Electronic Communications of the EASST
Volume 62 (2013)

Specification, Transformation, Navigation
Special Issue dedicated to Bernd Krieg-Brückner
on the Occasion of his 60th Birthday

Spreadsheets with a Semantic Layer

Andrea Kohlhase, Michael Kohlhase

20 pages

Guest Editors: Till Mossakowski, Markus Roggenbach, Lutz Schröder
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Spreadsheets with a Semantic Layer

Andrea Kohlhase, Michael Kohlhase

Jacobs University Bremen

Abstract: Spreadsheets are active documents that are heavily employed in administration, financial forecasting, education, and science because of their intuitive, flexible, and direct approach to computation. But they are also error-prone, poorly documented, often contain actual data in legacy form. Therefore, assistance for high-impact spreadsheet users is needed. To determine what kind of help could be useful, we analyze user expectations with an "Wizard-of-Oz" experiment. This shows that background knowledge is missing in spreadsheets.

In the SACHS project we approach the missing background knowledge by adding a semantic layer. We illustrate spreadsheets with a semi-formal domain ontology and equip them with a semantically transparent interface that allows new forms of interaction like "semantic navigation", "framing", or "playing with variants", on which a survey is given. Moreover, an integration of assessment knowledge into the SACHS approach is presented. We model it based on theory graphs and sketch a potential SACHS extension with innovative assessment interaction.

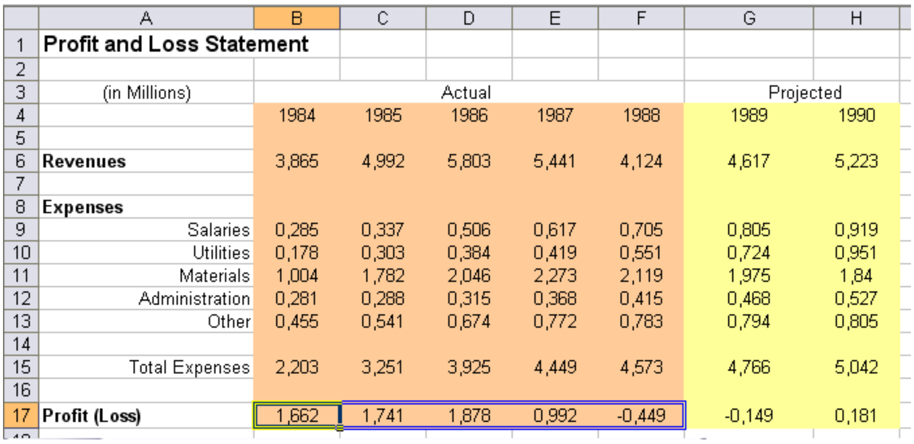
Keywords: Spreadsheets, Semantic Web, Knowledge Management

1 Introduction

In many spreadsheet-based applications even longtime users cannot interpret all data and are not certain about their origins (see [AE06] and its references). This often results in errors on the data level and misinterpretation or misapprehension of the underlying model. Usability and maintenance problems are not only well-known, they have severe consequences, see for example [Mur08]. It has been estimated that each year tens of millions professionals and managers create hundreds of millions of spreadsheets [Pan00]. Abraham and Erwig report an astounding error rate of up to 90% (!) in spreadsheets [AE06].

In this article we show that (interactional) semantic knowledge management techniques can be used to enhance the interaction with spreadsheets and alleviate usability problems caused by spreadsheet complexity. In particular, we give a survey on the user assistance system "**SACHS**" (**S**emantic **A**nnotation for a **C**ontrolling **H**elp **S**ystem) that aims at overcoming usability issues for MS Excel documents.

To deepen our understanding of the underlying problem, let us first discuss why spreadsheet technology is so attractive and why this sets knowledge traps. Spreadsheets are **active documents**, i.e., they are on the one hand of a document type that distinguishes between form and content (like cell layout vs. computed cell values), and on the other hand they exploit the distinction by using a presentation engine like Excel to adapt the surface structure of the document to the environment or user input. For example, if an author adds a column, the concerned relative



	A	B	C	D	E	F	G	H
1	Profit and Loss Statement							
2								
3	(in Millions)	Actual					Projected	
4		1984	1985	1986	1987	1988	1989	1990
5								
6	**Revenues**	3,865	4,992	5,803	5,441	4,124	4,617	5,223
7								
8	**Expenses**							
9	Salaries	0,285	0,337	0,506	0,617	0,705	0,805	0,919
10	Utilities	0,178	0,303	0,384	0,419	0,551	0,724	0,951
11	Materials	1,004	1,782	2,046	2,273	2,119	1,975	1,84
12	Administration	0,281	0,288	0,315	0,368	0,415	0,468	0,527
13	Other	0,455	0,541	0,674	0,772	0,783	0,794	0,805
14								
15	Total Expenses	2,203	3,251	3,925	4,449	4,573	4,766	5,042
16								
17	**Profit (Loss)**	1,662	1,741	1,878	0,992	-0,449	-0,149	0,181

Figure 1: Running Example: A Simple Controlling System Using MS Excel after [Win06]

cell references in all formulas of the spreadsheet are automatically updated by the presentation engine. Interestingly, the potential activeness of the document is not at all exploited when reading a spreadsheet. To remedy this we must first understand the ways activeness supports authoring.

The underlying semantic model for the content of spreadsheets consists of various layers (see [KK09a]). We will expose them using the spreadsheet in Figure 1, which we will use as a running example for this article[1]:

Data Cell values are contained in the data layer. These are the objects that are actively handled by the presentation engine. Their meaning is defined in other layers.

Surface The surface layer sets visual cues for the interpretation of the shown data. For instance, the grid of cells seen in Figure 1 can be roughly divided into three areas. The darker, ochre area in the center contains values of actual and past expenses and revenues; the lighter, yellow box on the right contains values projected from these. The white region that surrounds both boxes supplies explanatory text or header information that helps users to interpret these numbers. We call meaningful grid regions like the ochre one, here the one containing *actual* data, **semantic blocks**. Generally, non-empty cells that do not contain input or computed values contain text strings that give auxiliary information (comparable to a legend) on the cells that do. The author's cell format decision also influences the surface layer as one and the same datum can be for example presented with a chosen currency symbol or as a percentage rate.

Formula Authors experience a great deal of satisfaction from spreadsheets, because cell values can be computed automatically via assigned, sometimes very complex underlying formulae. Here, a **formula** is an expression built up from constants, an extended set of numeric and logic operators, and references to other cells.

[1] The SACHS system was developed for the **DCS** system, a financial controlling system based on Excel in daily use at the German Research Center for Artificial Intelligence (DFKI). Of course, we cannot use that as an example here for privacy reasons.

In our example, the yearly profit as cell value in [B17 : F17] can be computed from the resp. revenues in [B6 : F6] and the total expenses in [B15 : F15] by a simple subtraction, the total expenses can in turn be computed by summing up the various particular expense categories listed in cells [A9 : A13]. We call grid regions containing cells with the same underlying formula like [B17 : F17] **computational functional blocks**. Note that the projected profit values in cell range [G17 : H17] do not belong to this computational functional block as they are calculated as projected values, but they could have been computed as well within this computational functional block as difference between the projected revenues and expenses.

Background The background layer in a spreadsheet is invisible as it concerns the knowledge that is needed for the interpretation of data, that is not made explicit in the document. For example, the author of the spreadsheet in Figure 1 knew which business the data are taken from or which one they are concerned with, but a reader of this document has to know this information to understand its sense.

