Towards a Flexible Notion of Document Context

Andrea Kohlhase
Jacobs University Bremen
Campus Ring 1
28759 Bremen, Germany
a.kohlhase@jacobs-university.de

Michael Kohlhase
Jacobs University Bremen
Campus Ring 1
28759 Bremen, Germany
m.kohlhase@jacobs-university.de

ABSTRACT

Much of the scientific, technical, engineering, and mathematical knowledge that enables modern society is laid down and communicated in technical documents. Due to their static presentation presentation of the complex issues involved, they remain inaccessible to most readers and pose formidable barriers even for experts. To enable advanced interactions which would support understanding, software systems will have to incorporate machine-understandable (formal) information, while retaining the informal nature of the documents, which allows efficient communication of ideas and methods between humans. The simplistic dichotomy between "formal" (as expressed in a logic) and "informal" (everything else) is not helpful as a guide for designing representation formats for context. As a step towards a remedy we propose the notion of flexibly formal representations (*flexiforms*) based on the analysis of document content and its context in the Software Engineering project SAMSDocs where we elicited a formal context for an informal document collection.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms

Documentation, Theory, Design

Keywords

Flexiforms, Formality, Formalization, Knowledge Mgt.

1. INTRODUCTION

Advances in science, technology, engineering, and mathematics (STEM[1]) are driven by the communication of specialized, often mathematically founded knowledge and its appropriation and application in new situations. It is clear that for mastering this, humans need a deep understanding of the complex issues involved as well as a wide overview over the field(s) and literature. The idea of enlisting computer support in this endeavor is obvious, but has been stymied by the fact that natural language is inaccessibility to machine understanding: computers need syntactic representations of the knowledge to operate on.

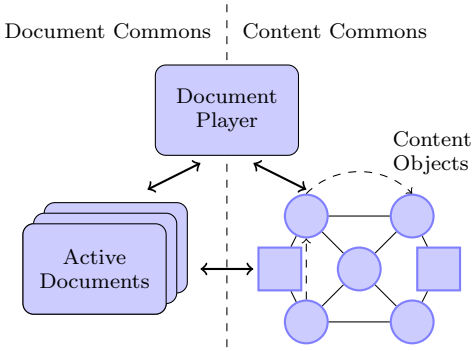


Figure 1: The Active Documents Paradigm

Semantic Interaction with Technical Documents.

An idea that is currently gaining traction in the scientific/technical community is to embed machine-actionable syntactic information into documents. In *semantic document formats* this is provided by making explicit in a formalization process the otherwise implicitly given content and context structure of the documents that convey this knowledge. In *semantic publishing* (see [7] for a recent workshop focusing on this topic), this information is exploited for additional user-level services. An example of this is [14], where text fragments in scientific papers are annotated with references to an ontology of scientific argumentation and experimentation, to help information retrieval. Our **Active Documents Paradigm** (ADP; see Figure 1) goes a step further, it uses formal representations to make the semantically enhanced documents interactive, to embed semantic services, or to validate them. In the ADP, active documents are generated by a **document player** from the **content commons**, a background ontology that organizes content objects (formal representations and document fragments) by their internal structure, their relations amongst each other and to context structures. In the ADP documents become flexible, adaptive interfaces to the domain objects and their contextual inter-relationships in the content commons.

[1]In the following, we treat mathematical documents as paradigmatic representatives for technical documents with STEM content.

SIGDOC'11, October 3–5, 2011, Pisa, Italy.

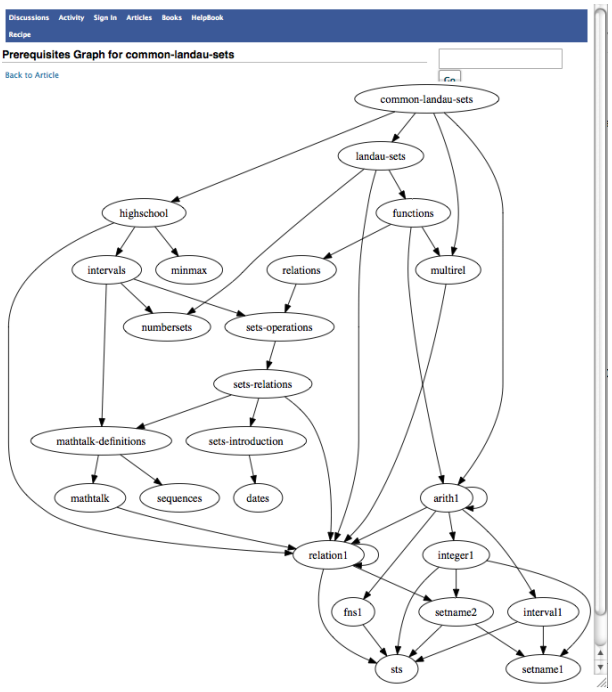


Figure 2: Navigating Prerequisites

The Planetary system [13, 5, 1] is an ADP-based semantic publishing system for scientific/technical documents that uses the OMDoc format [12] for representing the content commons. The context is explicitly represented as a theory graph, which governs visibility and scoping of concepts and and model assumptions. The format also allows to mix formal (i.e., represented as formulae or in a logical system) and informal (i.e., represented in natural language) element in arbitrary nesting levels. These representations can be aggregated and transformed into dynamic HTML+MathML-+RDFa documents, which are presented for interaction by a browser. The theory/context graph can be directly used to visualize the logical context of a document fragment and operationalize it for navigation (see Figure 2 based on [9]). The dependency information in this graph can be further used for aggregating from the content commons targeted documents (guided tours) that introduce the necessary prerequisites (the text in the lower half of Figure 2).

