarding Def, Ex, and Def from the technical report $\mathcal{R}$. However, in general slides

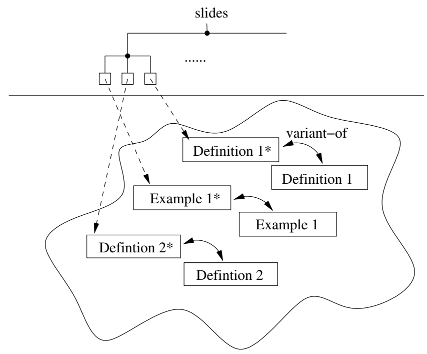

Figure 6: Variants of Infoms

represent a different, say more compact presentation of information. So Andrea will not use the INFOMs one-to-one, but rather modify them to "fit" her presentation. Figure 6 demonstrates the described situation[12]. Andrea is now able to characterize her new information units and the relations between $\mathcal{S}$ and $\mathcal{R}$ still hold.

Based on the arising complex network between documents and information units, respectively, I also propose to integrate value-added services into LOCUTOR. E.g. one of them identifies most referenced INFOMs to capture "useful" and "valuable" information units. Thus I recognize a further open research question: How to enable authors to search the content commons[13], i.e., how to handle the following scenario: Let there be an article $\mathcal{A}_1$ consisting of INFOMs $\Lambda$ and $\Omega$. Now another author wants to write an article $\mathcal{A}_2$ also using $\Lambda$. How do we assist the second author? Does he have to check out $\mathcal{A}_1$, copy-and-paste $\Lambda$ into $\mathcal{A}_2$ and LOCUTOR will take care to identify that $\Lambda$ is already inside the content commons? And, in particular, how does the author get to know that $\Lambda$ exists, anyway? Therefore I hope the case study (cf. section 4) will uncover authors' requirements.

## 3 MoC on NarCons

Up to know we have elaborated a structured view of (informal) documents. It appears that this two-layered, two-dimensional view is represented by a graph consisting of

[9]Concretion of "links" between entire documents is a further part of the research I want to undertake.

[10]INFOMs declared in different system ontologies.

[11]Think of the well-known SUBVERSION work-flow.

[12]I omit further ontological links for a better readability.

[13]Here I will consider results achieved in the case-based reasoning community.

a narrative layer and a content layer to be called NARCON here. Thereby we have already facilitated information reuse.

Now, I will describe first ideas towards a management of change on NARCONs to achieve *consistent* information reuse, i.e., a MOC on NARCONs to maintain consistency during the development of various document states. Thus this section is less a report on solutions than an attempt to publicize first suggestions towards a consistent management of change. Figure 7 depicts a survey of the proposed MOC system.

Documents will have to identify the underlying language $\mathcal{L} := \langle \mathcal{M}, \mathcal{O} \rangle$ they are an instance of: I will regard a lan-