The interpretation of the data in row 17 for instance consists in the profit/loss situation *over time* (i.e., in the years 1984-1990 as indicated by the values in row 4). In particular, the meaning of the values in row 17 is that they represent profits and losses as a *function* π of time: recall that a function is a right-unique relation (i.e., a set of pairs of input values and output values such that for every input there is a unique output). In our example the pair $\langle 1984, 1.662\rangle$ of values of the cells [B4] and [B17] is one of the pairs of π. We call cell ranges each of which corresponds to a specific such function **implicit functional blocks**. Note that the cell range [B17 : F17] in our running example is a computational functional block, whereas it is only part of the implicit functional block [B17 : H17].

The distinct notions of “semantic”, “computational functional” and “implicit functional block” show the complexity of making sense of spreadsheets. Even though spreadsheets are appealing as they are deceptively easy to use, they are also error-prone but high-impact, widely-disseminated but poorly documented, and contain actual data in legacy form (see e.g. [Pan00, Mur08]). Errors caused by misunderstandings at the surface and formula layer are tackled via user assistance for the spreadsheet player itself. For instance, in [AN08] a multi-layer interface approach is used and realized for Excel — especially addressing the problem of appropriateness of explanation. Support for comprehending spreadsheets is often concerned with data visualization techniques and data/formula dependency graphs (see [BP08] and [HM08] resp. as examples), which only cover the data layer.

In contrast to these layers that can be handled once and for all at the level of the system, the domain modeled in spreadsheets and with that the background knowledge is specific to the spreadsheet itself and must thus be handled specially. This makes the treatment of background knowledge in the background layer into an iceberg that endangers safe passage of spreadsheets. The background layer is touched in [Din09] with a documentation-through-annotation approach, that formulates a collection of design guidelines for supporting rich annotation in spreadsheets. Issues concerning the interpretation of spreadsheets are presented in [BM08]. To the best of our knowledge all of the available solutions for missing background knowledge attack the problem on the social or organizational level. But no *system* has yet been developed to provide user assistance for spreadsheets on the background layer.

In the next section we explore the hidden iceberg of background knowledge in spreadsheets using a **“Wizard-of-Oz” experiment** — a research method in which subjects interact with a computer system that they believe to be autonomous, but which is actually being operated or partially operated by an unseen human being (see [DJA93]). In particular, we wanted to determine distinct knowledge kinds necessary for ‘understanding the numbers’ in spreadsheets. The results suggested to start building our user assistance system for spreadsheets SACHS with the goal to integrate previously only implicit domain knowledge. We give a short overview of the SACHS architecture and system in Section 3. In particular, it illustrates DCS with a semi-formal domain ontology and makes it available with a semantically transparent interface. We show that the semantic enhancement of spreadsheets with SACHS not only enables access to more knowledge, it also allows new forms of spreadsheet interaction. Furthermore, the “Wizard-of-Oz” experiment yielded the need for different knowledge kinds. For example, spreadsheet users often are concerned with questions like “*Is it good or bad for my business if this cell has value 0.992?*” concerning assessment knowledge. In Section 4 we present a model for the knowledge involved with assessing numbers and concepts within a spreadsheet based on theory graphs. We sketch how this can be incorporated into the SACHS system by providing paper prototypes. Section 5 concludes the article and discusses future research directions.

2 Background Knowledge in Spreadsheets

HODNIGG and MITTERMEIR state that “*comprehension of a workbook is non-trivial as there are several factors that aggravate its comprehension.*” [HM08, p. 82]. But what are the necessary factors for comprehension? To develop the domain ontology for the background knowledge of the DFKI Controlling system DCS we organized interviews with a DFKI expert on the topic and recorded them as MP3 streams[2]. Even though these interviews were not originally intended as a “Wizard of Oz” experiment, in the following we will interpret them so because any “Wizard of Oz” experiment would have been isomorphic in setup and outcome[3]. Here, the interviewee plays the part of an ideal SACHS system and gives help to the interviewer who plays the part of the user.

This experiment gives us valuable insights about what background knowledge consists of and how it is — or should be — organized. Conceptually, the experiment reveals that the DCS system only models the factual part of the situation it addresses, while important aspects for ‘understanding the numbers’ remain implicit. Concretely, it yields *qualitatively distinct explanations* for a user assistance system, which the expert thought was necessary to understand the specific controlling system spreadsheet.

When studying the MP3 streams, we were surprised that in many cases a question of “*What is the meaning of ...*” for a specific knowledge item was answered by the expert with up to six of the following seven **explanation types**, the occurrence rate of each relative to the number of knowledge items is listed in the brackets (“What is the percentage of this specific explanation

[2] We recorded three interview sessions amounting to approximately 1.5 hrs concerning 39 distinct knowledge items and containing 110 explanations.

[3] The only difference is that the interviewers were aware that they were talking to a human. We contend that this is immaterial, since the aim of the experiment is to find out which kinds of knowledge were expected, not how a concrete system interface works.

type being selected for describing a knowledge item?"):

1. **Definition (Conceptual)** [71.8%] A *definition* of a knowledge item like a functional block is a thorough description of its meaning. For example the functional block "cover ratio per project in a research area" was defined as the percentage rate to which the necessary costs are covered by the funding source and own resources.
2. **Purpose (Conceptual)** [46.2%] The *purpose* of a knowledge item in a spreadsheet is defined by the spreadsheet author's intention, in particular, the purpose explains why the author put the information in. A principal investigator of a project or the respective department head e.g. needs to get the information about its cover ratio in order to know whether either more costs have to be produced to exploit the full funding money or more equity capital has to be acquired.
3. **Assessment of Purpose** [30.8%] Given a purpose of a knowledge item in a spreadsheet, its reader must also be able to reason about the purpose, i.e., the reader must be enabled to draw the intended conclusions/actions or to *assess the purpose*. For understanding whether the cover ratio is as it is because not enough costs have yet been produced, the real costs have to be compared with the necessary costs. If they are still lower, then the costs should be augmented, whereas if they are already exploited, then new money to cover the real costs is needed.
4. **Assessment of Value** [51.3%] Concrete values given in a spreadsheet have to be interpreted by the reader as well in order to make a judgement of the data itself, where this *assessment of the value* is a trigger for putting the assessment of purpose to work. For instance, the size of the cover ratio number itself tells the informed reader whether the project is successful from a financial standpoint. If the cover is close to 100%, "everything is fine" would be one natural assessment of its value.
5. **Formula** [23.1%] With a given formula for a value in a spreadsheet's cell the reader knows exactly how the value was computed, so that she can verify her understanding of its intention against the author's. Note that a lot of errors in spreadsheets result from this distinction. In our experiment, if a value of a cell was calculated with a formula explicitly given in the spreadsheet, then the expert explained the dependency of the items in the formula, but restricted from just reading the formula aloud. In particular, he pointed to the respective cells and tried to convey the notion of the formula by visualizing their dependency, not so much what the dependency was about.
6. **Provenance** [43.6%] The *provenance* of data in a cell describes how the value of this data point was obtained, e.g. by direct measurement, by computation from other values via a spreadsheet formula, or by import from another source; see [MGM^{+}08] for a general discussion of provenance. In our interviews — as many of the data of the concrete spreadsheet were simply an output of the underlying controlling data base — the provenance explanations mostly referred to the specific data base where the data comes from. But when the formula for a value was computed, but not within Excel, the expert tried to give the formula as provenance information, e.g. in the case of the cover ratio. This knowledge was often very difficult to retrieve afterwards for the creation of the semantic document.
7. **History** [15.4%] The *history*, i.e., the creation process of a spreadsheet over time, often is important to understand its layout that might be inconsistent with its intention. For instance, if an organizational change occurs that alleviates the controlling process and

makes certain information fragments superfluous, then those fragments will still be shown in the transition phase and beyond, even though their entropy is now 100% in the most of cases.

