

Managing Variants in Document Content and Narrative Structures

Michael Kohlhase, Achim Mahnke, Christine Müller*

Computer Science, Jacobs University Bremen, Germany

m.kohlhase/c.mueller@jacobs-university.de, achim.mahnke@dfki.de

1 Introduction

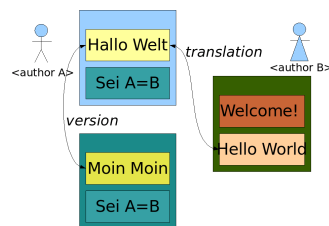
Sharing, reuse, and adaptivity are the key to efficient sustainable development, i.e. continuous long-term usability of document content. They need to be supported by tools and methods taking into account the semantic structure of a document in order to facilitate adaptivity and change management.

A particular issue in reuse and change management is the use of variants to handle consistent variations of documents, e.g. translations into different natural languages. In order to overcome the often found copy-and-paste style of reuse, the management of document variants have to be integrated into markup languages and document processing tools in order to encourage the integrated development of variants. On top of such documents, we can offer improved services like:

- Adaptation of the presentation to the reader's environment, e.g. to the kind of display, medium, or language
- Adaptation to different contexts, e.g. to the intended audience or learner's knowledge
- Checking consistency conditions, e.g. when translating a document, all constituent parts have to be present in the target language

Examples The figure to the right provides an example for the application of language variants: Author *A* has created a document in German.

Author *B*, who is writing about a similar topic in English, would like to reuse *A*'s text fragments in her document. She first selects the respective fragment, *translates* it into English, and finally *relates* her translated document fragments to the original (German) one in author *A*'s document. Once the system knows about this relation, it is able to e.g. assist other authors in the *translation* of their documents. For example, if author *A* decides to translate his document to English, parts of the work have already been accomplished by author *B*. Being



able to relate the German and English variants to each other, the system can help *A* to find and reuse the English fragment in alternative to the German one.

However, the challenge of language variants has already been addressed in other multilingual document management systems. To show more potentials of the variant concept, let us consider changes over time: Eventually, author *A* changes his text and, thus, creates a new *version* of one of his document fragments. This has to be reflected somehow in the document model. By modeling document versions as variants, we can also address the problem of versioning.

Variant Dimensions For the purpose of this document, we will stick to the above example scenario, but want to give an impression on what we see as candidates for document variations which can be managed by our proposed variant module:

Language-oriented In addition to natural language variants, as we have exemplified above, formal language variants such as programming or specification languages can also be suitable variant dimensions: For example, a book on teaching object-oriented programming can either have examples in JAVA and C++ depending on the concrete context it is created in.

Media-oriented The intended output medium has a great influence on how the document is written. While print-outs, e.g. a script for a lecture, are usually self-contained, the variant for reading on an electronic device would contain links to other documents or some material which cannot be printed, e.g. videos or animations.

Audience-oriented Experiences in creating lecture material for eLearning has shown that only parts of the content have to be adapted when preparing courses for different but close enough audiences, e.g. a higher education course in mathematics can essentially be the same document with variant examples tailored to the target audience, thus, the intended audience could be modeled by a variant dimension.

Obviously, this zoo of variant dimensions and relations cannot be handled with the simple methods developed for document management systems, which handle multilingual variants inside documents (or in localization files for programs) and versioning at the

*We would like to thank the members of the KWARC group and the DFKI Bremen for the fruitful discussions and continuous feedback on our work.

file-system level. We propose a general variant management infrastructure as an extension to knowledge repositories. Our approach operates across document boundaries and allows arbitrary groupings of variant objects, yielding an multi-dimensional information resource for complex presentation and variant management systems.

With our work on variants, we are building on ideas developed in the course of the MMiSS project (MultiMedia Instruction in Secure Systems) [MKB04; KBHL⁺03], which annotates document fragments by a set of *variant dimensions*.

To each element of a structured document the MMiSS author can attach a set of attributes stating to which variant domains this element belongs (cf. listing below). All variants of a certain element have the same *id* and a variant selection mechanism is used to choose between these alternative presentations.

```
\begin{assertion}[Label=1, Language=de, LevelOfDetail=low]
  Es sei ...
\end{assertion}
```

However, the MMiSS approach lacks a clear separation of narrative and content structures.

