Spreadsheet Interaction with Frames: Exploring a Mathematical Practice

Andrea Kohlhase[1] and Michael Kohlhase[2]

[1] German Center for Artificial Intelligence (DFKI)
Andrea.Kohlhase@dfki.de
[2] Computer Science, Jacobs University Bremen
m.kohlhase@jacobs-university.de

Abstract. Since Mathematics really is about what mathematicians do, in this paper, we will look at the mathematical practice of *framing*, in which an object of interest is viewed in terms of well-understood mathematical structures. The new perspective not only allows to deepen the understanding of a resp. object, it also facilitates new insights. We propose a model for framing in the context of theory graphs, and show how framing can be exploited to enhance the interaction with MKM systems. We use the framing extension of our SACHS system — a semantic help system for MS Excel — as a concrete example.

1 Introduction

It has often been said that to understand mathematics one has to understand *what mathematicians do*, and in fact the value of a mathematical education is usually appraised more for the practices and abstract skills acquired with it, than for the concrete knowledge gained. For the field of Mathematical Knowledge Management (MKM) this suggests that we have to support mathematical practices in our systems and representation formats (see [KK06] for a call to arms to do just this).

A particular mathematical practice that comes to mind is to view objects of interest in terms of already understood structures and make creative use of this new perspective. For instance, we can understand certain point sets in three-dimensional space by viewing them as zeroes of polynomials. Then we may derive insights about these point sets by studying the algebraic properties of polynomials. For the purposes of this paper we will say that we are **framing** the point sets as algebraic varieties (sets of zeroes of polynomials). Other intuitive examples of framing in mathematics consist e.g. in equipping a differentiable manifold with a (differentiable) group operation (arriving at a Lie group), or interpreting a Boolean algebra as a field of sets via Stone's representation theorem. The practice of framing is so valuable, since it allows to transport insights between seemingly disparate fields. Indeed, in mathematics many of the most famous theorems earn their recognition *because* they establish profitable framings.

We adopt the term 'framing' for the mathematical practice we want to study because we want to highlight the particular approach to context mathematicians choose. In contrast to many MKM applications where 'to contextualize' means to manipulate appearances (presentations), we are interested here in the potential of manipulating substances (representations). We do not want to use the term 'view' since that is already taken in MKM and fails to address either the cognitive process aspect or its collaborative aspect. The term '**frame**' has been used e.g. in Communication Research as "*schemata of interpretation that enable individuals to locate, perceive, identify and label occurrences within their life space and the world at large.*" [SRWB86]; a frame is understood as a *scaffolding of concepts that influence the understanding of situations*. Therefore it seems to sit well with our demands.

In this paper, we will argue that framing in mathematics usually involves some kind of mapping or even isomorphism between the participating structures. We will propose a model for the mathematical practice of framing in the context of theory graphs, and we will show how framing can be exploited in the interaction design with MKM systems using our extension of the SACHS system — a semantic help system for MS Excel —- as an example.

2 Modelling the Practice of Framing

We will set the mathematical practice of framing in the context of theory graphs following the "little theories approach" proposed in [FGT92], in which separate mathematical contexts are represented by separate theories. Structural relationships between contexts are represented as theory morphisms, which serve as conduits for passing information (e.g., axioms, definitions, and theorems) between theories (see [Far00]).

2.1 Semi-formal Theory Graphs and Framing

Theory graphs are one of the theoretical underpinnings of what is sometimes called **Formal Digital Libraries (FDL)**, which have been a focus of the MKM community. FDL have evolved from the libraries of theorem proving and verification systems, and the theory graph structure is used there for modularization by compartmentalizing knowledge about objects into modules (theories) and linking them by inheritance links (morphisms). This aspect seems to be an appealing starting point for modelling framing. But FDL are of rather limited use for mathematicians as most mathematics is not born formal. Indeed, formalization is a very specialized framing practice, which is more often than not at the very end of mathematical creative processes. Therefore, for our purposes we draw on **Semi-Formal Digital Libraries (SFDL)**, where axioms, definitions, theorems, and even theories can be given as annotated text fragments. As semi-formal representation formats like MathML, OpenMath, LaTeX, XHTML+MathML, MathLang [KWZ08], MathDox [CCB06] concentrate on mathematical formulae only or lack theory-level features, we will use our OMDoc format [Koh06], which generalizes the structural invariants of theory graphs to an informal level [RK08], but also accommodates fully formal representations. In the following, we will assume an OMDoc-based background SFDL with a fine-grained theory graph structure which acts as a content commons that contains our examples as theory subgraphs.

We will use the formal techniques and results about modular theory graphs from [MAH01,RK09] in an informal setting without checking the various category-theoretic prerequisites. This is generally justifiable by current practice in mathematics (see [BC00] for an extensive discussion): Arguments are presented informally and are considered *rigorous*, if they could in principle be elaborated into a formal system like first-order logic with set theory axioms which does meet the formal prerequisites. Even though such an elaboration is almost never done in practice, enough examples have been carried out that we can be confident that it is possible in all informal but rigorous arguments (in this paper). Thus we use the informal theory graph representation of the OMDoc format [Koh06], which provides an infrastructure for theory morphisms and inter-theory reasoning without requiring formality.