Another semantic service that explicitly makes use of the context induced by the OMDoc formalization is the definition lookup service in Figure 3. Still others include unit conversion, program execution, semantic folding of subformulae and text fragments, formula search, information retrieval, change impact analysis, and project management. All feed on different aspects of the structures, relations, and context links made explicit in the semantic annotations in the content commons. A general observation we make in the active documents paradigm is that we see a variant of the "garbage in – garbage out" principle: the more fine-grained and specific the classifications of semantic objects and their relations become, the more services they afford. This suggests a process of iterative formalization of documents to reach a sweet spot defined by the tradeoff between the envisioned level of semantic interaction and the available (human) resources for semantic annotation.

But we also observe from our work in the Planetary system and the formalization of SAMSDocs described in Section 3.1 that formalization and formality are not as simple as they seem. As the main topic of investigation in this paper we will study in Section 2 the notion of formality and formalization processes with a focus on mathematical knowledge, since mathematics has a long tradition of discussing this, and mathematical models and argumentation play a crucial role in technical documents. In Section 3 we propose two new concepts: flexiformality and flexiformalization to alleviate the conceptual difficulties with absolute formality and formalization identified in Section 2. We defend our new terminology as useful in Section 3.4. The discussion in these two sections is supported by two case studies of technical documents. Finally, in Section 4 we conclude that flexiforms allow to upgrade the practice of "reading a document" into an experience of "communicating with the knowledge the document conveys".

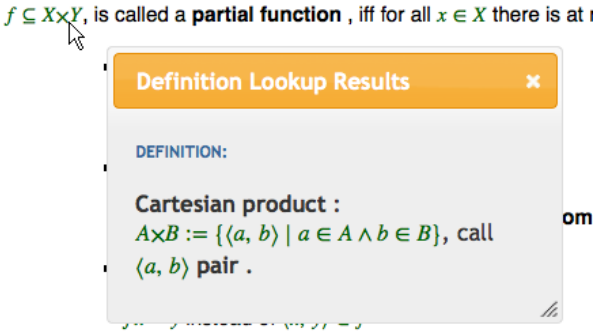


Figure 3: Definition Lookup Service

2. WHAT IS FORMALITY?

We can already see that the notion of formalization and formal representations seem to be much more flexible and less clear than we were told in the introductory logic or AI courses. Those identify the notion of formal representations with logics; i.e., systems that are built on a formal language, a model theory, and a proof calculus. This is a fundamentalist view, as it only allows us to distinguish between "formal" (representations in logics) and "informal" (all other representations), and thus is insufficient for analyzing formalization as an incremental process. Moreover, informality is contagious: A long, complex, and fully formalized proof can be made informal by adding a single informal term like "obviously" or "trivial" — a concept important for the efficient communication of (informal) argumentations that is not supported by logical systems (though one could argue that the integration of fully automated proof procedures into proof checkers is an expression of this in the formal world). Philosophically oriented foundational analyses of mathematical practice like [2] also recognize the level of "*informal rigour*" as a level of efficient communication of mathematics, but avoid a precise definition of what constitutes it.

2.1 Formalization in Mathematics

A lot has been written about (formal) representations of mathematics, much less about languages and tools that support stepwise formalization of unstructured natural language into formal representations, but very little about the *process* of formalization — probably, since formalization is at heart

a cognitive process that seems hard to tackle with the tools of our trade. Social media based approaches strive for collaboration to motivate formalization (e.g. [19]), but have mostly disregarded the need for communicating mathematics therein. But if we relax the problem of understanding formalization a little, then we can make use of the mathematical practice of developing formalization products as a sequence of documents (resulting in a collection $\mathcal{S}$), e.g.

i) an informal proof sketch on a blackboard, and
ii) a high-level run-through of the essentials of a proof in a colloquium talk,
iii) and the speaker's notes that contain all the details that are glossed over in
iv) a fully rigorous proof published in a journal, which may lead to
v) a mechanical verification of the proof in a proof checker.

The motivation for this paper is grounded on taking a document-view *versus* a collection view on formality in [10]. Here, we apply both perspectives to the *process of formalization*. Concretely, we drill down on (in)formality by using the document perspective on a formalization process (see Section 2.2 for a first requirements analysis) yielding a partial ordering relation "more formal than" on documents. In Section 3, particularly Subsect. 3.1, we consider documents as elements in a collection and what that means for their formality within their formalization process.

In our exemplary set $\mathcal{S}$ of documents the process of formalization can be taken as the transformation of a (less formal) document to a more formal one. But note that all but the last documents mentioned above are equally *informal* by the classical definition, which takes formality as "*rigidity of form (and thereby unambiguous precision of a particular logic representation)*" [15, p. 55]. In particular, the notion of "formal" is so confined, that the term "informal" becomes inflated and thus both unpractical. Therefore, a *scientific notion of (in)-formality that captures notions of mathematical rigor in documents* is needed.