Our work further relates to the approach Gergatoulis et.al. [GSK01] have developed for semistructured data. However, their solution is designed for data structures, not for narrative documents and they propose an extension to XML instead of providing a proper XML application paradigm for variants.

In this paper, we will propose a solution for the management of variants using the OMDOC (Open Mathematical Documents [Koh06]) format as a concrete application paradigm and propose an OMDOC module that extends the format. Note that the methods described in this paper are independent of the mathematical aspects of OMDOC. We will only need a content/narrative document infrastructure here; other content-oriented document markup languages can be extended analogously.

Specifying the Narrative Structure of Documents

The semantic markup language OMDOC1.2. provides two ways to mark up the knowledge contained in a mathematical document and its structure: *Content* OMDOCs are “knowledge-centered documents that contain the knowledge conveyed in a document” [Koh06]. In contrast, *narrative* OMDOCs are used to “reference the knowledge[-centered documents] and add the theoretical and didactic structure of a document”. The *Content* OMDOCs are stored in a knowledge repository, which is called *content commons* according to the terminology of the educational knowledge repository CONNEXIONS [CNX07]. The combination of the narrative structure and the (mathematical) content of a document as the formal representation of a document model, has been defined by Normen Müller as NARCONS, i.e. two-dimensional graphs consisting of a **narrative** layer and a **content** layer [Mül06]. The NARCON approach in OMDOC1.2 has been extended by the specification in [KMM07], which will be the starting point for the variant specification proposed in this paper: Accordingly, documents will be structured by `omdoc` elements which have two optional children to markup the content and structure of

the document: a `narrative` element specifying the narrative structure of the document and a `content` caching its content and, thus, henceforth referred to as the *content cache* of the document.

The listing below presents an the OMDOC representation of author A’s text according to the document model in the [KMM07]: The `narrative` element of the document includes meta information on the document, e.g. the author, as well as references to the content of the document. The `content` of the document caches the text fragments of author A’s document, which are referenced in the narrative structure.

```
<omdoc>
  <narrative>
    <metadata><dc:author>A</dc:author></metadata>
    <ref xref="#t1">
  </narrative>
  <content>
    <omtext xml:id="t1">Hallo Welt</omtext>
  </content>
</omdoc>
```

2 A Variant Module for OMDOC

We introduce two new markup elements named `variant` and `vardim`. The `variant` element is used to express the fact, that an object (specified in its `from` attribute) is a variant of another (specified in the `to` attribute). As we have seen above, there are multiple variant relations, so we need an extensible vocabulary of relation types. Relations are mathematical objects, which can be described in the OMDOC format itself, therefore we represent the type of the relation as a *symbol* in a content dictionary (cf. Listing 1), which can be referenced by the `name`, `cd`, and `cdbase` attributes. The `vardim` element categorizes an object (given in the `for` attribute) in terms of a *variant dimension* (e.g. language, version, format, formalism). Again, we use symbols in content dictionaries for the dimensions (so the `vardim` element carries the attributes `name`, `cd`, and `cdbase`). The `vardim` element represents the value of the object in this dimension, it is specified as a mathematical object. The following RelaxNG [vdV04] grammar summarizes the proposed extensions.

```
sym.att = attribute cd {xsd:NCName},
          attribute name {xsd:NCName},
          attribute cdbase {xsd:anyURI}?
variant = element variant {id. att ?, sym. att,
                          attribute from {xsd:anyURI*},
                          attribute to {xsd:anyURI*}}
vardim = element vardim {id. att ?, sym. att,
                        attribute for {xsd:anyURI*},
                        (math|OMOBJ)}
```

To fortify our intuition, let us re-consider the languages example from the introduction. In OMDOC, A’s text would now have the following form.

```
<omdoc>
  <narrative>
    <metadata><dc:author>A</dc:author></metadata>
    <ref xref="#t1">
  </narrative>
  <content>
    <omtext xml:id="t1">Hallo Welt</omtext>
    <vardim for="#t1" cd="language" name="langdim">
      <math><csymbol cd="language" name="de"/></math>
    </vardim>
  </content>
</omdoc>
```

The content part of the document contains the text itself and the specification that the text has the value “German” in the “language” dimension. Both of these concepts are defined as symbols in the content dictionary in Listing 1 below.