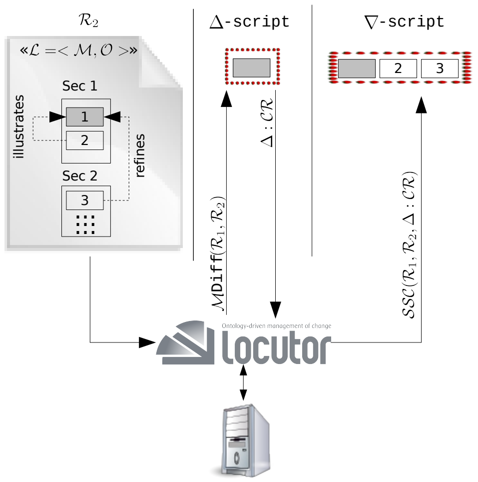


Figure 7: Management of Change

guage to be a pair consisting of a model $\mathcal{M} = (\mathcal{G}, \mathcal{E})$ and a system ontology $\mathcal{O}$. A model $\mathcal{M}$ consists of a grammar $\mathcal{G}$ together with an equality theory $\mathcal{E}$. $\mathcal{G}$ defines the syntactical rules to build up valid document and following the initial work in [Eberhardt and Kohlhase, 2004] $\mathcal{E}$ defines when two NARCONs are considered to be equal. Thus the system will use the model $\mathcal{M}$ to compute structural differences $\Delta$ between two document states (cf. section 3.1). Following the MMISS project [Krieg-Brückner *et al.*, 2004a] I will use a system ontology $\mathcal{O}$ to describe further semantic dependencies (cf. section 2.1). The LOCUTOR system will use this additional information to compute long-range effects of changes (cf. section 3.2). Furthermore to operate on representatives rather than on singletons I propose a taxonomy of change relations $\mathcal{CR}$ (cf. section 3.2) to enable authors to classify $\Delta$. So to systematically reason on such a classified $\Delta$ (cf. section 3.2), say to compute the structural semantic[14] closure ($\mathcal{SSC}$) $\nabla$ of each classified $\delta \in \Delta$ I will develop inference rules consolidated in a change relation calculus $\mathcal{CRC}$. Subject to an $\alpha \in \mathcal{CR}$ and a $\delta \in \Delta$ the $\mathcal{SSC}$ of an information unit $iu$ regarding $\delta : \alpha$ denotes all further information units affected by the modifications w.r.t. their structural relations to $iu$.

In particular I want to bring into light that annotating is rewarded by getting even more automatic assistance in the

[14] I use the term "structural semantics" in sense of marking-up the meaning by structure, i.e., the meaning of an information unit is obtained by its relations to other information units. I do not need any other entailment relation to model semantics but rather concentrate just on the structure.

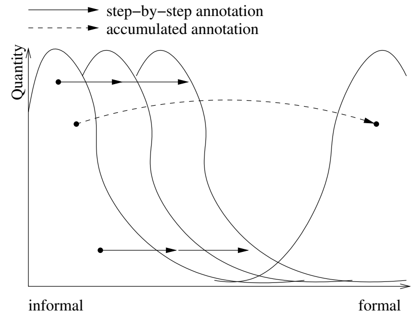


Figure 8: The Shifting Wave

future:

> *"The flatter a document the less the assistance!"*

Figure 8, called the "The Shifting Wave", depicts this slogan. In my approach I want to lead authors on the one hand to annotate informal documents step-by-step, i.e., to provide informal documents more and more with structural semantics and on the other hand to annotate their modifications. As a consequence of each single step the wave shifts a little bit more towards the formal world and thus can be better kept under control by formal systems, i.e., the computation of long-range effects is improved. But note, I do not want to ask too much of authors all at once! It is up to an author to which level she will annotate her changes.

### 3.1 Computation of Structural Differences

I propose to base the computation of structural differences on the insights of XML-`diff` tools and the initial work of [Eberhardt and Kohlhase, 2004]. According to this I will transform `diff`–algorithms and unification-based techniques, proposed there, to operate on NARCONs.

The first suggestion for such a computation of structural differences is to define a function $\mathcal{M}\texttt{Diff} : \mathcal{D} \times \mathcal{D} \rightarrow \Delta$, where $\mathcal{D}$ denotes NARCON-graphs and $\Delta$ a `diff`-script comprising structural differences between NARCON-Graphs.

With "$\mathcal{M}$" in the function name I want to stress to model a strong semantic notion of equality to generate more compact and less intrusive edit scripts. For instance, if we know that whitespace carries no meaning in a document format, two documents are considered equal, even if they differ (with respect to the distribution of whitespace characters) in every single line; as a consequence, $\Delta$ would be empty. This motivates the following general statement of the problem at hand [Eberhardt and Kohlhase, 2004]:

> **The General Difference Computation Problem (DCP)**: Let $\mathcal{K}$ be a class of NARCONs and an equality theory $\mathcal{E}$ on $\mathcal{K}$. Given two NARCONs $\mathcal{S}$ and $\mathcal{T}$, find an optimal edit-script that transforms $\mathcal{S}$ to $\mathcal{T}$.

In particular I will engage myself in the general DCP modulo an equality theory ($\mathcal{E}$-DCP) left unsolved in [Eberhardt and Kohlhase, 2004].

To exemplify the functionality of $\mathcal{M}\texttt{Diff}$ let us go back to our running example. If we apply $\mathcal{M}\texttt{Diff}$ on $\mathcal{R}$ after the modifications initiated by Michael the output of $\mathcal{M}\texttt{Diff}(\mathcal{R}_1, \mathcal{R}_2)$ would be $\Delta = \{$ ▭ $\}$.