Let us now briefly recap the salient features of semi-formal theory graphs to make the paper self-contained. A **theory** consists of a **signature** — i.e. a set of concepts or symbols — together with a set of **axioms** — i.e. distinguished members of the set of sentences induced by the signature — which act as basic assumptions of the theory. A signature mapping is called a **theory morphism**, iff all axioms of the source theory are consequences of the target theory's axioms. Thus we can use theory morphisms for the modularization of mathematics: Consider the diagram on the right where we have depicted theories as boxes consisting of the theory name, signature, axioms and theorem morphisms as arrows labeled with i:φ, where i is a name and φ a signature morphism. In our example we have a theory of monoids called Monoid (i.e. structures $\langle G, \circ\rangle$, where G is a set and $\circ: G \times G \to G$ an associative binary operation on G, such that there is an element e with $a \circ e = a$ and $e \circ a = a$). To extend this to a theory of commutative groups, we only have to add axioms for the existence of inverses and commutativity to the monoid axioms. So in a theory graph we only have to represent these *local* axioms and *import* the ones from Monoid. Note that the identity signature morphism induced by the import becomes a theory morphism by fiat. But we can do even more: to define the theory of rings called Ring, we can just import the Monoid axioms into the Ring theory via a signature morphism $\sigma := \{G \mapsto R^*, \circ \mapsto *, e \mapsto 1\}$ and the Comm_Group axioms via $\tau := \{G \mapsto R, \circ \mapsto +, e \mapsto 0, \cdot^{-1} \mapsto -\}$ and add the distributivity axiom.

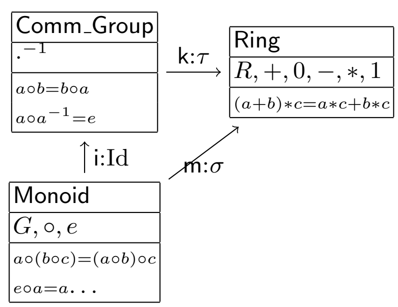


Fig. 1. A Theory Graph for Rings

The distinguishing property of theory morphisms is that they preserve theorems, i.e. after translation, all theorems from the source theory are theorems of the target theory. This is trivial for the definitional morphisms we have seen above, but also holds for **views**: theory morphisms, where we prove all the **proof obligations** (i.e. the translated axioms of the source) in the target theory. The representation theorems alluded to above give rise to views in this sense. We will need one more notion below: we will call a theory morphism $\sigma: S \to T$ **conservative**, iff $\sigma(s)$ is a theorem of T, iff s is one of S, i.e. the target theory does not introduce new knowledge about objects that can be expressed in terms of the source theory. Note that adding axioms to the target theory will usually render a theory morphism non-conservative; an exception are definitions like $G^* := G \backslash \{e\}$ in Monoid. The significance of conservative theory morphisms is that any theorem that can be expressed only in terms of the source language can be transported back to the source theory.

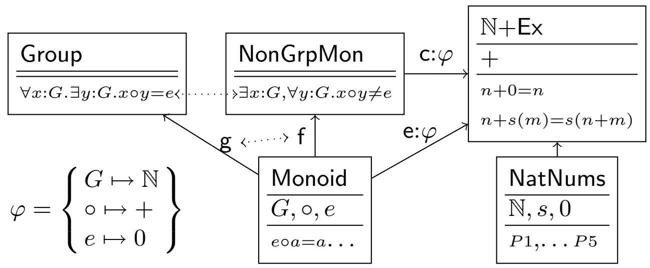
Concretely, we *model the framing practice* by defining a **framing** to be the establishment (creating or choosing) of a theory morphism from a source theory (the **framing theory**) into the theory representing the problem (the **framed theory**). The theory morphism itself is called a **frame**. In situations where there is a unique morphism from a theory S to T, we will also say that S is a frame for T in a slight abuse of terminology. But note that in many situations we naturally have more than one morphism between two theories, for instance above we have the morphisms m and $\mathsf{k}\circ\mathsf{i}$ (theory morphisms compose naturally to theory morphisms). Mathematically, m frames Ring in terms of its multiplicative monoid structure and $\mathsf{k} \circ \mathsf{i}$ as the additive one. Note that for every theory S, the identity is a theory morphism, we call it the **natural frame** for S. Finally, we will say that frames $\mathsf{f_i}: S \to T_i$ are **frame variants**, iff the T_i are pairwise inconsistent. In most practical cases the theories T_i add a single axiom each, e.g. specializing a parameter that was left unspecified in S in different ways. We will call these axioms the **loci** of the variants. We assume that frame variant relations (and their loci) are explicitly annotated in SFDL metadata; see [KMM07] for a proposal to integrate such data into the OMDoc format.

To strengthen our intuition about framing and the suggested model, we will have a closer look at three typical framing practices used in mathematics. From them we will draw more general conclusions concerning SFDL formalization and requirements for the interaction with frames.

2.2 Understanding Abstract Objects by Examples

A fine example of framing is the mathematical practice of supplying examples for abstract concepts. For instance most expositions of the concept of a monoid will give the natural numbers with addition as an example, and use it as a "near-miss" counterexample for being a group.