These seven explanation types were distilled from the recorded set of 110 explanations. The percentages given can function as a *relevance ranking* done by the expert with respect to the importance of explanation types for providing help.

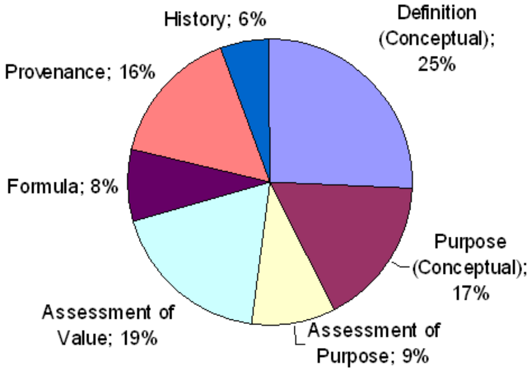


Figure 2: Explanation Types

Figure 2 portrays the distribution of occurrences according to each type (“How often occurred this specific explanation type out of all explanations?”). The “Wizard of Oz” experiment interpretation suggests that Figure 2 showcases the user requirements for SACHS as a user assistance system (see also [NW06]).

Excel is able to give help for 8% of the explanations we found in the help of a human expert, whereas not surprisingly Definition explanations are the most frequent ones.

A realized domain ontology together with an appropriate interface (as realized in the SACHS system described in the next section) bumps this up to 33%. Even though this is certainly an improvement, it leaves much more to be desired than we anticipated.

3 SACHS: A Semantic Help System for MS Excel

Semantic technologies like the Semantic Web add novel functionalities to existing information resources by enhancing them with explicit representations of the underlying knowledge objects and their relations and exploiting them for computing new information. The information resources in spreadsheets are cells and grids, and it is not clear how they can be enhanced semantically because they as objects belong to the spreadsheet player, in our case Excel. If we tie user assistance to these objects, then we either get generic help like in an Excel manual, particularly not help for understanding a specific spreadsheet, or we have to extend the (possibly proprietary) player software. Therefore, we used a different approach for the use of semantic technologies which we call **Semantic Illustration**: Instead of enhancing resources into semiformal ontologies by annotating them with formal objects that allow reasoning as in the Semantic Web ‘paradigm’, here a semantic system *illustrates* a software artifact $\mathscr{A}$ (an application, program, or document) with a semiformal ontology by complementing it with enough information to render new semantic services. This approach contains a somewhat analogous requirement phrased in [Tag09]. Conceptually, a help system $\mathscr{H}$ is independent of $\mathscr{A}$, even though an implementation may well integrate it into $\mathscr{A}$. In any case $\mathscr{H}$ is related to $\mathscr{A}$ via an interpretation mapping, so that it can serve as a “**semantic ally**” for $\mathscr{A}$. The Semantic Illustration approach opens the ‘use of semantic data' for non-semantic software applications: Any system with formal data can be mashed up with semantic applications. In SACHS we made use of the fact that spreadsheets are active documents whose surface structure *can* adapt to the environment and user input. Here, a specific DCS spreadsheet is the artifact $\mathscr{A}$ for which SACHS provides a help system $\mathscr{H}$.

The SACHS system is an add-in for MS Excel 2003 written in Visual Basic for Applications (VBA). The system was actively developed 2008-2009 and has been kept stable since then, since we are planning to re-implement it on a newer technology basis (XML-based Office Suite and the new OMDoc technology stack). The system is in a state, where it can be used to explore the information architecture and user interactions afforded by it, but it is not ready for general dissemination. Nonetheless, the code is available from the authors upon request.

In SACHS cells in the spreadsheet are linked by the "interpretation mapping" to elements of an accompanying OMDoc document that contains a representation of the background knowledge. In particular, all cells in a functional block are linked to the same OMDoc `definition` — the definition of its intended function. This assignment is internally represented by an extra worksheet within the spreadsheet, that is maintained by the spreadsheet author. With respect to the concrete DCS spreadsheets SACHS was created for, these were semi-automatically generated.

3.1 Semi-Formal Ontologies as Theory Graphs

In SACHS we utilize a semi-formal domain ontology for the respective spreadsheet as background layer. This ontology is encoded in the OMDoc format, which allows to mix informal elements (e.g. descriptions in natural language) with formal structure elements (like interlinked axioms, definitions, theorems, and proofs, organized into theories). The main structure we utilize here is that of a **theory graph** (i.e., a graph of theories interlinked by theory morphisms; see [RK13] for details), which can be understood as an object-oriented organization of the background knowledge corresponding to the spreadsheet: Closely related information objects (definitions, descriptions, statements of properties, etc.) are grouped into theories related by the **imports** relation, which corresponds to a functional dependency relation. In particular, the nodes of a theory graph consist of theories and the edges are given by imports relations; Figure 3 demonstrates the richness of a theory graph even for a small example like the running one.

For more details consider the situation on the right side of Figure 4, where we base a theory of profits in enterprises on theories of revenues and expenses. Formally the links (thin arrows) in this graph have in common that all assertions that are true in the link source are true in the link target (possibly after translation). Say SemAnteX Profit contains the assertion

(*) *If c^* owns c, then the larger $\pi(c, now)$, the better for c^*.*

which is proven from axioms in an envisioned theory Income that is imported by SemAnteX Profit. This allows us to introduce the other type of link in a theory graph: views. Formally, a **view** is a mapping of concepts from the source theory to the target theory, such that all axioms and definitions in the source theory are true in the target theory. Consider for instance a theory The More the Better with the single assumption that "*the more of a commodity x I obtain the better it is for me*", then the mapping that maps x to $\pi(c, now)$, where c is a company I own stock in, is a view by virtue of assertion $(*)$.

Theory graphs with imports relations and views support very efficient reuse of information. For user assistance, they support multiple explanations. In our example, we could now also

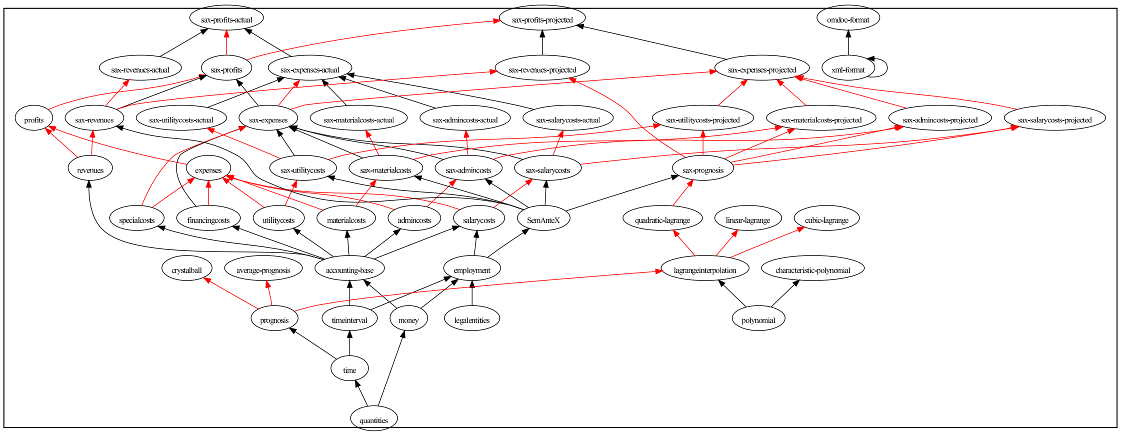


Figure 3: The Complexity of a Theory Graph of Background Knowledge for Figure 1

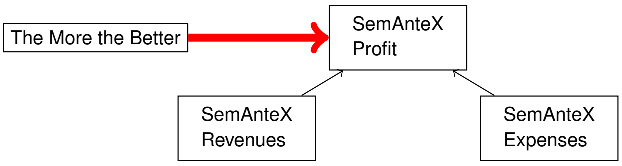
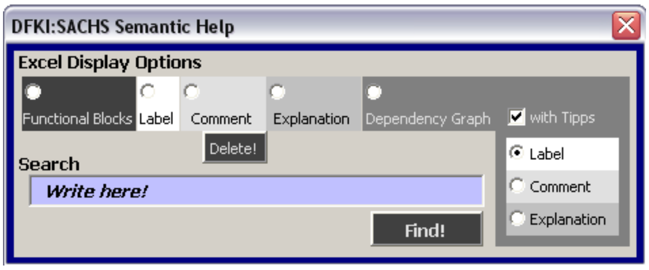