Up to this stage I want to point out that I did not use any ontology-based information[15], but only operate on properties defined in $\mathcal{M}$. Furthermore I want to stress that I will not handle information units in terms of a "black box", but consider changes within the inner structure as well as in the content, e.g. modifications on the actual text of $\boxed{\text{Def}}$. So one could say, that we have achieved a NARCON-based variant of SUBVERSION so far.

But let us now consider a situation where Michael modified the meaning of $\boxed{\text{Def}}$. The output of $\mathcal{M}\texttt{Diff}$ would be the same, omitting $\boxed{\text{Ex}}$ and the second $\boxed{\text{Def}}$, which is correct but unsatisfying.

In the next section I will explain how I propose to extend $\Delta$ to also capture the structural semantic closure of structural differences.

## 3.2 Computation of Long-Range Effects of Changes

By regarding *all* relations in general and the ontological relations in particular the system will be able to compute long-range effects of changes and give authors significant feedback of the impact of their modifications. That is, identifying exactly *when*, *where*, *why*, and *by what* updates could corrupt documents w.r.t. the structural relations. Thus not only the directly affected information unit is reported to the user, but all structural related ones as well w.r.t. the classification of $\Delta$.

**A Taxonomy of Change Relations**

In order to be able to reason on changes, say to reason on $\Delta$, I will develop a taxonomy of *change relations* $\mathcal{CR}$ to classify structural changes. As to the matter of fact the implementation of an automatism to classify structural changes is "AI-hard" I will enable authors to annotate $\Delta$ with $\mathcal{CR}$ (short: $\Delta : \mathcal{CR}$). Note, by this additional information about structural changes we solve (**P2**)! So we extend the *two-valued states* of changes, i.e., modified and non-modified, to *annotated two-valued states* of changes. To clarify the notion of a $\mathcal{CR}$-taxonomy I demonstrate a "first-try-example" (see Figure 9) for a toy example.

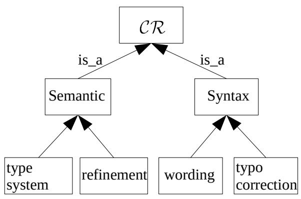


Figure 9: A $\mathcal{CR}$-taxonomy

To demonstrate the emphasis of classified change relations, let us recall our running example, again especially regarding Michael: He modified the first $\boxed{\text{Def}}$ without being aware of the fact that other information units depend on this one. We already solved this problem with the new view of documents and NARCON-graphs, respectively. However, so far we are only able to notify Michael and Immanuel about the fact that there are some dependencies, rather than to notify them about the effects of Michael's modifications on these dependencies. So if Michael now classifies his modifications to be syntactical, e.g. typo corrections, the system will compute and fix these changes with respect to the structural relations defined in $\mathcal{L}$, i.e., the system will merge the typo corrections into the next document state of existing working-copies just like in the SUBVERSION approach. If, however, Michael classifies his changes to be semantical, e.g. if he changed the entire type system of the first $\boxed{\text{Def}}$, the situation to fix such a modification changed! So in order to compute and manage the long-range effects of (semantic) changes I will elaborate a system for *reasoning on classified structural changes*.

**Reasoning on Classified Structural Differences**

To systematically reason on annotated changes, say to reason on $\Delta : \mathcal{CR}$, I will develop inference rules consolidated in a $\mathcal{CR}$-Calculus ($\mathcal{CRC}$) operating on NARCONS. Regarding the proposed calculus I will build on the $\mathcal{DG}$–calculus operating on development graphs [Hutter, 2000] to evaluate what properties and rules can be adopted for NARCONS. A main aspect in this analysis will be the structural properties of development graphs and the calculus itself. Then, based on the $\mathcal{CRC}$, I propose to deduce the effects of changes on structural relations, i.e., with these "rules of re-action to changes" at hand I will define an algorithm to compute for each $\Delta : \mathcal{CR}$ (short: $\ddot{\Delta}$) the structural semantic closure ($\mathcal{SSC}$) $\nabla$, that is, all information units structurally related to the ones explicitly affected by $\Delta$. Therefore I propose another function with the following signature:

$$\mathcal{SSC} : \mathcal{D} \times \mathcal{D} \times \ddot{\Delta} \rightarrow \nabla$$

Here $\nabla$ extends $\Delta$ in the sense of $\nabla := \Delta \cup \{(iu, \texttt{trace}(iu)) | iu \in \mathcal{IU}_\mathcal{O}\}$, where $\mathcal{IU}_\mathcal{O}$ denotes the set of semantically affected INFOMS and $\texttt{trace}(iu)$ represents the path of involved ontological relations.