In [Koh06, section 15.4] we had argued that examples are triples $\langle o, P, A\rangle$, where o is a mathematical object ($\langle \mathbb{N}, +\rangle$ in our example), P is a property (being a monoid), and A is an assertion establishing $P(o)$. Re-interpreting examples as theory morphisms allows to package the same information much more plausibly. Consider the following theory graph fragment, where the theory $\mathbb{N}+\mathsf{Ex}$ builds on the natural numbers (specified e.g. by the Peano Axioms) and is connected by



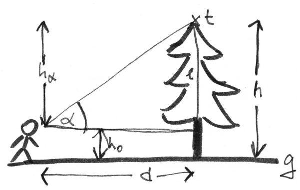
$$\varphi = \left\{ \begin{array}{l} G \mapsto \mathbb{N} \\ \circ \mapsto + \\ e \mapsto 0 \end{array} \right\}$$

a view e to Monoid. Note that e carries with it a set of proof obligations, which together state the fact that the structure $\langle\mathbb{N},+\rangle$ is a monoid. To make $\mathbb{N}$+Ex into a counterexample for a group (not all natural numbers have additive inverses) we introduce a theory NonGrpMon of non-group monoids with a local axiom that states that there is an element x for which no y is an inverse (note that this is just the negation of the group axiom). Then any framing c for $\mathbb{N}$+Ex naturally acts as a counterexample to the assumption that it is a group, since the local axioms of Group and NonGrpMon are contradictory. In our terminology, the frames f and g are frame variants and the local axioms are the variant loci — we show this by the dotted bidirectional arrow.

2.3 Problem Solving

Another example of framing arises in word problems, i.e. mathematical problems clothed in words. Problems like the following one appear in many high school textbook on elementary trigonometry.

> ***Problem 0.8.15:*** *How can you measure the height of a tree you cannot climb, when you only have a protactor and a tape measure at hand.*



The standard solution is to assume that the tree in question stands on flat ground, to mark the tree at eye height and to use the protractor for sighting the top of the tree and the mark to determine the angle α between the sightings. The tape measure can be used to determine the eye height (h_0) and the distance d between sighting point and the center of the tree. Then the height h of the tree is $h = h_0 + h_\alpha = h_0 + d\tan(\alpha)$ according to the sketch on the right.

Even in this simple situation, framing is complex; consider what happens in the solution process. The first step is to realize that certain concrete properties of the problem do not matter, in this case the shape of the tree, its color, and indeed that it is a tree at all; so in a first framing step, we map the problem to a simpler one of determining the length of a mathematical line segment without directly measuring it. The second step in solving the problem is to carefully add further objects to the problem (e.g. the mark and the sighting point) so that a solution can be found. And in a third step, the solution is mapped back to the original problem and verified there.

In our example, we would posit a theory graph like the one on the right, grounded in a theory **"Planar Geometry (PG)"**, which supplies knowledge about right triangles, angles, and trigonometry. On this, we build a theory **“Planar Geometry for our Problem (PGP)”** that abstracts from all biological details of trees (they come from a Forestry theory) and only extends PG with two perpendicular line segments l and g and a point p at the end of l. This will be framing theory for our problem; the framing is given by the theory morphism p:φ, where φ maps p to the tree top, l to the center of the tree's trunk, and g to the ground the tree stands on. Since our problem also inherits from the theory Forestry that contains assumptions like the one that the trunks of (fir) trees are straight and grow vertically, p is a view and thus constitutes a frame in our model. Note that all the theory morphisms in the graph up to now are conservative, so by [MAH01, Proposition 12] we can extend it by the union theory **“Solution (SOL)”** and the two dashed theory morphisms p′ and q′, which are again conservative. SOL contains the full information to understand the solution. As q′ is conservative over Problem, the computed height is the correct one for the problem.

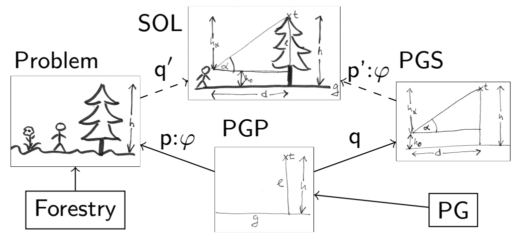


Fig. 2. Framing and Extending a Problem

2.4 Problem Transformation

In the third example, we study the contribution of framing to understanding and anchoring of mathematical structures using the well-studied **“Mutilated Checker Board Problem (MCBP)”** (see [Win01,KP06] and references there). The MCBP is based on a combinatorial problem, which we can formalize as a pair covering problem which (following a formulation of McCarthy [Win01]), we can pose as follows. Given a set S and a relation D on S, then we call a relation $R \subseteq D$ a **partial covering**, iff the pairs in R are pairwise disjoint, and a **covering of** S, iff the union of all pairs in R is S. Now the **“Pair Covering Problem (PCP)”** is to find a covering R for a given set S and relation D or to show that no covering exists. We are going to look at two special PCP.

In the **“Adjacent Fields Covering Problem (AFCP)”**, $S \subseteq \mathbb{N} \times \mathbb{N}$, and $\langle\langle i,j\rangle,\langle k,l\rangle\rangle \in D$, iff $|i-k|+|j-l| = 1$. In the **“Disjoint Set Covering Problem (DSCP)”**, S is the disjoint union of sets B and W and $D = B \times W$. In the MCBP S is a mutilated checker board (the squares of the board minus the black ones in the corners) and D is the adjacency relation. Finally, the **“Matchmaker Problem (MMP)”** is given as follows in [Sch09].