Figure 4: A Simple Theory Graph

explain the concept of a profit via the red view rather than the two imports relations, resulting in an explanation "*the more profit you make the better you will be off*". In essence, a theory graph can be viewed as an "and/or graph" for user assistance systems, in our example we can explain SemAnteX Profit by SemAnteX Revenues *and* SemAnteX Expenses *or* by The More the Better — depending on prior knowledge and preferences of the user. Note that views theoretically allocate help provisions for distinct context cultures as described in [Hug08]. We utilize the theory graph structure of the background ontology of a spreadsheet.

3.2 In-Place Help

One of the major issues for user assistance systems consists in "providing help at an appropriate level" (e.g. [NW06]). In recent years, the demand for *embedded* user assistance, i.e., user assistance that is provided without having a user to push a help button and go searching for help, has grown and was even called the "*future for software help*" [Ell07]. But what does 'embedded' really signify? The Excel objects that carry meaning are the cells. They are interpreted by the user in both the grid layout like within a table with an assigned row and column specification and the underlying formula. With SACHS we offer a


DFKI:SACHS Semantic Help

Excel Display Options

○ Functional Blocks ○ Label ○ Comment ○ Explanation ○ Dependency Graph ☑ with Tipps

◉ Label
○ Comment
○ Explanation

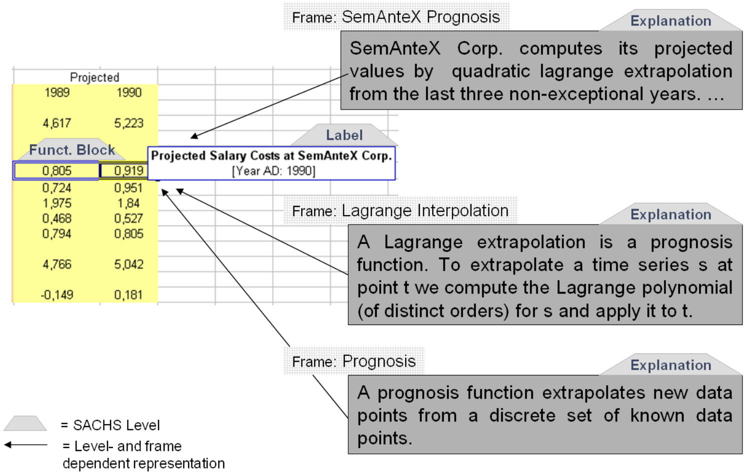
Search

Write here!

Figure 5: The SACHS Panel

third interpretation by aligning cells with concepts in the domain ontology. Hence, we realized embeddedness by using cell clicks as entry points for the help system, so that every click on a cell generates help.

Concretely, the SACHS system addresses the Definition explanation type with the direct help text generator shown in the SACHS panel in Figure 5. The OMDoc domain ontology has three text slots of different granularity that can be used for help texts. Accordingly, the SACHS panel offers the choice of getting "**labels**" (a title), "**comments**" (a short description), or "**explanations**" (a detailed description); see Figure 6 for an example of the various granularities. The generated help texts are enhanced by instantiating the parameters with concrete values from the cell context. For instance the time interval is instantiated to the year 1990 in Figure 6. The SACHS panel also offers to search the domain ontology for concepts that contain user-given strings, to present them once selected and (if available) to link to the according cell. In the functional block mode every cell click highlights all those cells that functionally belong together (as showcased in Figure 1 and 6).


Frame: SemAnteX Prognosis
Explanation
Projected
Funct. Block
Label
Frame: Lagrange Interpolation
Explanation
Frame: Prognosis
Explanation
= SACHS Level
= Level- and frame dependent representation

SemAnteX Corp. computes its projected values by quadratic lagrange extrapolation from the last three non-exceptional years. …

1989	1990
4,617	5,223
0,805	0,919
0,724	0,951
1,975	1,84
0,468	0,527
0,794	0,805
4,766	5,042
-0,149	0,181

Projected Salary Costs at SemAnteX Corp.
[Year AD: 1990]

A Lagrange extrapolation is a prognosis function. To extrapolate a time series s at point t we compute the Lagrange polynomial (of distinct orders) for s and apply it to t.

A prognosis function extrapolates new data points from a discrete set of known data points.

Figure 6: SACHS-Generated Help Texts for Cell [H9] in Figure 1

3.3 Semantic Navigation

Another option in the SACHS panel is the generation of a **dependency graph** for the concept connected to the selected cell. For instance, if this option is chosen for cell [B15] (expenses 1984), then the first two levels of the graph as seen in Figure 7 are generated. Concretely, all the different expense types like "Salary Costs" and "Utility Costs", that build up the expenses of the company called "SemAnteX", are listed as well as a node for the company itself. If the user wants to elaborate on a specific concept like "Salary Costs", then a click on the corresponding node expands it by another level (as happened for Figure 7). This feature is comparable to hyperlinks in help texts, but adds **semantic navigation** cues. We mashed-up the graph-based interface with the interactions needed within a spreadsheet to allow the user to navigate the spreadsheet via the structured background ontology by the definitional structure of the intended functions. Here, the color-coding of the nodes indicates whether the concept is connected to a specific cell in the workbook. Darker grey means that it is available on the active spreadsheet, lighter grey hints that the assigned cell is on another spreadsheet but still within the active workbook, and light violet points out a semantic concept with no connection to spreadsheets. Note that the user has the choice of text granularity in each node (via right mouse click) or all nodes (via SACHS panel).

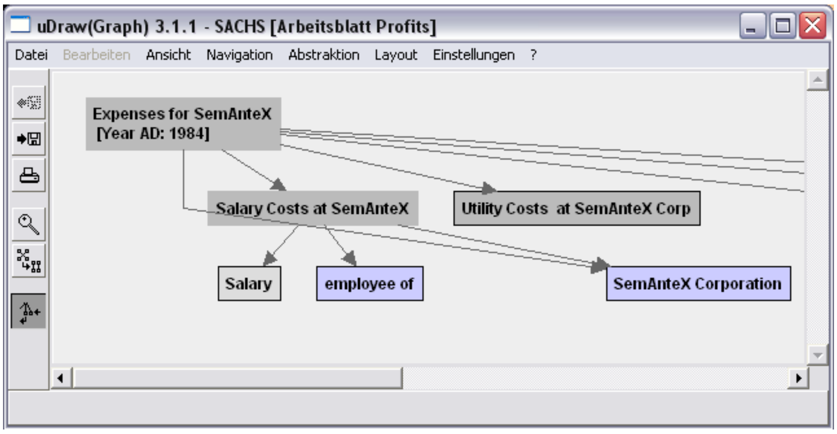


Figure 7: Dependency Graph for Cell [B15]

3.4 Framing

While the domain knowledge provided by the semantic background is important information, first evaluations made it clear that the interpretation mapping (hence the information provided by the SACHS system to the user) is strongly dependent on the author's point of view — how she *frames* the data.