To clarify the functionality of the suggested $\mathcal{SSC}$ function, let us again take our running example into account but now let us assume Michael changed the meaning of the first $\boxed{\text{Def}}$, e.g. he classifies his changes to be a modification to the type system of $\boxed{\text{Def}}$ denoted by the $\mathcal{CR}$ concept $\mathcal{TS}$. So $\mathcal{SSC}$ would compute

$$\mathcal{SSC}(\mathcal{R}_1, \mathcal{R}_2, \boxed{\phantom{Def}} : \mathcal{TS}) =$$

$$\{\boxed{\phantom{Def}}, (\boxed{\text{Ex}}, \textit{illustrates}), (\boxed{\text{Def}}, \textit{refines})\}$$

So we finally solved (**P1**) and (**P4**) and are able to give answers to the until now outstanding question "How does one $\Delta$ affect existing relations and how do existing relations affect the computation of $\nabla$, respectively?"

As can seen from the illustrative running example the "great challenge" of my thesis is

- to define *ontological relations* for MOC, e.g. a possible additional relation might be *adapted-analogously*, to facilitate authors to augment their informal documents by more *structural semantics*
- to define proper *change relations* to "characterize" modifications
- to define a calculus parameterized by *classified change relations* operating on NARCONS

in order to compute how changes will be reflected onto the pool of information units of composing documents. I hope the result will improve consistent information reuse and distribution.

# 4 Case Study

I will undertake three case studies to evaluate applicability of my proposed system:

[15] If one wants to involve ontologies at this stage this would correspond to the creation of an ontology $\mathcal{O}$ with just a concept "document" "is_a"-related to the concept *infom*.

**The Lecture Study** A "NARCON-like" approach has already been successfully used within the sTEX project [Kohlhase, 2005] to enable authors to add semantic information to documents without changing the visual appearance. A large corpus of slides for the lecture General Computer Science I & II at International University Bremen have been marked up by my supervisor MICHAEL KOHLHASE using sTEX. But the project currently lacks any management of change! So this gives me a great ability to test my suggestions on a large amount of data.

**The e-Learning Study** The `Connexions` e-Learning system is a rapidly growing collection of free scholarly materials and a powerful set of free software tools to help *authors* publish and collaborate, *instructors* rapidly build and share custom courses, and *learners* explore the links among concepts, courses, and disciplines [CNX, 2006]. As a matter of fact that during my thesis I am sponsored by the EU-project `ONCE-CS` [ONCE-CS, 2005] to integrate OMDOC [Kohlhase, 2006] into the `Connexions` projects. Besides integrating my MOC into the system, I will add more structural semantics to the corpus of this projects via the OMDOC system ontology to improve the links among concepts, courses, and disciplines.

**The Wiki Study** SWIM [Lange and Kohlhase, 2006] is a semantic wiki for collaboratively building, editing and browsing a mathematical knowledge base. Its pages, containing mathematical theories, are stored in OMDOC format. This project is currently being developed by CHRISTOPH LANGE for his master thesis. CHRISTOPH LANGE is an upcoming Ph. D. student in the KWARC group (`http://kwarc.eecs.iu-bremen.de/`) and so I hope to benefit from his collaborations and the SWIM user interface on the one hand and to assist his work with my MOC on the other hand.

## Acknowledgments

The author would like to thank Dieter Hutter, Michael Kohlhase, and Christoph Lange for stimulating discussions in the early stages of this work. Moreover I want to thank the anonymous reviewers for giving me valuable feedback and comments in the course of the review process. So in my future work I will consider the recommended PhD thesis of Makus Nick [Nick, 2005] and his legitimated complaint on the lack of discussing the complexity of my proposed system I could not answer in this early stage, however.