> *In a small but very proper Russian village, there were 32 bachelors and 32 unmarried women. Through tireless efforts, the village matchmaker succeeded in arranging 32 highly satisfactory marriages. The village was proud and happy. Then one drunken Saturday night, two bachelors in a test of strength, stuffed each other with pirogies and died. Can the matchmaker, through some quick arrangements come up with 31 satisfactory marriages among the 62 survivors?*

Obviously, the DSCP and AFCP specialize the PCP, and the MMP specializes the DSCP, if we take the set B to be the set of village bachelors and W the set of unmarried women. Similarly, the MCBP specializes the AFCP if we identify checker board squares with their positions in $\mathbb{N} \times \mathbb{N}$, so we have the theory graph on the right. There are two crucial insights that are important for solving the MCBP and that are driven by framing. The first one is that the DSCP is unsolvable unless $|B| = |W|$, as framing the problem as a matchmaking exercise will make clear even to non-mathematicians. The other insight is that by mapping the sets B and W in DSCP with the set of black and white squares respectively, then we obtain a view e into MCBP[3]. This allows to transport the insight that DSCP is unsolvable to the MCBP.

$$\begin{tikzcd}
\mathsf{MMP} & & \mathsf{MCBP} \\
\mathsf{DSCP} \arrow[u, "c"] \arrow[urr, dashed, "e"] & & \mathsf{AFCP} \arrow[u, "d"'] \\
& \mathsf{PCP} \arrow[ul, "a"] \arrow[ur, "b"'] &
\end{tikzcd}$$

2.5 Conclusions for SFDL Formats and Interaction with Frames

Let us now see how the theory-morphism based model fares with respect to the different aspects of framing shown in the examples and which insights it provides for SFDL formats as well as interaction design for implementing framing.

The first example uses frames to specialize abstract objects into concrete examples, adding details by fixing the base set of the monoid to $\mathbb{N}$ and the operation to the addition function. At the same time the frame can be used to generalize $\langle \mathbb{N}, + \rangle$ by abstraction. If we want to exploit frames for user interaction in MKM systems, the user should be enabled to select and change frames. In a theory graph a **frame generalization** can be seen as an extension from a frame f to a frame $\mathsf{f} \circ \mathsf{g}$. In this sense the framing e above can be seen as the generalization $\mathsf{c} \circ \mathsf{f}$, where we have generalized $\mathbb{N}$+Ex from an example for a non-group monoid to an example of a monoid. Here, the only possible *frame specialization* is taking back this generalization since we cannot change the framed theory. The example also shows that frame variants play an important role in understanding abstract mathematical objects and theories, and should therefore be supported by the interface (see for instance Figure 8 for a concrete example). If variant relations and loci are annotated in a SFDL, the explication of mathematical objects may become a simple planning exercise on theory graphs. Note that in our model of framing we can interpret the practice of giving examples as supplying the reader with a basic supply of prototypical framings (here Monoid is the framing theory for $\mathbb{N}$+Ex) that the reader can later draw upon for problem solving.

In the second example we see that the framing morphism drives problem solving. It opens the real-world situation to methods from Planar Geometry, identifies the salient features, and pulls back the geometric solution into the original problem over a conservative morphism. We can also observe another effect: the opportunity to *model framing in the SFDL* allows us to (partially) tackle the formalization divide. The current practice in formal methods is that informal problem descriptions remain as unstructured texts outside the (formal)

[3] Note that even if we frame the set S in the AFCP as squares in a rectilinear grid, they are still uncolored, therefore the target theory of the frame e has to be MCBP.

system. As a consequence, the relation between the original and the formal representation of the problem remains unclear and has to be accepted by a leap of faith. If we view formalization as a framing process in the SFDL, we can support it by MKM systems and take the guessing out of formalization.

In the third example we have used framing in two facilities: for problem solving via conservative extensions, but also in the form of views of the problems into situations that appeal to the intuitions of the (human) reader. This allows to anchor the abstract, mathematical concepts in the real world and thus trigger insights that help problem solving. There is an interesting situation for user interaction: say the user started out with the natural frame for MCBP, which she then generalized to d to view it as an AFCP and then further generalized the frame to $\mathsf{d}\circ\mathsf{b}$ to consider the original problem as a Pair Covering Problem. In this situation she can *specialize* the PCP to a Disjoint Set Covering Problem via a. Formally, we call a frame change $\mathsf{g} \mapsto \mathsf{g}'$ a **frame specialization via** f, iff $\mathsf{g}' \circ \mathsf{f} = \mathsf{g}$. And indeed $\mathsf{d} \circ \mathsf{b} \mapsto \mathsf{e}$ is one in our example. But *in* the problem solving phase, framing is not safe, therefore in the envisioned user interface, we need to allow speculative frame specialization. In the example we might want to further specialize $\mathsf{d} \circ \mathsf{b}$ to $\mathsf{c} \circ \mathsf{a}$ (beyond what is known in the theory graph) to study the Mutilated Checker Board Problem as a Matchmaker Problem and possibly establish a suitable theory morphism that justifies the frame specialization a posteriori.

$$\begin{tikzcd} & B \\ A' \arrow[ur, dashed, "\mathsf{g}'"] & A \arrow[u, "\mathsf{g}"'] \arrow[l, "\mathsf{f}"] \end{tikzcd}$$

Note that in all three examples the different, salient aspects of framing could directly be tied to the existence of suitable theory morphisms in the underlying content commons. In the following we will present a first MKM system that illustrates how framing can extend the user interaction.

3 SACHS: A Semantic Help System for MS Excel

We will illustrate how framing can extend the user interaction in a semantic help system under development at the German Center for Artificial Intelligence (DFKI), Bremen. The SACHS system (Semantic Annotation for a Controlling Help System [KK08a]), aims to address usability problems in spreadsheet-based applications. For details about the ideas and design decisions behind the SACHS system we refer the reader to our paper "*Compensating the Computational Bias of Spreadsheets with MKM Techniques*" in this volume. We only recap those aspects here that are relevant for our framing extension — which we refer to as "framing-aware SACHS".