If we understand framing as the practice of viewing novel situations in terms of something already understood, then we can now model the framing practice by defining a **framing** to be the establishment (creating or choosing) of an imports relation or view from a source theory (the **framing theory**) into the theory representing the problem (the **framed theory**). Hence, a frame is understood as a scaffolding of concepts that influence the understanding of situations. In Figure 4 the theory The More the Better and theories Revenues and Expenses represent a frame for Profit. The view explanation makes use of framing to anchor the explanation of profit in the user's previous experience.

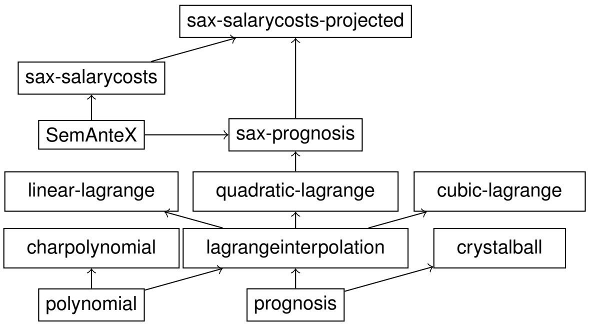
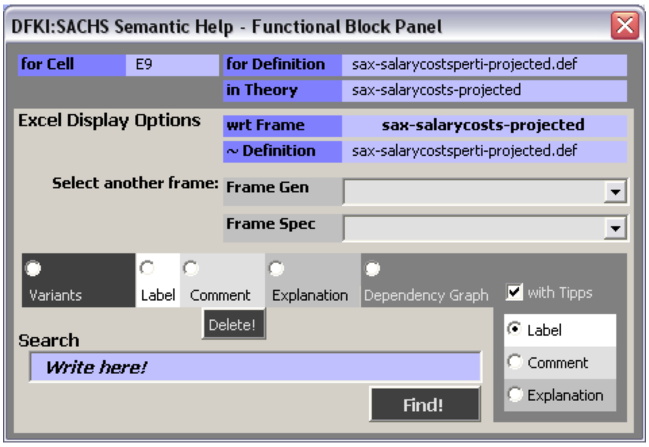


Figure 8: A Fragment of the SACHS Domain-Ontology Theory Graph

Let us look at the graph in Figure 8 that represents a part of the domain ontology theory graph. The projected salary costs in cell [G9] feed on two concepts: salary costs and a projection “sax-prognosis”, that is a project-specific prognosis function. Both concepts are realized in distinct theories, which are represented in Figure 8. If we follow the graph, then we can see that the cell value in [G9] is computed with a quadratic LaGrange function, which is a LaGrange interpolation, which is used as prognosis function. Now, we don't know which information the user reading our spreadsheet wants to know: Is she interested which specific LaGrange function is used, is she interested more generally what a LaGrange interpolation is, or does she wonder what a projection is? All these are different frames. A first **theory of framing** based on theory-graphs and theory morphisms was developed in [KK09b] and implemented in SACHS.

Note that the concept of frames does not depend on cells but really on functional blocks. Therefore, SACHS' interface for framing was implemented in the functional block mode. Concretely, in Figure 9 we find the SACHS panel extended by framing features. Once a cell is selected, the assigned definition in the ontology 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. The user is offered to change the frame via frame *generalization* or frame *specialization*.


DFKI:SACHS Semantic Help - Functional Block Panel

for Cell: E9
for Definition: sax-salarycostsperti-projected.def
in Theory: sax-salarycosts-projected

Excel Display Options
wrt Frame: sax-salarycosts-projected
~ Definition: sax-salarycostsperti-projected.def

Select another frame: Frame Gen
Frame Spec

☑ with Tipps
● Label
○ Comment
○ Explanation

Search: Write here!

Figure 9: The SACHS Functional Block Panel

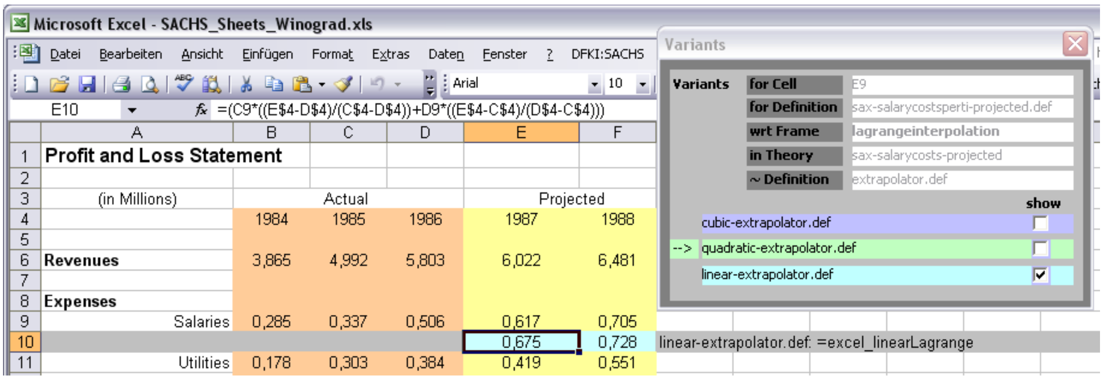
Note that the home theory of cell [G9] — i.e., the theory that contains the definition sax-salarycosts-projected.def in the interpretation — is the theory sax-salarycosts-projected as seen in the SACHS panel in Figure 9. It imports the theories sax-salarycosts and sax-prognosis. These theories can hence 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. After 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.

Importantly, with each change of frame the semantic information given to the user changes. For instance, in Figure 6 we can see different explanations for the same selected cell with respect to distinct frames. Note that usually the user can only get the information regarding the author's framing as the 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.

3.5 Frame Variants

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 10, 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 9 leads to the opening of the "Variants Panel" demonstrated in Figure 10.


Microsoft Excel - SACHS_Sheets_Winograd.xls
Datei Bearbeiten Ansicht Einfügen Format Extras Daten Fenster ? DFKI:SACHS
Arial
10
Variants

E10 =(C9*((E$4-D$4)/(C$4-D$4))+D9*((E$4-C$4)/(D$4-C$4)))

	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					0,675	0,728
11	Utilities	0,178	0,303	0,384	0,419	0,551

linear-extrapolator.def: =excel_linearLagrange

Variants

for Cell	E9
for Definition	sax-salarycostsperti-projected.def
wrt Frame	lagrangeinterpolation
in Theory	sax-salarycosts-projected
∼ Definition	extrapolator.def

		show
	cubic-extrapolator.def	□
-->	quadratic-extrapolator.def	□
	linear-extrapolator.def	☑

Figure 10: Frame-Based Variants

We see that there are three possible variants for the Lagrange extrapolation function: the linear, the quadratic, and the cubic Lagrange extrapolations. Remember that the quadratic one was originally used as the SemAnteX prognosis function (indicated by the arrow in front of this variant in Figure 10). 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 10) 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 10) is evaluated.[4] 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] This implementation is just a prototype: one limitation is that computed values on the spreadsheet might change by this action as Excel automatically adjusts any existing formula that use cells moved by the insertion. This is not a problem though, since the user's focus is on the values of the selected and unchanged functional block and its variant (otherwise using the variants isn't sensible).