Note that the NARCON in the listing above is just the internal representation in the knowledge repository, or the form that systems will use to communicate to each other, not what the author actually writes. Everything except the string “Hallo Welt” can be generated by the authoring environment.

Listing 1: A Content Dictionary for the Language Variant Dimension

```
<theory name="language">
  <symbol name="translation">
    <metadata>
      <dc:description>
        This variant relation specifies that text
        fragments are translations of each other.
      </dc:description>
    </metadata>
  </symbol>

  <symbol name="langdim">
    <metadata>
      <dc:description>The language dimension.</dc:description>
    </metadata>
  </symbol>

  <symbol name="de">
    <metadata>
      <dc:description>
        This variant dimension specifies that a text
        fragment is written in German
      </dc:description>
    </metadata>
  </symbol>
  ...
  <symbol name="en">...</symbol>
  <symbol name="fr">...</symbol>
</theory>
```

Referencing the same CD, the internal representation of B ' text has the following form.

```
<omdoc>
  <narrative>
    <metadata><dc:author> $B$ </dc:author></metadata>
    <ref xref="#t9">
  </narrative>
  <content>
    <omtext xml:id="t9">Hello World</omtext>
    <vardim for="#t9" cd="language" name="langdim">
      <math><csymbol cd="language" name="en"/></math>
    </vardim>
    <variant from="#t9" to="#t1"
      cd="language" name="translation"/>
  </content>
</omdoc>
```

Here B has to specify that her text is a translation of A 's, which the new OMDOC module represents by the variant element in the content part. Again all other information is supplied by the authoring environment which could also support B in the translation process.

Later, A updates his text (he is from northern Germany). The authoring tool will add the following two lines to the content of his document and changes the href pointer on the ref element to #t2.

```
<omtext xml:id="t2">Moin moin</omtext>
<variant from="#t2" to="#t1" cd="version" name="change"/>
```

Note that both versions of the text are now in the content part, and thus accessible. That #t2 is a newer

version of #t1 is specified by reference to the following content dictionary, which also serves as a background reference and documentation for the version management tool, which deduces the language dimension of the new text to be “German” by the assumption that primary languages are invariant under version changes. In OMDOC, such assumptions can be encoded by axiom elements.

```
<theory name="version">
  <symbol name="change">...</symbol>
  <axiom>
    If  $x$  change  $y$  and  $\text{vardim}(\text{lang}, y) = z$  then  $\text{vardim}(\text{lang}, x) = z$ .
  </axiom>
</theory>
```

3 Abstract Documents

Note that the narrative parts of the documents in our examples above reference concrete objects. In particular, they fix a concrete variant (i.e. a concrete language and version). Following this approach, documents are *extensionally* described, i.e. by their specific parts. Given the concrete text fragments in a knowledge base, an author can write an “abstract” document, which does not fix specific variants. He then leaves the instantiation of the document to a presentation engine that adapts the document to the concrete situation and context of a reader. In this *intensional* approach authors solely specify the dimension of the new document instead of pre-selecting concrete variants.

The extension of an intensionally given document can change over time. For example, if the intention of an abstract document D is to be “as German as possible, else English”, and the text fragments in the content commons are translated step by step, then the extension of D changes from an all-English text to an all-German one via many mixed intermediate stages. While in the extensional approach, authors would have to re-write their concrete document whenever another German text fragment has been translated, authors of intensionally specified documents are relieved from frequently updating the German version of the text.