The main problems in the conceptualization of formality are that we have to understand the space of reifications of technical/scientific knowledge and at the same time capture the intuitively clear notion of "degrees of formality" in formalization processes (see [17] for an interdisciplinary view). To understand the central issues of a conceptual model for formality and informality, let us look at intra-document issues in a paper on Turing machines.

2.2 The Document View Case Study

Intuitively, we speak of different 'degrees' of formality in mathematical documents. Take for instance this introduction of an accelerated Turing machine[2]:

> An accelerated Turing machine (sometimes called Zeno machine) is a Turing machine that takes 2^{-n} units of time (say seconds) to perform its n-th step. from[3] (1)

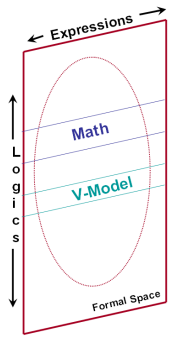
This is a very informal definition, which leaves open many aspects, for instance, which of the many (equivalent) notions of Turing machine is referenced. This situation is common in the high-level introduction of a research paper, which is intended to 'remind' the reader of the definition rather than introducing the concept. In a sense, it can be seen as a reference to a more rigorous definition in the background (e.g. in the original paper this one builds upon). Indeed in our formalization of [3] we made the underlined fragment a typed link (shown as underlined in (1)) to the following definition, which in turn links to similar definitions.

> **Definition 1.3**: An **accelerated Turing machine** is a Turing machine $M = \langle X, \Gamma, S, s_o, \square, \delta\rangle$ working with with a computational time structure $T = \langle \{t_i\}_i, <, +\rangle$ with $T \subseteq \mathbb{Q}_+$ ($\mathbb{Q}_+$ is the set of non-negative rationals) such that $\sum_{i\in\mathbb{N}} t_i < \infty$. (2)

Intuitively, (2) is more "formal" than (1) in two ways: It details the components of an accelerated Turing machine explicitly, and it links to a rigorous definition of a "Turing machine" and a "time structure". If we think of these two formalization actions in terms of formalization in logic, the first one extends the signature (set of concept names), and the second one imports axioms of the referenced definitions. Much to the retroactive surprise of the authors, (2) is not only more formal/specific than (1), but also more general: It allows any sum-bounded step size sequence not just $(2^{-n})_{i\in\mathbb{N}}$. Again, this is a rather typical situation: We realize and take advantage of generalization potential in formalization. To be fully faithful to the original, we would have had to instantiate (2) to a (2′) which imports (2) and specializes the step size sequence. But there is another aspect: A vague introduction like (1) has another purpose, which runs counter the notion of formalization ("the more formal the better"[3]): An underspecified introduction of the concepts signals that the results of the paper apply to "any formalization/concretization", which makes the paper more applicable.

2.3 What is Informal Mathematical Knowledge?

A good starting point for a definition of "formality" and "informality" that is useful for markup techniques is that in the semantic markup process documents are 'intended to be formalized' in some way, so we take the 'meaning' of a document to be the set of its formal representations. But even the space of fully formal reified mathematical knowledge is large and difficult to grasp — it contains all well-formed expressions in all logics, so we conceptualize it as a two-dimensional space $\mathcal{F}$ on the right: Let $\mathcal{S}$ be the set of all logical systems and for any $S \in \mathcal{S}$ let $\mathcal{L}(S)$ be the language of S, i.e., the set of well-formed expressions in S. Now, the space $\mathcal{F}$ of formalization products can be constructed as $\mathcal{F} := \{\langle S, e\rangle \mid S \in \mathcal{S}, e \in \mathcal{L}(L)\}$, and any formal representation is a point in $\mathcal{F}$.



We are deliberately liberal in what we understand as a logic. We include logics that allow a formalization of mathematical

[2]The examples in this section are taken from a case study of marking up the content of a paper on accelerated Turing machines [3] together with the required background knowledge. We conducted this study to evaluate the adequacy of the nascent concept of flexiformalization for typical mathematical documents.

[3]Indeed, in our formalization case study, we were initially motivated by the possibility of formalizing the argument [3] for a logic-based proof checker; see [4] for a precedent and discussion.

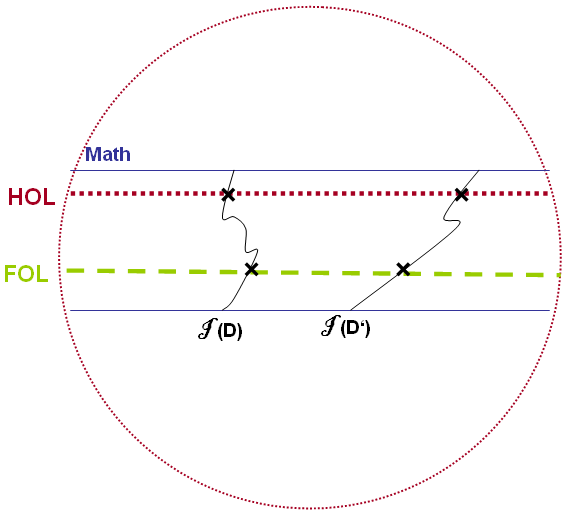


Figure 4: Formal Space Detail

concepts like first-order logic (FOL) or Higher-Order Logic (HOL), but also modal logics or description logics that describe relational structures like the V-Model relations (see Section 3.1) between document fragments. In the Formal Space picture above we have indicated these different kinds of logics as horizontal bands in the space of formalization products.