A controlling system is a means for the organization to control finances, i.e. to *understand* profits and losses and draw conclusions, thus a lack of overview hampers the process: if users are not sufficiently informed they cannot optimize the company outcome. Even though MS Excel spreadsheets have the potential to serve well as an interface for a financial controlling system, they are more often than not too complex in practice. Even longtime users cannot interpret all data and are not certain about their origins.

A key observation in SACHS is that spreadsheets are *active documents* whose surface structure can adapt to the environment and user input. For SACHS we take a foundational stance and analyze spreadsheets as semantic documents, where the formula representation is the computational part of the semantic relations about how values were obtained. To compensate the diagnosed computational bias we propose to augment the two existing semantic layers of a spreadsheet — the surface structure and the formulae by one that makes the intention of the spreadsheet author explicit. We encode this intention in SFDL and can use it as a basis to provide multi-layered, semantic help services. As we cannot disclose DFKI financial data, we will use the traditional spreadsheet from [Win06] as a running example (see Figure 3).

	A	B	C	D	E	F
1	**Profit and Loss Statement**					
2						
3	(in Millions)		Actual		Projected	
4		1984	1985	1986	1987	1988
5						
6	**Revenues**	3,865	4,992	5,803	6,022	6,481
7						
8	**Expenses**					
9	Salaries	0,285	0,337	0,506	0,617	0,705
10	Utilities	0,178	0,303	0,384	0,419	0,551
11	Materials	1,004	1,782	2,046	2,273	2,119
12	Administration	0,281	0,288	0,315	0,368	0,415
13	Other	0,455	0,541	0,674	0,772	0,783
14						
15	Total Expenses	2,203	3,251	3,925	4,449	4,573
16						
17	**Profit (Loss)**	1,662	1,741	1,878	1,573	1,908

Fig. 3. A Simple Spreadsheet after [Win06]

The central concept we establish is that of a **functional block** in a spreadsheet, i.e. a rectangular region in the grid where the cells can be interpreted as input/output pairs of a *function*. For instance, the cell range [E9:F9] (highlighted with the selection of [E9] by a borderline) is a functional block, since the cells represent projected salary costs as a function π of time; the pair $\langle 1987, 0.617\rangle$ of values of the cells [E4] and [E9] is one of the pairs of π. The semantic help functionality of the SACHS system is based on an **interpretation**, i.e. a meaning-giving function that maps functional blocks to concepts in a SFDL. For instance our functional block [E9:F9] is interpreted to be the function of the projected salaries in a year for a business which we assume to be available as semantic background.

In [KK08a] we have presented the SACHS information and system architecture, and have shown how the semantic background can be used to give semantic help to the user on several levels like labels, explanations (as showcased in Figure 7) and dependency graphs (see Figure 4 on the right). For example, a user may not be aware that the spreadsheet concerns the profit statement of "SemAnteX Corp.", but can learn this from SACHS's dependency graph feature presented in Figure 4 by selecting cell [E9].

While the information about functional blocks and the meaning of their values (e.g. units), the provenance of data, and the meaning of formulae provided by the semantic background is a nice-to-have, in the development process it became painfully obvious that the interpretation (hence the information provided by the SACHS system to the user) is strongly dependent on the standpoint of the author — how she *frames* the data. In fact even the interpretation into a SFDL itself can be seen as a large frame. Therefore in the work reported in this

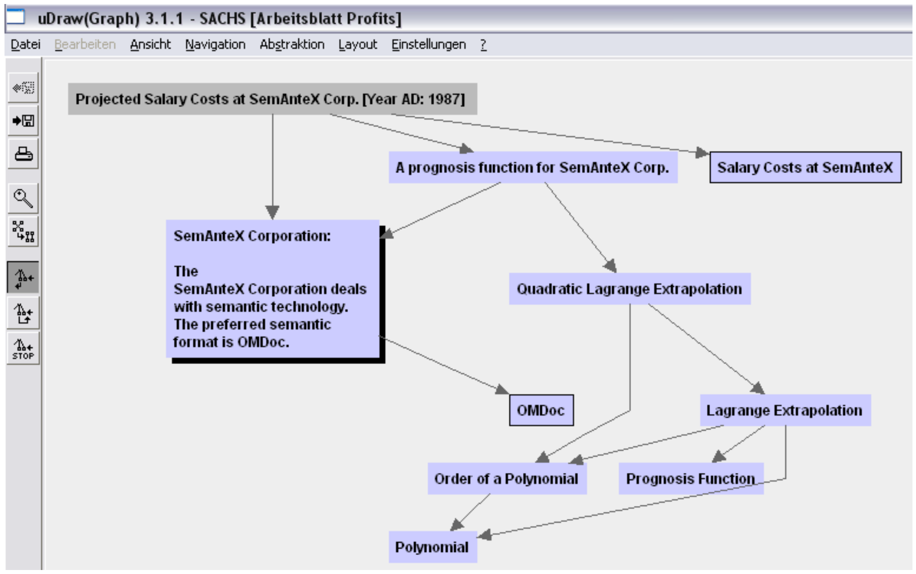


Fig. 4. Dependency Graph with 'uses'-Edges

paper we went one step further and integrated framing as part of the interface to give the user of the spreadsheet more control over the interaction.

3.1 Framing in SACHS

Semantic help systems need various kinds of information: concepts of the system, its user interface, input/output data, etc. For all of these, we need to know a lot about the objects themselves and their relations, i.e. we need ontologies about them. Generally, when we talk about interacting with knowledge-based systems, we have to distinguish knowledge about the system itself from knowledge structures about the domain the system addresses. We consider the first kind of knowledge as part of the *system ontology* and the second kind part of the *domain ontology*.

