What we *understand* is what we get: Assessment in Spreadsheets

Abstract

Spreadsheets are heavily employed because of their intuitive, flexible, and direct approach to computation. In previous work we have studied how an explicit representation of the background knowledge associated with the spreadsheet can be exploited to alleviate usability problems with spreadsheet-based applications. The `SACHS` system implements this approach to provide a semantic help system for DCS, an Excel-based financial controlling system. In this paper, we evaluate the (comprehension) coverage of the `SACHS` system with a "Wizard of Oz" experiment and see that while `SACHS` fares much better than DCS alone, it systematically misses important classes of explanations. For supporting judgement about the information contained in spreadsheets, we provide a first approach for an "assessment module" in `SACHS`.

1 Introduction

Spreadsheets are great active documents, they are intuitive, flexible, and offer a direct approach to computation. Unfortunately, an obverse statement is eqally true, as they are error-prone but high-impact, widely-disseminated but poorly documented, and contain actual data in legacy form (see e.g. [Pan00, Mur08]). Support for comprehending spreadsheets is often concerned with data visualization techniques and data/formula dependency graphs (see [BP08] and [HM08] resp. as examples). User assistance like a help system seems to be another valuable approach but still largely missing except for a documentation-through-annotation approach in [Din09] and conceptual recognition of an interpretation issue in [**?**].

In our previous research we addressed this issue with semantic technology resulting in the `SACHS` system [KK09a, KK09b, KK09c, KK09d]. It is a semantic help system for "**DCS**", a financial controlling system based on Excel [Mic89] in daily use at the German Research Center for Artificial Intelligence (DFKI). Here, a spreadsheet is illustrated with a semi-formal ontology of the relevant background knowledge via an interpretation mapping. The formal parts of the ontology are then used to control the aggregation of help texts (from the informal part of the ontology) about the objects in the spreadsheet. This enables in turn new semantic interaction services like "semantic navigation" or "framing" (see [KK09d]) from within the spreadsheet.

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? With the `SACHS` system in a usable state, we have evaluated coverage with a Wizard-of-Oz experiment. Interestingly, this has revealed that the DCS system only models the factual part of the situation it addresses, while important aspects for 'understanding the numbers' remain implicit — and as a consequence the `SACHS` system also fails to tackle them. For instance, users often ask questions like "*Is it good or bad if this cell has value 0.992?*" and experienced controllers may tell users "*Cell E6 must always be higher than E15*". We consider this knowledge (which we call **assessment knowledge**) to be an essential part of the background knowledge to be modeled in the semantically enhanced spreadsheet systems, since we can only profit from help if it is understood in 'all' its consequences. In particular, the assessment knowledge must be part of user assistance (e.g. answering the first question) and can be used to issue warnings (e.g. if the controller's invariant inherent in the second statement is violated).

	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: A Simple (Extended) Spreadsheet after [Win06]

To keep the paper self-contained we give a short overview of the SACHS system in the next section, followed by the coverage evaluation experiment in section 3. We will present a preliminary approach for modelling the background knowledge involved in assessment in section 4 and envision how this can be used in the SACHS system in section 5. Section 6 concludes the paper and discusses future research directions.

2 SACHS: Semantic Annotation for a Controlling Help System

For SACHS a foundational stance is used, where spreadsheets are considered as semantic documents. In particular, the formula representation represents the computational part of the semantic relations about how values were obtained. To compensate this diagnosed computational bias the two existing semantic layers of a spreadsheet — the surface structure and the formulae — are augmented by one that makes the intention of the spreadsheet author explicit (see [KK09a]).