3.6 Background Knowledge Coverage by SACHS?

To evaluate the user assistance situation for spreadsheets with the SACHS system: the Formula explanation type (8%) is rudimentarily covered by Excel, and the Definition explanation is offered by SACHS (25%). It can be argued that the ontology-based SACHS architecture is well-suited to cope with Purpose explanations (17%) — indeed, some of the purpose-level explanations have erroneously found their way into definitions for the DCS, where they rather should have been classified as 'axioms and theorems' (which are currently not supported by the SACHS interface). The explanation category Provenance (16%) has been anticipated in the SACHS architecture (see [KK09a]) but remains unimplemented in the SACHS system. The Assessment of Purpose type (9%) is completely missing from SACHS as well as History (6%) and Assessment of Value (19%).

4 Assessment

Now we take up the problem of Assessment of Value explanations. On the one hand, it is ranked second in the list of explanation types with a stunningly high percentage of 51.3%, which can be interpreted as the second-best type of explanations from the point of view of our expert. On the other hand, the very nice thing about assessment for computational data is that we can hope for a formalization of its assessment in the form of formulas, which can be evaluated by the spreadsheet player in turn.

4.1 Modelling Assessment

A naive approach of complementing spreadsheets with assessment knowledge could be the inclusion of Assessment of Value information into the definition text itself. This is — ontologically speaking — a very impure approach as such judgements do not solely depend on the concept itself. For instance, they also depend on the respective Community of Practice: At one institution e.g. a cover ratio of 95% might be judged as necessary, at another 100% (or more) might be expected. Clearly we need a better theory of modeling assessment.

So before we address the question of how to model assessment, first we have to take a closer look at assessment itself: What is it about? Assessments consist of value judgements passed on situations modeled by (parts of) spreadsheets. As such, we claim that assessments are deeply in the semantic realm. To strengthen our intuition, let us consider some examples; we will use a slightly varied version of the simple spreadsheet document in Figure 1, which we have already used in [KK09a, KK09b] for this. The following can be considered typical assessment statements:

I) "*Row* 6 *looks good.*"

II) "*The revenues look good.*"

III) "*I like this* [points to cell [E17]] *but that* [points to cell [F17]] *is a disaster.*"

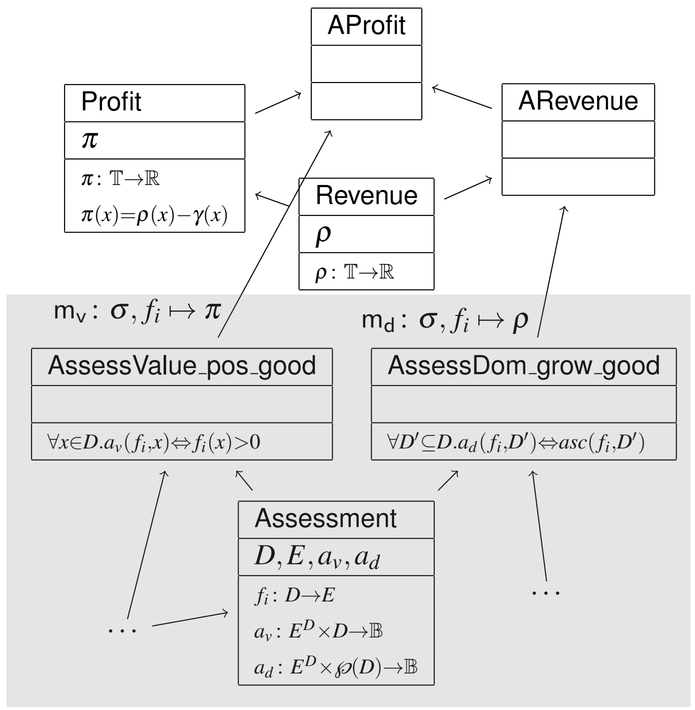
IV) "*I like the profit in 1987 but of course not that in 1988.*"

V) "*Upper Management will be happy about the leftover funds in* [nn] *that they can now use elsewhere, but the PI of the project will be angry that he got less work out of the project than expected. Not to mention the funding agency; they cannot be told of this at all, because* *it violates their subsistence policy.*"

On the surface, the first statement refers to a row in the spreadsheet, but if we look closer, then we see that this cannot really be the case since if we shift the whole spreadsheet by one row, then we have to readjust the assessment. So it has to be about the intended meaning of row 6, i.e., the development of revenues over the years. Indeed we can paraphrase I with II — another clue that the assessments are really about situations modeled by a functional block in the spreadsheet. But assessments are not restricted to functional blocks as statements III and IV only refer to individual cells. Note again that the statements are not about the numbers 0.992 and -0.449 (numbers in themselves are not good or bad, they just are). Here the assessment seems to be intensional, i.e., about the intension "the profit in 1987/8" rather than the extension. Another way to view this is that the latter two assessments are about the argument/value pairs $\langle 1987, 0.992\rangle$ and $\langle 1988, -0.449\rangle$. We will make this view the basis of our treatment of assessment in SACHS: We extend the background ontology by a set of assessment theories that judge the intended functions in the functional blocks of the spreadsheet on their functional properties.

4.2 Assessment via Theories and Morphisms

Consider the partial theory graph in Figure 11, which we will use to account for the assessments in the examples I to IV above. The figure shows the theories Revenue and Profit which are part of the background knowledge, the **assessed theories** ARevenue and AProfit, and the **assessment theories** (set in the gray part) that will cover assessment itself.



AProfit
Profit
π
$\pi\colon \mathbb{T}\to\mathbb{R}$
$\pi(x)=\rho(x)-\gamma(x)$
ARevenue
Revenue
ρ
$\rho\colon \mathbb{T}\to\mathbb{R}$
$\mathsf{m_v}\colon\ \sigma, f_i \mapsto \pi$
$\mathsf{m_d}\colon\ \sigma, f_i \mapsto \rho$
AssessValue_pos_good
$\forall x{\in}D.a_v(f_i,x){\Leftrightarrow}f_i(x){>}0$
AssessDom_grow_good
$\forall D'{\subseteq}D.a_d(f_i,D'){\Leftrightarrow}asc(f_i,D')$
Assessment
D, E, a_v, a_d
$f_i\colon D{\to}E$
$a_v\colon E^D{\times}D{\to}\mathbb{B}$
$a_d\colon E^D{\times}\wp(D){\to}\mathbb{B}$
...
...

where $\mathbb{B}$ is the set of Boolean values, $\mathbb{R}$ is the set of real numbers, and $\mathbb{T}$ the set of time intervals (over which profits are measured). Furthermore, $\sigma := \{D \mapsto \mathbb{T}, E \mapsto \mathbb{R}\}$

Figure 11: A Partial Assessment Graph for Profits

The theory Assessment provides three concepts: a generic function f_i (used as a placeholder for the intended function of the functional block we are assessing), a function a_v for assessing whether a value in a cell is 'good', and finally a function a_d for assessing whether a function is 'good' over a subdomain. This generic theory — note that this does not provide any judgements yet, only the functions to judge — is then refined into concrete assessment theories by adding axioms that elaborate the judgement functions a_v and a_d, which are then used to provide concrete judgement functions to the assessed theories, via interpreting theory morphisms. The theory AssessValue_pos_good restricts the interpretation of a_v so that it assesses the function f_i as ‘good’ on an argument x, iff $f_i(x)$ is positive, and the theory AssessDom_grow_good restricts the interpretation of a_d to a function asc to evaluate f_i as ‘good’ on a subdomain $D' \subseteq D$, iff f_i is increasing on D'. Note that these assessments are still on the ‘generic function’ f_i over a ‘generic domain’ D with a ‘generic range’ in E. These are made concrete by the theory morphisms $\mathsf{m_v}$ and $\mathsf{m_d}$ that map these concrete sets and functions into the assessed theories, thereby applying the judgmental axioms in the assessment theories in the assessed theories.