To distinguish between the system and domain ontologies, the following test suggests itself: anything the system is parametric in must be part of the domain ontology, anything that is particular to the system belongs to the system ontology. For instance, in SACHS the system ontology contains information about concepts like spreadsheet cells, functional blocks, the interpretation, etc. whereas domain ontologies include knowledge about monetary systems, accounting concepts, or prognosis. If the SACHS system were applied to grading spreadsheets, the system ontology that is tied to the underlying spreadsheet application would remain fixed, but the domain ontologies would need to talk about grades, students, semesters, courses, etc. Accordingly in semantics-based systems like SACHS, the domain-level functionality is driven by an explicit representation of the domain ontology — in the case of the SACHS system as an OMDoc-based SFDL.

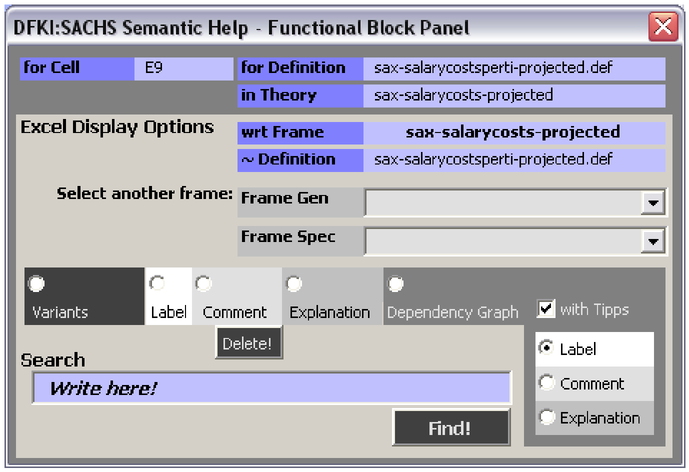


Fig. 5. SACHS's Functional Block Panel

As a consequence, we can also distinguish system- from domain-level framings in semantic help systems. Domain-level framings are triggered by theory morphisms in the SFDL, whereas the interaction design of the system must account for system-ontology level framings directly. In Figure 5 we find the SACHS panel extended by framing features. Once a cell is selected, the assigned definition in the SFDL with its home theory is shown as the *framed theory*. The natural framing theory determines the *framing theory* in the first step and all the background information is subsequently shown with respect to this frame. On the system-level the user is offered to change the frame via frame *generalization* or frame *specialization*. Moreover, in the field labeled "∼Definition" the corresponding definition in the chosen framing theory is presented. The user might also choose to recover domain-dependent *variants* from the semantic background.

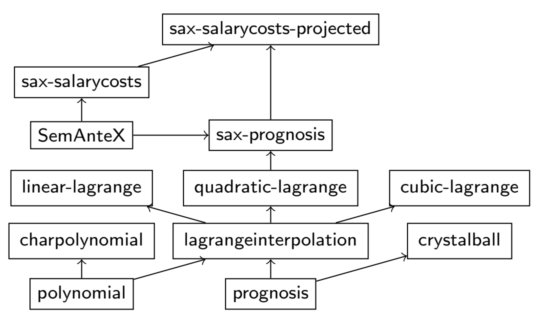


Fig. 6. A Fragment of the SACHS Domain-Ontology Theory Graph

To get a better understanding of the role of framing in the interaction with the SACHS system, let us have a closer look at the more specific use example for cell [E9] bearing the dependency graph in Figure 4 in mind, which tells us (among other stuff) that the number 0,617 was computed *a*) using a prognosis function adapted to SemAnteX, that is *b*) based on the Quadratic Lagrange Extrapolation function that is *c*) a Lagrange Extrapolation that is *d*) a function used for prognosis. To illustrate the framing potential we have to turn to the theory level of the semantic background sketched in Figure 6. Note that the home theory of cell [E9] — i.e. the theory that contains the definition sax-salarycosts-projected.def in the interpretation — is the theory sax-salarycosts-projected. It imports the theories sax-salarycosts and sax-prognosis. These theories can hence be used as *frame generalizations*. If we are more interested in the latter theory, we select it and get a new choice of frame generalizations SemAnteX and quadratic-lagrange. Choosing the latter the only available frame generalization becomes lagrangeinterpolation. Finally, here we can select prognosis as a frame for the projected salary costs at SemAnteX Corp.

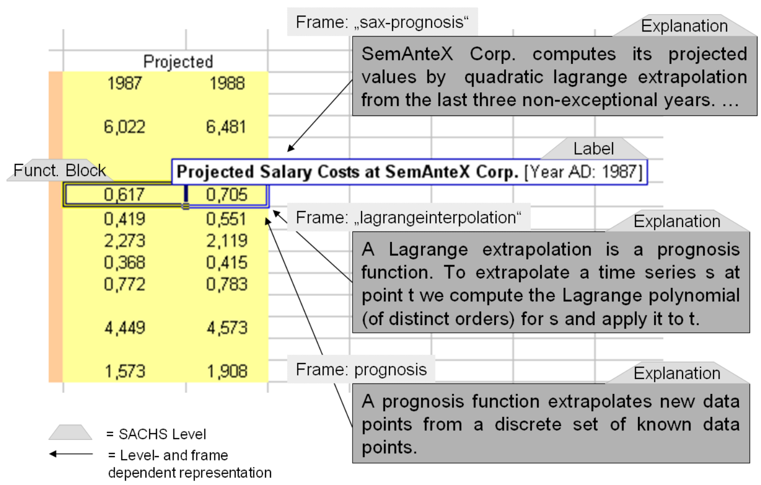


Fig. 7. Explanations within Distinct Frames

Importantly, with each change of frame the semantic information given to the user changes. For instance, in Figure 7 we can see different explanations for the same selected cell with respect to the resp. distinct frames. Note that usually the user can only get the information with respect to the author's framing as the resp. OMDoc document is fixed and consequently the imports-relation for the home theory. Another author might have chosen to e.g. import the lagrangeinterpolation theory directly instead of importing the more specific sax-prognosis. Here, the SACHS panel broadens the user's opportunities and takes back the rigor and subjectivity of the author's choice of framing.