## References

[Autexier *et al.*, 2002] S. Autexier, D. Hutter, T. Mossakowski, and A. Schairer. The Development Graph Manager MAYA, 2002.

[CNX, 2006] CONNEXIONS. Project home page at `http://www.cnx.org`, seen August 2006.

[DocEng, 2006] The ACM Symposium on Document Engineering. Web site at `http://www.documentengineering.org`, seen April 2006.

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

[Hutter, 2000] Dieter Hutter. Management of Change in Structured Verification. In *Proceedings 15th IEEE International Conference on Automated Software Engineering*, number 2000 in ASE, pages 23–34. IEEE Computer Society, 2000.

[Kohlhase and Kohlhase, 2005] Andrea Kohlhase and Michael Kohlhase. An Exploration in the Space of Mathematical Knowledge. In Michael Kohlhase, editor, *Mathematical Knowledge Management, MKM'05*, number 3863 in LNAI. Springer Verlag, 2005.

[Kohlhase, 2005] Michael Kohlhase. Semantic markup for TEX/LATEX. Manuscript, available at `http://kwarc.eecs.iu-bremen.de/software/stex`, 2005.

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

[Krieg-Brückner *et al.*, 2004a] B. Krieg-Brückner, B. Krämer, D. Basin, J. Siekmann, and M. Wirsing. Multimedia Instruction in Safe and Secure Systems. Abschlussbericht, Universität Bremen, 2004. BMBF project 01NM070, 2001-2004.

[Krieg-Brückner *et al.*, 2004b] Bernd Krieg-Brückner, Arne Lindow, Christoph Lüth, Achim Mahnke, and George Russell. Semantic interrelation of documents via an ontology. In G. Engels and S. Seehusen, editors, *DeLFI 2004, Tagungsband der 2. e-Learning Fachtagung Informatik, 6.-8. September 2004, Paderborn, Germany*, volume P-52 of *Lecture Notes in Informatics*, pages 271–282. Springer-Verlag; D-69121 Heidelberg, Germany; http://www.springer.de, 2004.

[Lange and Kohlhase, 2006] Christoph Lange and Michael Kohlhase. A semantic wiki for mathematical knowledge management. In Max Völkel, Sebastian Schaffert, and Stefan Decker, editors, *Proceedings of the 1st Workshop on Semantic Wikis, European Semantic Web Conference 2006*, Budva, Montenegro, 2006. CEUR Workshop Proceedings. To appear, provisional online version at `http://www.eswc2006.org/technologies/usb/proceedings-workshops/eswc2006-workshop-semantic-wikis.pdf`.

[Mahnke and Krieg-Brückner, 2004] A. Mahnke and B. Krieg-Brückner. Literate ontology development. In Robert Meersman, Zahir Tari, and Angelo Corsaro et al., editors, *On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops*, volume 3292 of *Lecture Notes in Computer Science*, pages 753–757. Springer; Berlin; http://www.springer.de, 2004.

[Matthews, 1997] P. H. Matthews. *The Concise Oxford Dictionary of Linguistics.* Oxford University Press, 1997.

[Mossakowski *et al.*, 2006] T. Mossakowski, S. Autexier, and D. Hutter. Development graphs – proof management for structured specifications. *Journal of Logic and Algebraic Programming*, 67(1-2):114–145, 2006.

[Mossakowski, 2005] Till Mossakowski. *Heterogeneous Specification and the Heterogeneous Tool Set.* Habilitation, Universität Bremen, 2005.

[Nick, 2005] Markus Nick. *Experience Maintenance through Closed-Loop Feedback.* PhD thesis, Technische Universität Kaiserslautern, October 2005.

[ONCE-CS, 2005] Open Network of Centres of Excellence in Complex Systems. Web site at http://complexsystems.lri.fr/Portal/tiki-index.php, 2005.

[Probst *et al.*, 1997] G. Probst, St. Raub, and Kai Romhardt. *Wissen managen*. Gabler Verlag, 4 (2003) edition, 1997.

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

[Verbert and Duval, 2004] 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.