Of course, the presentation engine could take the document of author A (cf. section 2), follow all variant relations to retrieve alternative objects with the appropriate variant dimension, and assemble them to a new presentation. But it is much more intuitive to introduce an explicit *intensional* level of representation by introducing “abstract objects” that act as placeholders for all the possible variant objects which can instantiate it. In our example above, we could do this by adding an abstract text fragment (which is empty, since we do not have a language-independent representation for text) together with variant specifications.

```
<omtext xml:id="phw"/>
<variant from="#t1" to="#phw" cd="mks" name="concretization"/>
<variant from="#t9" to="#phw" cd="mks" name="concretization"/>
<variant from="#t2" to="#phw" cd="mks" name="concretization"/>
```

Then we can specify an abstract document simply as

```
<omdoc><narrative> <ref xref="#phw"></narrative></omdoc>
```

Note that this approach is general enough to accommodate the analysis of the Mathematical Knowledge Space (MKS) ([KK05]), which suggests to distinguish the *Presentation Objects* found in formatted

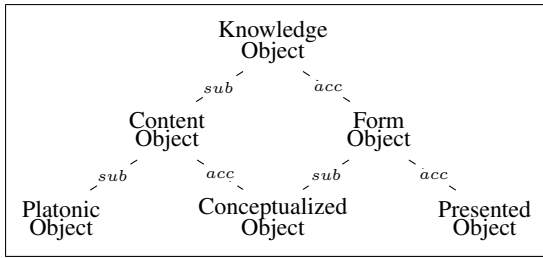


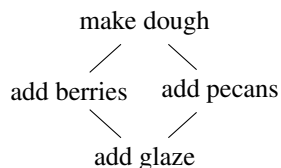
Figure 1: The MKS triangle

documents, from its substance, which is referred to as the *Platonic Object*. While creating his document, the author instantiates the Platonic Object by choosing a suitable representation (conceptualization) and presentation form for his knowledge item and finally creates the concrete Presentation Object. The same knowledge substance can therefore be represented differently (which leads to *Content Objects*) and presented in many ways resulting eventually in a set of objects which we consider as variants of the same Content or Platonic Object (see fig. Figure 1).

4 Narrative Variants

Up to now, we have restricted the variant relation to content objects, where it is supported by the notion of a *content cache*, which allows duplication and reordering.

However, it would be nice to extend the variant infrastructure to narrative structures as well, e.g. to mark up two linearizations of a knowledge structure as equivalent: For example, if two steps in a recipe for Blueberry-Pecan Muffins are independent, the choice of which to take first is arbitrary. In the recipe on the right the two narrative structures “make dough, add berries, add pecans, add glaze” and “make dough, add pecans, add berries, add glaze” are meaningful.



Unfortunately, the narrative part of an OMDOC document is presented directly to the user and does not allow duplicates like the content cache. But on the level of the knowledge base (or the World Wide Web of NARCONS for that matter), all document fragments can be addressed by URIs in the *to* and *from* attributes of variant elements.

Listing 2: Narrative Variants

```

<narrative xml:id="pintro"/>

<narrative xml:id="p22" type="section">
  <ref xref="#example1"/>
  <ref xref="#example2"/>
  <ref xref="#elaboration3"/>
</narrative>

<narrative xml:id="p23" type="section">
  <ref xref="#csintro"/>
  <ref xref="#example2"/>
  <ref xref="#example3"/>
</narrative>
  
```

So, given the three narrative structures in Listing 2 and the variant information in Listing 3 our envisioned presentation engine can adapt an abstract course to the disciplinary background of the audience.

Listing 3: Variant Information

```

<vardim for="#p22" cd="area" name="audience">
  <math><csymbol cd="area" name="math"/></math>
</vardim>
<vardim for="#p23" cd="area" name="audience">
  <math><csymbol cd="area" name="cs"/></math>
</vardim>
<variant from="#p22" to="#pintro" cd="mks" name="???"/>
<variant from="#p23" to="#pintro" cd="mks" name="???"/>
  
```

5 Conclusion

We have presented an infrastructure for representing and managing variants in content-oriented documents: In essence, variants are accumulated in the content- and document commons, and their categories and relationships are specified by two new content elements. While our proposal integrates very well with the OMDOC format, since that provides a notion of content dictionaries and thus extensible vocabularies, there is no reason other content-commons-based formats could not be similarly extended.