We will now work towards a conceptual model for the 'meaning of informal mathematics'. For this we need to understand the structure of the space of informal mathematics, which we look at from an abstract point of view first. We consider documents as underspecified representations of formalization products, so for any document D, there is a set $\mathcal{I}(D) \subseteq \mathcal{F}$ it could be formalized as. Note that $\mathcal{I}(D)$ is non-empty, since we postulate documents to be formalizable (in principle) and indeed $\mathcal{I}(D)$ is usually quite large, since even rigid mathematical documents omit many aspects and details of the formalization products.

In particular, mathematical objects (e.g. the definition of an accelerated Turing machine in (1)) can be formalized in different logics, and in a given logic as different expressions — these include different concretizations of the concept as well as logically equivalent formulations of a concretization. In Figure 4 we show a detail view of $\mathcal{F}$, where each document or object D corresponds to a cross-section $\mathcal{I}(D)$ of logical expressions.

In Figure 5 we have depicted the space $\mathcal{F}$ as a plane on the right hand side, and a sequence of documents with their interpretations depicted as cones based in $\mathcal{F}$. We understand this sequence as a *stepwise formalization* process, beginning with a document D. In our example, each successive formalization steps will fix certain formal aspects, restricting the set of possible formalization products further and further. Following this intuition we can define that a document D is **more formal than** D' (write $D' \lll D$), iff $\mathcal{I}(D) \subseteq \mathcal{I}(D')$. This relation on documents and objects is a partial ordering relation (because the subset relation is) and provides an answer to the question of graded formality raised by the case study. Fragments of a document D correspond to subformalization products of $\mathcal{I}(D)$, so we can extend the 'more formal than'-relation to document fragments and the objects of formalization.

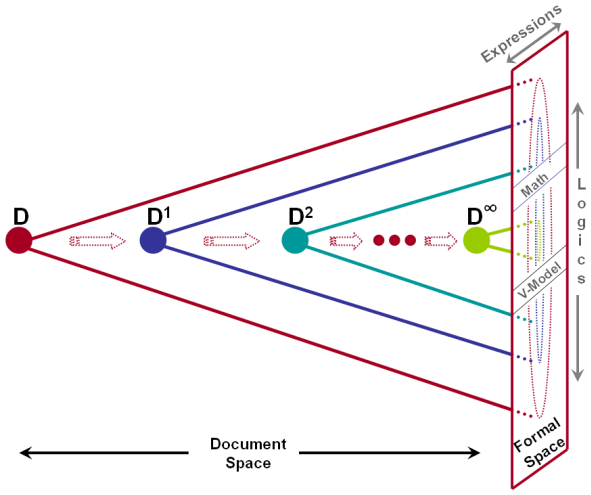


Figure 5: Formality and Informality

3. FLEXIFORMS AND FLEXIFORMALIZATION

A consequence of this notion of "more formal than" must be that the formalization "steps" metaphor implicit in the sequence $\mathcal{S}$ has to be refined: A study of formality structures of a document collection in a Software Engineering scenario showed that "formalization steps" can only be identified within the scope of marked "dimensions of formality" [10] (see Section 3.1).

Intuitively, in contrast to 'degrees of formality' from the document perspective, we assign different 'levels of formality' to documents in a collection, e.g. $\mathcal{S}$ consisting of a proof sketch, a colloquium presentation, the speaker's notes, a published proof, and its verification. Here, the *purpose of formality* varies from document type to document type. For instance, a proof sketch serves insight, whereas a presentation communicates insights. In both document types underspecification is important. In contrast, it is regarded harmful in a published proof and a fatal flaw in input logic for a theorem prover in a verification document. Nevertheless, the objects within such a set are related, even though we cannot use the "more formal than" relationship.

As we already have a case study on the relationships between documents resp. document fragments in a Software Engineering scenario [10], we only report the results with respect to formality here.

3.1 The Collection View Case Study

In [10] we studied the classifications and relationships of "**objects**", i.e., autonomous meaningful text units, within a collection of documents, which were created within the lifecycle of a Software Engineering project. The **V-Model** governed the development process. It resulted in a collection of documents SAMSDocs, which are presented according to the V-Model in Figure 7. Document formats ranged from MS Word over LaTeX to specific theorem prover input documents. Interestingly, many of the inter-document relationships of objects involved the notion of formality but did