Of course theories AssessValue_pos_good and AssessDom_grow_good are just chosen to model the examples from the start of this section. A realistic formalization of assessment, would provide a large tool-chest of theories describing the “shape” of the function f_i for knowledge engineers to choose from. With this, providing a judgement about a value becomes as simple as choosing a cell and an assessment theory: the cell determines the intended function, with its domain and range and thus the mapping of the theory morphism. Thus the assessed theory can be constructed automatically by the SACHS system.

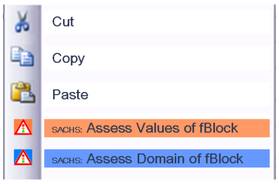
In our example we have restricted ourselves to unary functions, but of course it is very simple to provide assessment theories for any arity that occurs in practice. Moreover, we have only used assessment theories that only refer to inherent properties of the intended functions (e.g. being monotonically increasing), but many real-world assessments are context-dependent. E.g. one might want the profit of a German Company to grow more rapidly than the DAX. This is where the knowledge-based approach we are proposing really starts to shine: we just add an assessment theory with an axiom

$$\forall t.a_v(f_i,t) \Leftrightarrow \frac{f_i(t)}{f_i(p(t))} > \frac{\mathrm{DAX}(t)}{\mathrm{DAX}(p(t))}$$

where $p(t)$ is the predecessor time interval of t.

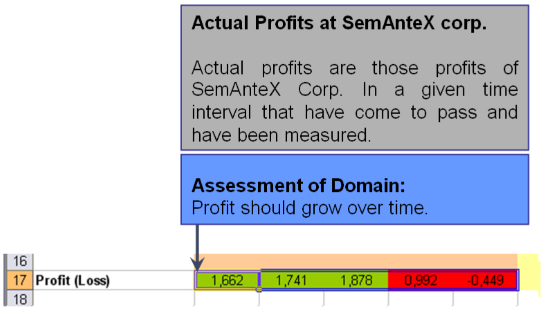
4.3 The Envisioned Assessment Extension in SACHS

We will now show how assessments modeled in the theory graph can be made useful for the user. As the assessments are bound to (the intended function of) a functional block, we extend the context menu with entries for all assessment functions. On the right we assume a right mouse click on the cell [B17] to show the context menu with the two assessment functions a_v and a_d.


Cut
Copy
Paste
SACHS: Assess Values of fBlock
SACHS: Assess Domain of fBlock

When the “Assess Values of fBlock” entry is selected, SACHS is put into a special “assessment mode”, which brings assessment information to the user’s attention. In the background the SACHS system determines the version of the a_v axiom inherited by the AProfit, translates it into an Excel formula, and evaluates it to obtain the judgements.

Here the axiom is $\forall t.a_v(\pi,t) \Leftrightarrow \pi(t) > 0$, and it is evaluated on all cells in the functional block, resulting in the values t,t,t,t,f, which SACHS color-codes as shown in Figure 13 to warn the user of any cells that get a negative judgement. At the same time, the assessment mode extends the explanatory labels by explanations texts from the background ontology. Selecting the menu element “Assess Domain of fBlock” gives the result in Figure 12.

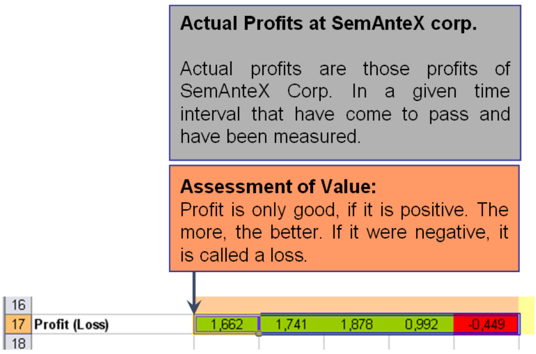


Actual Profits at SemAnteX corp.

Actual profits are those profits of SemAnteX Corp. In a given time interval that have come to pass and have been measured.

Assessment of Domain:
Profit should grow over time.

Figure 12: Assess the *Domain*



Actual Profits at SemAnteX corp.

Actual profits are those profits of SemAnteX Corp. In a given time interval that have come to pass and have been measured.

Assessment of Value:
Profit is only good, if it is positive. The more, the better. If it were negative, it is called a loss.

Figure 13: Assess the *Values*

But as the assessments are synchronized with the assessed theories in the background theory graph, we can also analyze the assessments for possible causes. Recall that profits are defined as the difference between revenues and expenses, it makes sense to trace assessments through the dependency graph provided by the SACHS system for understanding the definitional structure of the spreadsheet concepts.

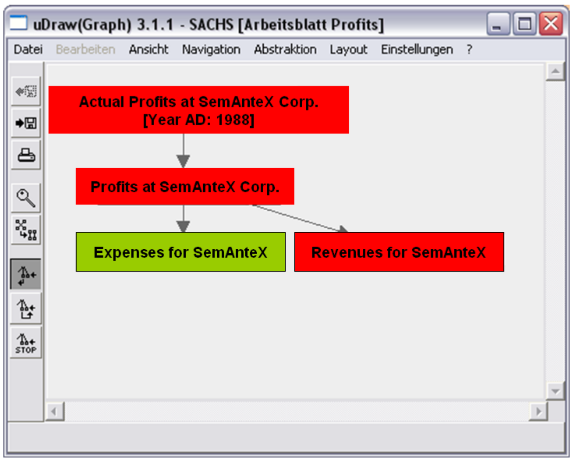
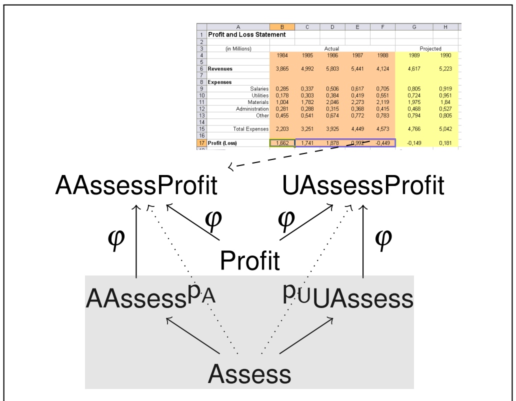


Figure 14: Assess *all* Values



$\varphi := \sigma, f \mapsto \pi$ and σ as in Figure 11

Figure 15: Multi-Context Assessment

Note that this analysis is anchored to the cell: Figure 14 shows the definitional graph for the negatively assessed cell [F17] for the profits in the year 1988. Here the revenues are also negatively assessed (color-coded red in the definitional graph), so the problem might be with the revenues. Note as well that this graph cannot be used for a causal analysis, as the arrows here are still definitional dependency relations. We conjecture that causal analysis knowledge can transparently be included in the background ontology and can be made effective for the user in a similar interface. But we leave this for further research.

4.4 Multi-Context Assessments and Framing

Note that the assessments above are “author assessments” since they are supposedly entered into the background ontology by the spreadsheet author. But the author’s assessment is not the only relevant one for the user to know: In the exemplary assessment statement V in Section 4.1 we have a single explanation that refers to three different assessments that differ along the role of the “assessor”. Multiple assessment contexts can be accommodated in our proposed model — any user of the system can enter assessments. These user assessments can even be stored in a private extension to the background ontology if the user does not have write access to the system-provided one.

In fact we can enable multi-context assessment by just providing the a_v and a_d functions with another argument that determines a fitting user or Community of Practice (see [KK06] for an introduction to Communities of Practice and their reification in the background knowledge). This will generally get us into the situation in Figure 15, where we have an assessment of profits by the author — in theory AAssessProfit — and one by the user — UAssessProfit (we have abstracted from the internal structure of the theories). The dashed arrow is the (functional) interpretation that maps the functional block to the author-assessed theory.