The general treatment of variants as first-class citizens renders special constructs e.g. to handle multilinguality obsolete. For instance OMDOC 1.2 took great pains to cater for multiple language variants in the language itself: All top-level elements had multi-language groups of CMP elements (Commented Mathematical Properties) as children. This allowed multilinguality, but left the exact relation between children unspecified (though the specification hinted at a translation relation in places). Moreover, the process of adding new languages to a given element was unclear, unless the translator had access to the original document. In the new proposal, we can do away with CMPs altogether (and have done so in the exposition above). The relation between language variants is made explicit in a content dictionary (other relations like “rough translation” or “paraphrase” could be specified) and extensibility by translators is built into the system. Incidentally OMDOC 1.2 also allowed FMP (Formal Mathematical Properties) elements as siblings with the same “translation” intuition. But the “formalization” relation is an asymmetric relation (we formalize mathematical statement given in natural language in a first-order logic but not vice-versa). This highlights the fact that “translation” is an equivalence relation, which variant-aware knowledge management systems need to take into account. This can be documented in the respective content dictionaries as above.

6 Outlook

Document Adaptation based on variants As shown in Section 3, the *intensional* variant model allows to create *abstract documents*, i.e. documents that solely consists of abstract objects rather than concrete variants. In order to instantiate abstract documents, we need to explicate the variant dimension of the appropriate concrete document, henceforth referred to as the *variant context*. To do so, we introduce a

`vcontext` element in OMDOC, which encapsulates a set of `vardim` elements (cf. Section 2).

The RelaxNG [vdV04] grammar of `vcontext` is given below:

```
vcontext = element vcontext {id. att ?,
    attribute base list {xsd:anyURI*},vardim*}
```

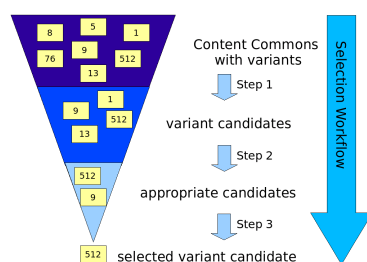
The optional `vcontext` element is a collection of zero or more `vardim` elements, which are used to specify dimensions such as *language* or *audience*. The optional `base` attribute contains a whitespace-separated list of URIs pointing to variant contexts of other NARCONS, hence it can be used to inherit variant contexts. The effective variant context is computed by the variant selection algorithm, whereas the order of the `vardim` elements defines their prioritization, e.g. the first `vardim` has a greater weight than the second one.

The listing below proposes the specification of a `vcontext` in OMDOC: Three `vardim` elements are used to specify the two dimensions language and audience, where two languages (German and English) are given. The order of the `vardim` elements defines their priority: Here, the user is looking for variants for mathematicians which should preferably be German, but if the latter are not available the user would also accept English fragments.

```
<vcontext xml:id="vcontext-id" base="#inherited-vcontext-id">
  <vardim xml:id="d1" cd="area" name="audience">
    <math><csymbol cd="area" name="math"/></math>
  </vardim>
  <vardim xml:id="d2" cd="language" name="langdim">
    <math><csymbol cd="language" name="de"/></math>
  </vardim>
  <vardim xml:id="d3" cd="language" name="langdim">
    <math><csymbol cd="language" name="en"/></math>
  </vardim>
</vcontext>
```

Implementation of the Variant Selection

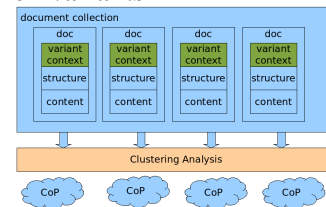
In order to tailor an abstract document to a specific *variant context*, a *variant selection process* is required, which is based on the specifications



of variants by the `variant` and `vardim` elements. The figure above displays the selection workflow: The first step is to retrieve all variant candidates for the document's constituent `omdoc` elements from the content commons. In the second step the appropriate variants are selected based on the variant dimensions given in the `vcontext` element. If multiple variants are available, these are further filtered depending on the priority of the dimensions. Given the respective variants, the concrete document, including a specific narrative structure and a concrete content, can be generated: The references in the narrative structure of the document are bended to the selected variants, which are stored in the `content` cache of the document.