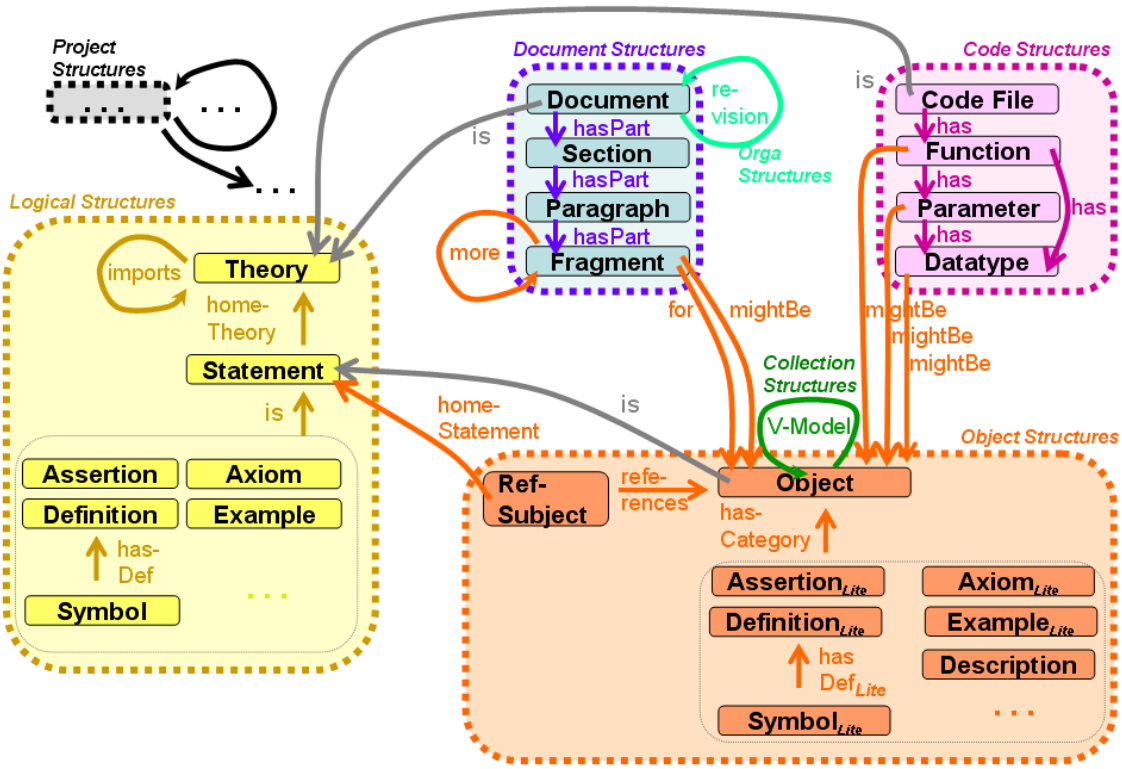


Figure 6: The SAMSDocs Domain Model

not adhere to the sequentiality of $\mathcal{S}$. An object in the contract for example became more formal when implemented and more informal, but rigidly so, when communicated in the manual for customer approval. Another example was some object in the module specification, which became more formal in the implementation, but when in the verification process it was discovered that it needed a slight change, the module specification was updated accordingly.

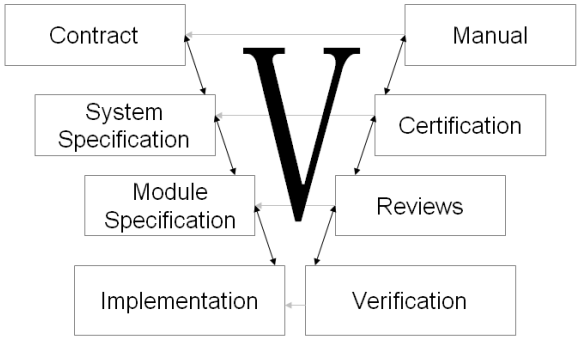


Figure 7: A V-Model for Documents

In more detail, we analyzed distinct formality structures, e.g. the logical structures of a document (like definitions and theorems about them) versus the organizational management of documents or specific document fragments (like a refinement via the V-Model). All these primary and secondary classifications and relationships build a **multi-dimensional space of formality** — if made explicit in the formalization process. In particular, we cannot speak any longer of "the formality of an object", as there might well be more than one at one point in time.

Kirsh notes in [8, p. 276] that "*how agents frame a problem, how they project meaning into a situation, determines the resources they see as relevant to its solution*", which adequately describes the formalization process in this project. Depending on the respective domain ontology in focus, document fragments were **spotted** (coined as - classified - objects or references to specified other objects), **chunked** (connected to structural items like assigned to theories) and **related** (i.e., relationships between objects or structures were made explicit). Note that the 'formality' depended on the concerned dimension of the domain ontology and progress of formalization was always done with respect to one of those dimensions.

Figure 6 shows an extract of the elicited collection domain model of SAMSDocs. It consists of several independent domain ontologies and document models, every component has its own characteristics. The most formal one consists of the *logical structures* (here, the OMDoc ontology), the most informal one of the *object structures*. Note that the informal structures can copy formal ones, but their objects and relationships are not as rigid as in the logical structures, thus they can be considered as "lite" elements. This is important in the formalization process where we often have a time gap between the spotting of an object and its assignment to a chunk. But in the ontology this property has to be existent right away, otherwise the document is invalid and cannot be parsed. Some structures depend on the document type. For example, a software file might pro-

vide functions with parameters (*code structures*), whereas a Word document is hierarchically structured into sections which in turn consist of paragraphs and so on (*document structures*). In the SAMSDocs collection we also found *organizational structures*, in particular revision descriptions within each document. Some reoccurring *project structures* like project-specific definition tables were identified and supported. Additionally, the project development process was mirrored in the objects spotted. In particular, the semantic objects were developed over the creation of the document collection, so a link of one object instance in two distinct documents could be described by the V-Model. Such relationships we called *collection structures*. The objects of all the distinct structures could be connected via informal "inter"-links. The linearity of the document structures e.g. made it necessary to be able to concatenate document fragments (with different locations) to build a complete object element.