The set of *frame specializations* wrt. a certain framing theory consists of all theories that import this framing theory. Frame specializations can supply the user with surprising insights. For example, the theory prognosis is imported by the theory crystallball, which offers the prognosis method of sitting in front of a crystal ball and — disregarding the data set — coming up with a mapping from times to values. With this, the reader may realize that there are always worse possible prognosis functions.

Another interesting service a framing-aware SACHS can offer is the display of *variants*. That is, the concrete framing assumption reified in the MS Excel formula for a cell can be changed. The conventional way to deal with such variants in a spreadsheet is to just replace the formulae in the functional block with new ones and see what the result is; a destructive and error-prone process at best. Given enough background knowledge we can do better. In our example, we have three theories specializing lagrangeinterpolation with concrete Lagrange extrapolations of different order, from which we can derive spreadsheet formulae, which in turn can be entered into the spreadsheet. In the example in Figure 8, we are looking for variants for the '∼Definition' lagrangeinterpolation.def in the framing theory for the definition sax-salarycostsperti-projected.def assigned by the author to cell [E9]. Concretely, selecting the option "Variants" in the SACHS panel shown in Figure 5 leads to the opening of the "Variants Panel" demonstrated in Figure 8. We see that there are three possible variants for the Lagrange extrapo-

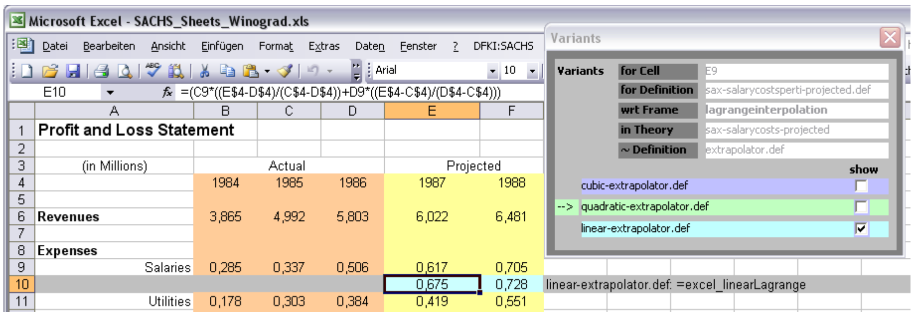


Fig. 8. Frame-based Variants

lation function: the linear, the quadratic, and the cubic Lagrange extrapolations. Remember that the quadratic one was used as the SemAnteX prognosis function, this is marked by the arrow in front of this variant. In the example the user selected the variant linear-extrapolator.def. Once the check box is checked the SACHS system generates new space in the spreadsheet (the light grey row 10 in Figure 8) enabling the presentation of the variant values for the entire functional block. The according variant formula (in the MS Excel formula box at the top of Figure 8) is evaluated. Note that framing influences which concrete variants are available: if we have framed [E9] as the result of a Lagrange extrapolation, we should be allowed to vary the order k of the Lagrange Polynomial (if we have enough data points). If we have however framed [E9] only as the result of a general prognosis function then we should also have crystal ball prognosis at our disposition as a variant.

4 Conclusion

In this paper we have analyzed a common mathematical practice from an MKM perspective, i.e. with an eye towards finding the underlying knowledge structures and representing them in content markup formats so that they can be exploited to support the practices in mathematical software systems. We model this practice of framing a mathematical object as establishing a theory morphism into a theory describing it. We have shown that in many paradigmatic framing cases, the model is able to account for the salient aspects of framing. The theory graph based model is appealing for MKM, since it allows to leverage a large body of existing work.

To test the model further, we have applied it in a situation that is only loosely coupled with classical mathematics: a semantic help system based on spreadsheets. The connection to our model of framing is that the semantic facilities feed on a semiformal digital background library that is theory graph structured. We have shown that taking framings into account in the user interface allows users to find their subjective perspective in the semantic help system. The necessary framing possibilities were naturally present in the background theory graph for our example. We attribute this to the fact that the theory graph was developed as a comprehensive overview over the background knowledge and not just tailored to the single spreadsheet application at hand.

Framing-aware interactions allow users to choose the right level of abstraction of explanations. But note that this is more than just another form of user-adaptivity. Frame-driven interaction broadens the users' opportunities as it allows them to become independent of the *author's framing* — e.g. her choice of concepts and level of rigor — by framing the material to fit their own particular background, their concrete situation, and their subjective goals. In a framing-blind interface, the author dominates the choice of these parameters.

In [KK08b, Section 3.3] we have analyzed requirements for semantic formats to be used in educational technology. In particular, we distinguished three contexts in educational situations: a "content context", a "learner context" and an "interaction context". Usually only the first two are recognized and operationalized in systems. Here, the choice of frames and the navigation between framings are part of the interaction context made explicit in the SACHS user interface. Interestingly, theory graphs that have been thought of as exclusively belonging to the content context now enable a simple formulation of a complex aspect of the interaction context.