There, the central concept is that of a **functional block**[1] in a spreadsheet, i.e., a rectangular region in the grid where the cells can be interpreted as input/output pairs of a *function* (the **intended function** of the functional block). For instance, the cell range [B17:F17] in figure 1[2] (highlighted with the selection of [B17] by a borderline) is a functional block, since the cells represent profits as a function π of time; the pair $\langle 1984, 1.662\rangle$ of values of the cells [B4] and [B17] is one of the pairs of π. The semantic help functionality of the SACHS system is based on an **interpretation** mapping, i.e., a meaning-giving function that maps functional blocks to concepts in a background ontology. For instance our functional block [B17:F17] is interpreted to be the function of "Actual Profits at SemAnteX Corp." which is assumed to be available in the semantic background.

In [KK09a] the SACHS information and system architecture is presented. It is 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 3) and dependency graphs like the one for cell [G9] in figure 2. This graph-based interface allows the user to navigate the structured background ontology by definitional structure of intended functions. In this case the graph

[1] Our notion of "functional block" seems to be somewhat similar to ABRAHAM's "*regions*" [Abr05]

[2] This spreadsheet is our running example, also taken up in section 4.

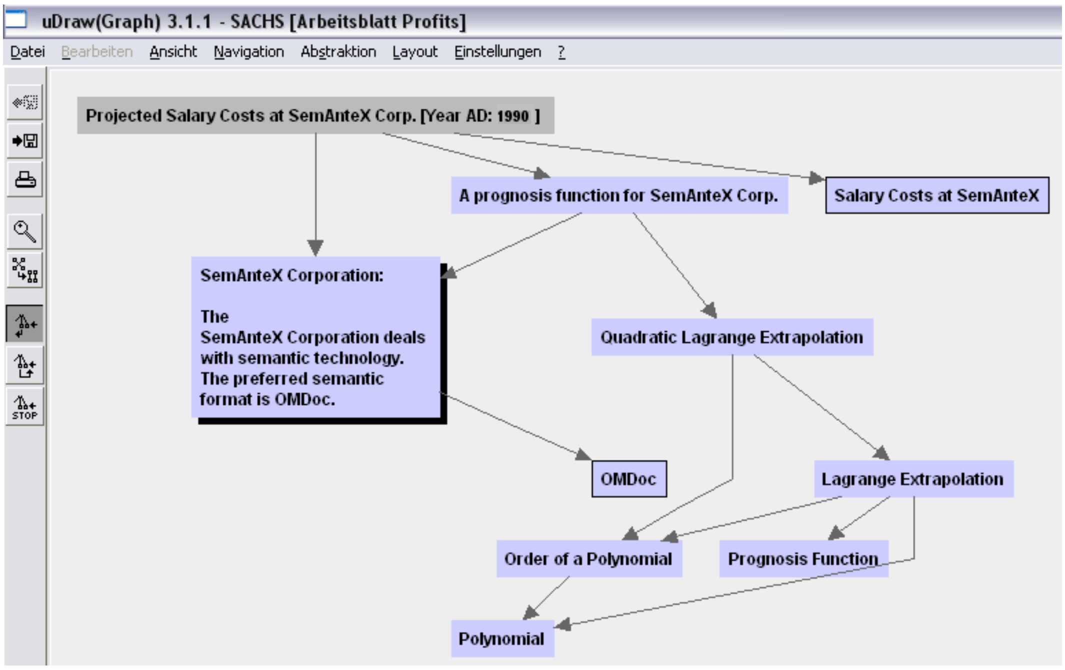


Figure 2: Dependency Graph with 'uses'-Edges

also reveals that the spreadsheet concerns the profit statement of the business "SemAnteX Corp.", which is not represented in the spreadsheet alone.

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 are important information, the development process made it painfully clear that the interpretation (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. A first **theory of framing** based on theory-graphs and theory morphisms was developped in [KK09d]. Based on this extended interaction facilities were developed. Among others, this enables the SACHS system to (i) tailor the help texts to the frame chosen by the user (and thus presumably to the task she pursues; see three distinct explanations in figure 3), and (ii) to provide frame alternatives for exploring the space of possible **spreadsheet variants** e.g. for different prognosis scenarios.