3.2 Flexiforms

We have shown that documents sensibly are formal (supporting syntax-driven reasoning processes) and informal (appealing to a human reader) at the same time, that is they are of *flexible formality*. Moreover, they can be more or less formal in each dimension of the formality space at the same time. In order to be able to express this quality, we introduce the adjective "**flexiform**" to describe the fact that a representation is of flexible formality in any of the adequate dimensions of formality.

A good example for a flexiform document is a mathematical text, which contains informal representations — e.g. historical remarks or proof sketches — as well as formal definitions *and* version management information. We understand the term "flexiform" in an inclusive sense. In particular, we include fully formal representations like algebraic specifications of software properties and in principle also fully informal documents like e-mails into the set of flexiform documents. See [16] for an exploration into the structure of documents we now call flexiform documents. We are also interested in

- **flexiform theories**, i.e., mathematical theories that are represented with flexible formality (within the mathematical formality dimension). Flexiform theories tend to be formal objects that make use of informal representations for documentation.
- **flexiform digital libraries**, i.e., collections of flexiform documents whose relations may be marked up with flexible degrees of formality, ranging from the 'source' relation pointing from a derived to an original resource, as defined by the Dublin Core metadata vocabulary, to theory morphisms.
- **flexiform fragments**, i.e., self-contained fragments that make up the flexiform objects or collections above. Good examples for small-scale flexiform fragments are given by parallel markup of formulae in MathML, where an (informal) representation in presentation MathML and a (formal) one in content MathML are combined in a joint fragment and interlinked by cross-references that mark up corresponding subformulae.

We will use the noun "**flexiform**"[4] to denote an arbitrary flexiform object, fragment, or collection as exemplified above. This concept of the class of flexiforms is useful, since it has very good closure properties: Flexiform fragments can be composed to flexiform documents, which can be collected to flexiform libraries, which in turn can be formalized to flexiform theory graphs or excerpted to flexiform documents. In particular all of the knowledge management processes mentioned above can now be described in terms of flexiforms.

The class of flexiforms, as defined here, is very broad; it includes arbitrary (informal) documents, datasets, and logical axiomatizations. We restrict the set of completely informal representations to those that are intended to or could in principle be formalized, excluding e.g. poetry, which are outside our interest in this paper. The class of flexiforms particularly includes all mathematical documents; indeed, since the foundational crisis of mathematics, mathematicians contend that all mathematics can in principle be formalized e.g. the ZFC set theory, even though this is almost never executed in practice. Note furthermore, that the question of mere formalizability is quite independent of 'truth' or provability in mathematics, as even false conjectures can be formalized and — given sufficiently liberal logics — even 'faulty' proofs. Concretely, the class of flexiforms includes specifications from program verification, semantically annotated course materials, textbooks in the "hard sciences", etc.

3.3 Flexiformalization

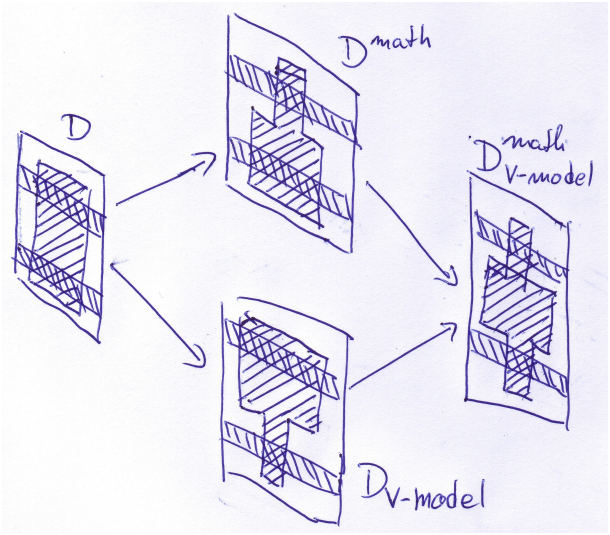


Figure 8: Document Diamond

The most salient aspect diagnosed in [10] is that real world collections contain a multi-dimensional space of classifications and relations. Of course our flexiformalization model must be able to account for this. In Figure 8 we have sketched the situation. A flexible formalization process begins with a document D, which is informal with respect to its mathematical and V-Model aspects (we depicted this by having a large intersection of $\mathcal{I}(D)$ with the logics that talk about math and V-Model aspects, given by the shaded bars, see also Figure 5). Now, D can be independently formalized into D^{math} and $D_{\text{V-Model}}$ (reducing the intersection of

[4]The term "flexiform" is also used independently by DMR Limited and InterDev Pty Ltd, but in both cases it describes unrelated technical concepts.

$\mathcal{I}(D)$ with the math and V-Model bars respectively). But as the mathematical and V-Model relations do not interact, we make the diagram in Figure 8 commute as a diamond by going to $D^{\text{math}}_{\text{V-Model}}$.