Incidentally various learning theories discuss the framing practice as the basis for abstraction processes and ultimately as 'causes' for learning. For example, Klaus Holzkamp argues that every human being engages in an ever-present "*inner dialogue*" [Hol95, p. 25], the result of which turns into her specific actions. The dialogue entertains the idea of at least two distinct frames that inform the learning process. This suggests that framing is also an essential practice in any learning environment, hence the application of this MKM technology might reach much further than the application discussed in this paper.

Finally, framing-aware systems allow the user to explore variants afforded by the background knowledge. In controlling systems this seems to be especially useful to test variant modeling assumptions (like our prognosis functions), but testing variants is a central practice in the sciences and engineering as well.

To close the circle to our introduction, we believe that eventually, the MKM community should build systems that support *what mathematicians do*. In particular, they should exploit theory graphs to support the practice of framing in the mathematical domain proper as we strongly conjecture that such systems will be better suited to re-enliven reified mathematical knowledge.

References

[BC00] Henk Barendregt and Arjeh M. Cohen. Representing and handling mathematical concepts by humans and machines. In *ISSAC '00: Proceedings of the 2000 international symposium on Symbolic and algebraic computation*, New York, NY, USA, 2000. ACM.

[BF06] Jon Borwein and William M. Farmer, editors. *Mathematical Knowledge Management, MKM'06*, number 4108 in LNAI. Springer Verlag, 2006.

[CCB06] A.M. Cohen, H. Cuypers, and E. Reinaldo Barreiro. Mathdox: Mathematical documents on the web. In OMDoc – *An open markup format for mathematical documents [Version 1.2]* [Koh06], chapter 26.7, pages 278–282.

[Far00] William Farmer. An infrastructure for intertheory reasoning. In David McAllester, editor, *Automated Deduction – CADE-17*, number 1831 in LNAI, pages 115–131. Springer Verlag, 2000.

[FGT92] William Farmer, Josuah Guttman, and Xavier Thayer. Little theories. In D. Kapur, editor, *Proceedings of the 11th Conference on Automated Deduction*, volume 607 of *LNCS*, pages 467–581, Saratoga Springs, NY, USA, 1992. Springer Verlag.

[Hol95] Klaus Holzkamp. *Lernen: Subjektwissenschaftliche Grundlegung*. Campus Verlag, 1995.

[KK06] Andrea Kohlhase and Michael Kohlhase. Communities of Practice in MKM: An Extensional Model. In Borwein and Farmer [BF06], pages 179–193.

[KK08a] Andrea Kohlhase and Michael Kohlhase. Compensating the semantic bias of spreadsheets. In Joachim Baumeister and Martin Atzmüller, editors, *Wissens- und Erfahrungsmanagement LWA (Lernen, Wissensentdeckung und Adaptivität) Conference Proceedings*, volume 448, 2008.

[KK08b] Andrea Kohlhase and Michael Kohlhase. Semantic knowledge management for education. *Proceedings of the IEEE; Special Issue on Educational Technology*, 96(6):970–989, June 2008.

[KMM07] Michael Kohlhase, Achim Mahnke, and Christine Müller. Managing Variants in Document Content and Narrative Structures. In Alexander Hinneburg, editor, *Wissens- und Erfahrungsmanagement LWA (Lernen, Wissensentdeckung und Adaptivität) conference proceedings*, pages 324–229. Martin-Luther-University Halle-Wittenberg, 2007.

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

[KP06] Manfred Kerber and Martin Pollet. A tough nut for mathematical knowledge management. In Michael Kohlhase, editor, *Mathematical Knowledge Management, MKM'05*, number 3863 in LNAI, pages 81–95. Springer Verlag, 2006.

[KWZ08] Fairouz Kamareddine, J.B. Wells, and Christoph Zenglere. Computerizing mathematical text with mathlang. *Electron. Notes Theor. Comput. Sci.*, 205:5–30, 2008.

[MAH01] Till Mossakowski, Serge Autexier, and Dieter Hutter. Extending development graphs with hiding. In H. Hußmann, editor, *Fundamental Approaches to Software Engineering (FASE 2001)*, number 2029 in LNCS, pages 269–284. Springer Verlag, 2001.

[RK08] Florian Rabe and Michael Kohlhase. An exchange format for modular knowledge. In Piotr Rudnicki and Geoff Sutcliffe, editors, *Knowledge Exchange: Automated Provers and Proof Assistants (KEAPPA)*, November 2008.

[RK09] Florian Rabe and Michael Kohlhase. A web-scalable module system for mathematical theories. Manuscript, to be submitted to the Journal of Symbolic Computation, 2009.

[Sch09] Charles F. Schmidt. Productive thinking...the gestalt emphasis. `http://www.rci.rutgers.edu/~cfs/305_html/Gestalt/gestalt.html`, 2009.

[SRWB86] David A. Snow, E. Burke Rochford, Steven K. Worden, and Robert D. Benford. Frame alignment processes, micromobilization, and movement participation. *American Sociological Review*, 51(4):464–481, 1986.

[Win01] Wolfgang Windsteiger. On a solution of the mutilated checkerboard problem using the theorema set theory prover. In S. Linton and R. Sebastiani, editors, *Proceedings of the Calculemus 2001 Symposium*, pages 28–47, Siena, Italy, 2001.

[Win06] Terry Winograd. The spreadsheet. In Terry Winograd, John Bennett, Laura de Young, and Bradley Hartfield, editors, *Bringing Design to Software*, pages 228–231. Addison-Wesley, 1996 (2006).