During the selection process various exceptions have to be handled: For example, the variant context can be too restrictive, e.g. by only accepting “German” variants, so that no variant candidate can be found if there are simply no “German” text fragments in the content commons. This could be solved in a naive way by offering no results for the respective object to be displayed, but this is obviously not a preferable solution. Instead, the users are encouraged to provide alternative values for a respective dimension, ordering them according to their preferences, e.g. they can also accept “English” texts although these are less preferred. For example, in the above `vcontext` example, the user wants to see variants which are tailored to mathematicians (first `vardim` element with id “d1”) and which are preferably written in “German”, but can also be “English” if no “German” variant can be retrieved from the content commons.

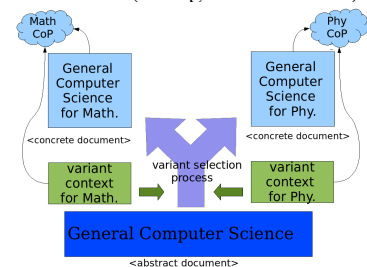
Identifying Communities of Practice (CoPs) based on Variants



By interpreting the selection of concrete variant dimensions and their values as *shared practice*, we want to identify *communities of*

practice [LW91] (CoPs) based on similar preferred variant contexts. Vice versa, we aim at describing CoPs by their most frequently used variant dimensions for various concrete documents (cf. figure on the left).

The modeling of CoPs based on `vcontexts` will, in particular, support the previously described variant selection process:



Instead of requiring a `vcontext` for each user, CoP models allow for reusing existing `vcontext` for users of the same community. For example, let us assume that students of a course are members of one CoP. A teacher could now specify the `vcontext` for two different course, e.g. his General Computer Science (CS) lecture for Mathematicians and the CS lecture for Physicians. Depending on a student's course, the selection process can now reuse the variant information in order to instantiate abstract course material, i.e. generating appropriate (concrete) lecture notes for both communities, i.e. the mathematicians and physicians (cf. figure on the right).

References

- [CNX07] CONNEXIONS. Project homepage at <http://www.cnx.org>, seen February 2007.
- [GSK01] Manolis Gergatsoulis, Yannis Stavarakas, and Dimitris Karteris. Incorporating dimensions in XML and DTD. In *Database and Expert Systems Applications, 12th International Conference, DEXA 2001 Munich, Proceedings*, pages 646–656, 2001.
- [KBHL⁺03] B. Krieg-Brückner, D. Hutter, A. Lindow, C. Lüth, A. Mahnke, E. Melis, P. Meier, A. Poetzsch-Heffter, M. Roggenbach, G. Russell, J.-G. Smaus, and M. Wirsing. Multimedia instruction in safe and secure systems. In M. Wirsing and R. Hennicker D. Pattinson, editors, *Recent Trends in Algebraic Development Techniques*, volume 2755 of *Lecture Notes in Computer Science*, pages 82–117. Springer-Verlag Heidelberg, 2003.
- [KK05] Andrea Kohlhasse and Michael Kohlhasse. An Exploration in the Space of Mathematical Knowledge. In Michael Kohlhasse, editor, *Mathematical Knowledge Management, MKM'05*, number 3863 in LNAI. Springer Verlag, 2005.
- [KMM07] Michael Kohlhasse, Christine Müller, and Normen Müller. Documents with flexible notation contexts as interfaces to mathematical knowledge. In Paul Libbrecht, editor, *Mathematical User Interfaces Workshop 2007*, 2007.
- [Koh06] Michael Kohlhasse. OMDOC – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in LNAI. Springer Verlag, 2006.
- [LW91] Jean Lave and Etienne Wenger. *Situated Learning: Legitimate Peripheral Participation (Learning in Doing: Social, Cognitive and Computational Perspectives S.)*. Cambridge University Press, 1991.
- [MKB04] 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.
- [Mül06] Normen Müller. An Ontology-Driven Management of Change. In *Wissens- und Erfahrungsmanagement LWA (Lernen, Wissensentdeckung und Adaptivität) conference proceedings*, 2006.
- [vdV04] Eric van der Vlist. *RELAXNG: A simple schema language for XML*. O'Reilly, 2nd edition, 2004.