We have seen in Section 2.3 that the term formalization is as difficult to get as the term formalization product even applied to an object. The latter is now defined to be a flexiform fragment, but can we supply a helpful replacement for "formalization"? Shipman and McCall suggested "*incremental formalization, in which, first, users express information informally and then the system aids them in formalizing it*" [18, p. 199], which is based on having a notion of progressive formalization. The document view of formality in section 2.3 lead us to the definition of a partial ordering of documents. So the term "incremental formalization" can be used, but has to be specialized to take into account the multi-dimensionality discussed in the collection view section.

Whenever we can find such a relation among flexiforms we can speak of a transformation process from one into another, which we call **flexiformalization**. As a side note we observe that the pair of terms flexiformalization/flexiform behaves differently than the pair formalization/formalization product discussed above: mathematical documents cannot be formalization products because the formalization process is almost never completed (formalization products usually only cover part of the knowledge in a document). It is one of the advantages of our new conceptualization that flexiformalization of flexiforms to more formal flexiforms does not have to be complete to be meaningful.

A semantic markup process of a collection can be viewed as flexiformalization as the informality of marked-up objects changes in the process. Moreover, authoring documents within a collection can be seen as flexiformalization as well if the collection structure is (naturally) explicated in the process. For instance, we can study SAMSDocs as a flexiformalization result since the objects/documents are connected within the collection via the V-Model. In [11] we made a point that authoring is different from formalizing. Flexiformalization dissolves this contrast too.

3.4 Yet another New Buzzword?

The introduction of new terminology has to be undertaken with care. Buzz words for selling science to the public or colleagues are dangerous as they devaluate publishing as market strategy and moreover, they prevent readers from realizing new from hot ideas. We believe that "flexiformalization" is a new idea, which will not only help to digitalize more knowledge resources, but also will allow to add more, user-centered services. In this section we try to convince you too.

In a nutshell, we have argued in Section 2 that the term "formalization" interpreted as formalization product describes an end product of a *complete* formalization process, but as such is not flexible enough to support the "more formal than" order on documents/document fragments. In Section 3 we looked at the extension of this order to a total order on the multi-dimensional formality space. Here, the term "formalization" interpreted as formalization process is flawed, as "more formal than" is only adequate in one formal dimension. In contrast, the notion of "flexiformalization" backs up the "more formal than" relation within a document in any formal dimension. In the following we like to cue you in to the advantages of the new terminology.

Precision: The formalizing process builds on multiple formality levels and dimensions, therefore we oversimplify the process description when we talk about "formalization" because it is de facto a "*flexi*formalization". With precision come all the well-known advantages of deep understanding.

New Services: Note for example that with understanding the formalizing task as "flexiformalization" comes the opportunity for innovative support services. For instance, formalizing different dimensions conceptually require different people/demons. Studying the intention of formality in terms of flexiforms may lead to a new appreciation of formality per se as the efficiency of formalizing, particularly over- or sub-formalization with respect to a certain task, can be discussed much more precisely.

New Knowledge Documents: Moreover, if we understand documents as flexiforms, then more documents come into focus for knowledge management, that is: We gain knowledge sources.

Dimensions of Formality: In the course of the SAMSDocs project we frequently experienced that people were curious to know the progress of formalization. This was inherently hard or even impossible to answer as 'the' formalization was not yet 'formal'. If it were recognized that there are distinct dimensions of formality as analyzed in [10, sec. 2], then progress of formalization could be appropriately evaluated. In other words, if formalization were recognized as flexiformalization, then communication and appreciation is alleviated.

Levels of Formality: In [11] we differentiated between spotting, chunking and inter-relating to describe the development within the formalizing process. If we understand documents as flexiforms, then we acknowledge the impreciseness of formalization or the informality of knowledge dissemination. This is important, because every level of formality can be exploited by machines, what is necessary though is that it *is* explicated.

Degrees of Formality Some SAMS project members expected a 'full' semantic preloading of SAMSDocs, whereas others wanted to be pragmatic about formality depth, i.e., only to the point where pre-determined management services were enabled. Here, implicit "degrees of formality" were addressed without being able to express it explicitly. On the one hand the flexiform model gives us a notion of "more formal than" among the stages of a flexiformalization process. On the other hand the relation $\lll$ is not a total ordering (because $\subseteq$ isn't), so a critical aspect of intuitive "degrees of formality" understood across unrelated documents is still not captured.

4. CONCLUSION

In this paper we have tried to understand a crucial, foundational aspect of semantic publishing, an emerging paradigm for making scientific/technical documents interactive and thus upgrading the practice of "reading a document" into an experience of "communicating with the knowledge

the document conveys". As interactivity requires machine-actionable representations, preparing documents for semantic publishing, requires a process of reifying technical, particularly mathematical, knowledge into representations that are amenable to knowledge management technologies. We have analyzed this formalization process from a perspective of authors' practices (what do authors do when they reify knowledge: They write documents in collections) and from a theoretical perspective (what is the meaning of such document collections: Formal representations). We have found "document- and collection-inherent flexibility of formality" as well as "application-induced flexibility", therefore we propose the notion of flexiforms for reifications and flexiformalization for the process of reification of knowledge.