In the framing-based user interface described above we use imports relations and views as framings and provide frame-based exploration of variants. In this example the canonical frame (the identity morphism from AAssessProfit to itself) can be generalized to the frame p_A with source theory Assess, which spans a frame variant space that includes the frame p_U and thus the user assessment, which the user can choose to explore this assessment. Needless to say, this works for any number of assessments (private or public).

5 Conclusion

We have analyzed the reasons for users’ difficulties in understanding and appropriating complex spreadsheets, as they are found e.g. in financial controlling systems. A Wizard-Of-Oz experiment shows that one of the causes for this is that underlying semantic documents are biased to computational aspects and fail to model the provenance, interpretation, and ontological relations of the objects and concepts operationalized by the system. To remedy the situation we propose to explicitly model the intention of a spreadsheet as an intention model in an OMDoc-encoded theory graph that serves as an explicit knowledge base for a spreadsheet help system. We have developed the SACHS system that draws on such intention models to offer various semantic services that aid the user in understanding and interacting with the spreadsheets.

An unanticipated benefit of our design decision to use theory graphs (a data structure inherited from formal methods in Software Engineering) was that we could model the practice of framing a mathematical object as establishing an imports relation or view into a theory describing it. The model is able to account for the salient aspects of framing: 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.

But the Wizard-Of-Oz experiment also revealed that significant categories of explanations are still systematically missing from this setup, severely limiting the usefulness of the system. We have tried to extend the background ontology with a model of assessment to see whether the Semantic Illustration paradigm is sufficiently flexible to handle assessment. The proposed model shows that this is indeed the case, but still has various limitations. For instance, the need to pollute the background ontology with one new theory per assessment theory, where each assessed theory is largely empty seems somewhat unnatural and possibly intractable. Also, we lack a convincing mechanism for coordinating the exploration of assessment variants: In our example in Figure 1, if we change the assessment of a profit value, we would like to change that of the respective revenue cell to a corresponding assessment.

Finally, we have only talked about Assessment of Value explanations. It seems that we can model Purpose and Assessment of Purpose explanations with a similar construction as the one proposed in Section 4.1: We start out with a base assessment theory which provides an assessment function like a_v, which in turn acts on a generic intended function f_i of the functional block in question. But this — instead of mapping into Boolean values — maps into a set of purposes and tasks formalized in a 'task ontology' by which we would extend the background ontology. This might also make it possible to generate explanations for assessments in SACHS.

Acknowledgements: This paper contains our view on the foundations of the SACHS (funded by DFKI) and the ensuing SiSsI project (funded by the German Research Council under grant KO-2484-12-1). Both project aim to extend semantic document modeling to spreadsheets. Bernd Krieg-Brückner has been proposing for years to do this. He always maintained that spreadsheets would be ideal targets for semantic preloading. Indeed our research shows that spreadsheets are semantically much more interesting than anticipated. We also thank the other SACHS and SiSsI project members as well as Klaus Hofmann for valuable discussions. The quality of the article was greatly enhanced by the helpful feedback of our reviewers, which we really appreciate.

Bibliography

[AE06] R. Abraham, M. Erwig. Inferring templates from spreadsheets. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*. Pp. 182–191. ACM, New York, NY, USA, 2006. doi:http://doi.acm.org/10.1145/1134285.1134312

[AN08] O. D. Andrade, D. G. Novick. Expressing Help at Appropriate Levels. In *SIGDOC'08 Conference Proceedings*. Pp. 125–130. ACM, 2008.

[BM08] D. A. Banks, A. Monday. Interpretation as a factor in understanding flawed spreadsheets. *CoRR* abs/0801.1856, 2008.

[BP08] R. Brath, M. Peters. Spreadsheet Validation and Analysis through Content Visualization. *CoRR* abs/0803.0166, 2008.

[CDSW09] J. Carette, L. Dixon, C. Sacerdoti Coen, S. M. Watt (eds.). *MKM/Calculemus Proceedings*. LNAI 5625. Springer Verlag, July 2009.

[Din09] M. Dinmore. Documenting Problem-Solving Knowledge: Proposed Annotation Design Guidelines and their Application to Spreadsheet Tools. *CoRR* abs/0908.1192, 2009.

[DJA93] N. Dahlbäck, A. Jönsson, L. Ahrenberg. Wizard of Oz studies: why and how. In *Proceedings of the 1st international conference on Intelligent user interfaces*. IUI '93, pp. 193–200. ACM, New York, NY, USA, 1993.
doi:http://doi.acm.org/10.1145/169891.169968
http://doi.acm.org/10.1145/169891.169968

[Ell07] M. Ellison. Embedded user assistance: The future for software help? *interactions* 14(1):30–31, 2007.
doi:http://doi.acm.org/10.1145/1189976.1189997

[HM08] K. Hodnigg, R. T. Mittermeir. Metrics-Based Spreadsheet Visualization: Support for Focused Maintenance. *CoRR* abs/0809.3009, 2008.

[Hug08] M. Hughes. User Assistance: Writing for a High-Context Culture. Online publication (http://www.uxmatters.com/mt/archives/2008/05/user-assistance-writing-for-a-high-context-culture.php), May 2008. Accessed on 2009-05-15.

[KK06] A. Kohlhase, M. Kohlhase. Communities of Practice in MKM: An Extensional Model. In Borwein and Farmer (eds.), *Mathematical Knowledge Management (MKM)*. LNAI 4108, pp. 179–193. Springer Verlag, 2006.
http://kwarc.info/kohlhase/papers/mkm06cp.pdf

[KK09a] A. Kohlhase, M. Kohlhase. Compensating the Computational Bias of Spreadsheets with MKM Techniques. Pp. 357–372 in [CDSW09].
http://kwarc.info/kohlhase/papers/mkm09-sachs.pdf

[KK09b] A. Kohlhase, M. Kohlhase. Spreadsheet Interaction with Frames: Exploring a Mathematical Practice. Pp. 341–356 in [CDSW09].
http://kwarc.info/kohlhase/papers/mkm09-framing.pdf

[MGM+08] L. Moreau, P. Groth, S. Miles, J. Vazquez, J. Ibbotson, S. Jiang, S. Munroe, O. Rana, A. Schreiber, V. Tan, L. Varga. The Provenance of Electronic Data. *Communications of the ACM* 51(4):52–58, 2008.
doi:http://doi.acm.org/10.1145/1330311.1330323

[Mur08] S. Murphy. Spreadsheet Hell. *CoRR* abs/0801.3118, 2008.

[NW06] D. G. Novick, K. Ward. What Users Say They Want in Documentation. In *SIGDOC'06 Conference Proceedings*. Pp. 84–91. ACM, 2006.

[Pan00] R. R. Panko. Spreadsheet Errors: What We Know. What We Think We Can Do. In *Symp. of the European Spreadsheet Risks Interest Group (EuSpRIG 2000)*. 2000.

[RK13] F. Rabe, M. Kohlhase. A Scalable Module System. 2013. Manuscript, submitted to Information & Computation.
http://kwarc.info/frabe/Research/mmt.pdf

[Tag09] T. Tague. The Big Picture – How Semantic Technologies Introduce a New Paradigm for Interaction. Invited talk at the Semantic Technology Conference, 2009.
http://www.semantic-conference.com/session/2120/

[Win06] T. Winograd. The Spreadsheet. In Winograd et al. (eds.), *Bringing Design to Software*. Pp. 228–231. Addison-Wesley, 2006.