We contend that this new conceptualization contributes to the understanding of semiformal knowledge repositories, as far as they are created by (flexi)-formalizing informal ones. Our practical experiences with the various (flexi)-formalization projects cited above has shown that our OMDoc format can be used for all flexiformalization products and thus for all stages of the formalization process. We have just not seen it as a flexiformalization format in the past.

Also, there is more conceptual work to be undertaken, we have looked at the formalization process in this paper, but not at the dual "verbalization process", which generates less formal representations from more formal ones in the new context of flexiforms. Such processes have been studied in general under the heading of "natural language generation" in computational linguistics and specialized e.g. for the mathematical domain as "proof presentation" for formal, machine-found proofs (see [6]). We conjecture that our notion of flexiforms may add some clarity to the discussion there.

Acknowledgements. We gratefully acknowledge that the ideas presented in this paper have been shaped by intensive discussions with Cris Calude working on the Turing Machine case study. We are grateful for the support and the stimulating environment he provided in the six months the authors spent on sabbatical at the University of Auckland. An equally important influence was the collaboration with Christoph Lange with whom we discussed the influence of multi-dimensional metadata, needs and services in our SAMSDocs case study. Finally, we also want to thank the reviewers for their helpful comments and pointers.

5. REFERENCES

[1] Planetary developer forum. http://trac.mathweb.org/planetary/.

[2] H. Barendregt and A. M. Cohen. Electronic communication of mathematics and the interaction of computer algebra systems and proof assistants. *Journal of Symbolic Computation*, 32:3–22, 2001.

[3] C. Calude and L. Staiger. A note on accelerated turing machines. CDMTCS Research Report 350, Centre for Discrete Mathematics and Theoretical Computer Science, Auckland University, 2009.

[4] C. S. Calude and S. Marcus. Mathematical proofs at a crossroad? In J. Karhumäki, H. Maurer, G. Paun, and G. Rozenberg, editors, *Theory Is Forever*, LNCS, pages 15–28. Springer-Verlag, Berlin, 2004.

[5] C. David, D. Ginev, M. Kohlhase, B. Matican, and S. Mirea. A framework for modular semantic publishing with separate compilation and dynamic linking. In García Castro et al. [7].

[6] U. Egly, A. Fiedler, H. Horacek, and S. Schmitt, editors. *Proceedings of the Workshop on Proof Transformation, Proof Presentations and Complexity of Proofs (PTP-01)*. Universitá degli studi di Siena, 2001.

[7] A. García Castro, C. Lange, E. Sandhaus, and A. de Waard, editors. *Proceedings of the 1st Workshop on Semantic Publication, Extended Semantic Web Conference*, number 721 in CEUR Workshop Proceedings, Aachen, 2011.

[8] D. Kirsh. Problem solving and situated cognition. In P. robbins and M. Aydede, editors, *Handbook of Situated Cognition*, pages 264–306. 2009.

[9] A. Kohlhase and M. Kohlhase. Semantic transparency in user assistance systems. In B. Mehlenbacher, A. Protopsaltis, A. Williams, and S. Slatterey, editors, *Proceedings of the 27th annual ACM international conference on Design of communication (SIGDOC)*, pages 89–96, New York, NY, USA, 2009. ACM Press.

[10] A. Kohlhase, M. Kohlhase, and C. Lange. Dimensions of formality: A case study for MKM in software engineering. In S. Autexier, J. Calmet, D. Delahaye, P. D. F. Ion, L. Rideau, R. Rioboo, and A. P. Sexton, editors, *Intelligent Computer Mathematics*, number 6167 in LNAI, pages 355–369. Springer Verlag, 2010. `http://arxiv.org/abs/1004.5071`.

[11] A. Kohlhase, M. Kohlhase, and C. Lange. sTeX – a system for flexible formalization of linked data. In A. Paschke, N. Henze, T. Pellegrini, and H. Weigand, editors, *Proceedings of (I-Semantics)*. ACM, 2010.

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

[13] M. Kohlhase, J. Corneli, C. David, D. Ginev, C. Jucovschi, A. Kohlhase, C. Lange, B. Matican, S. Mirea, and V. Zholudev. The planetary system: Web 3.0 & active documents for stem. *Procedia Computer Science*, 4:598–607, 2011.

[14] C. H. Marcondes. A semantic model for scholarly electronic publishing. In García Castro et al. [7].

[15] D. E. Millard, N. M. Gibbins, D. T. Michaelides, and M. J. Weal. Mind the semantic gap. In *HYPERTEXT '05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, pages 54–62, New York, NY, USA, 2005. ACM.

[16] C. Müller. *Adaptation of Mathematical Documents*. PhD thesis, Jacobs University Bremen, 2010.

[17] L. Radford. The seen, the spoken and the written: a semiotic approach to the problem of objectification of mathematical knowledge[1]. *For the Learning of Mathematics*, 22:14–23, 2002.

[18] F. M. Shipman III and R. J. McCall. Incremental formalization with the hyper-object substrate. *ACM Trans. Inf. Syst.*, 17(2):199–227, 1999.

[19] F. Sousa, M. Aparicio, and C. J. Costa. Organizational wiki as a knowledge management tool. In *Proceedings of the 28th ACM International Conference on Design of Communication*, SIGDOC '10, pages 33–39, New York, NY, USA, 2010. ACM Press.