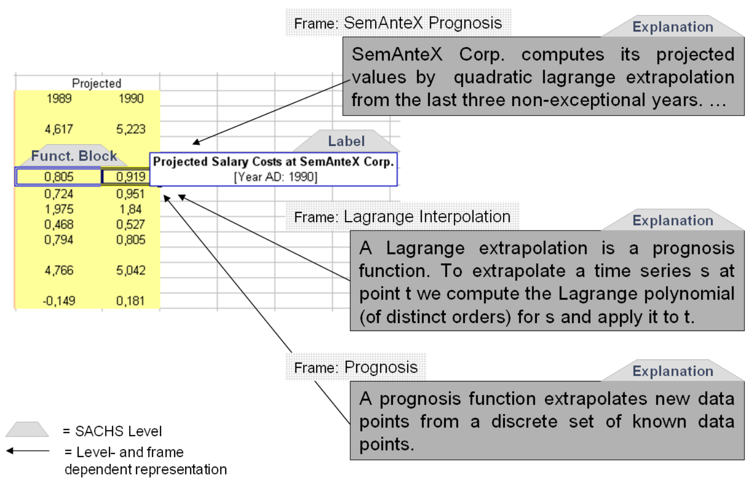


Figure 3: Explanations within Distinct Frames

3 Help Needed, but Where?

To develop the domain ontology for the background knowledge of the DFKI Controlling system we organized interviews with a DFKI expert on the topic and recorded them as MP3 streams[3]. Even though these interviews were not originally intended as a "Wizard of Oz" experiment, in the following we will interpret them so. A **"Wizard of Oz" experiment** is a research experiment in which subjects interact with a computer system that subjects believe to be autonomous, but which is actually being operated or partially operated by an unseen human being (see [Wik09]). 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 the *different qualities of knowledge* in 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 ...*" was answered by the expert with up to six of the following **explanation types**, the occurrence rate of which relative to the number of knowledge items is listed in the brackets:

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.

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

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.

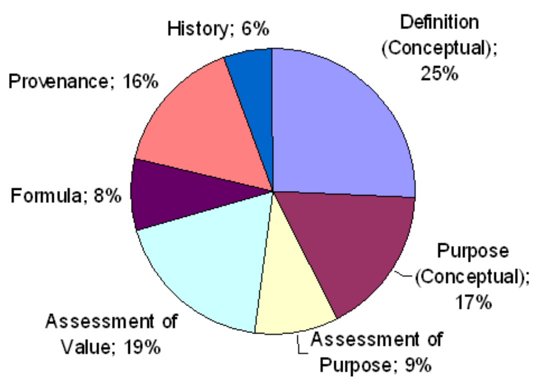


Figure 4: Help Needed — But Where?

Figure 4 portrays the distribution of occurrences according to each type. The "Wizard of Oz" experiment interpretation suggests that figure 4 showcases the user requirements for `SACHS` as a user assistance system (see also [NW06]). In particular, we can now *evaluate the `SACHS` system* with respect to this figure. Unsurprisingly, Definition explanations were the most frequent ones. Indeed, the `SACHS` system addresses this explanation type either with the theory graph-based explanation interface in figure 2 or the direct help text generator shown in figure 3. But the next two types are not covered in the `SACHS` system, even though it can be argued that the ontology-based `SACHS` architecture is well-suited to cope with Purpose explanations — indeed, some of the purpose-level explanations have erroneously found their way into `SACHS` definitions, where they rather should have been classified as 'axioms and theorems' (which are currently not supported by the `SACHS` interface). The next 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 is completely missing from `SACHS` as well as Assessment of Value. Explanations of type Formula are only rudimentarily covered in `SACHS` by virtue of being a plugin that inherits the formula bar from its host application Excel, which has some formula explanation functionality. Finally, the explanation type History is also not yet covered in `SACHS`.

To summarize the situation: Excel is able to give help for 8% of the explanations we found in the help of a human expert. The implemented `SACHS` system bumps this up to 33%, while the specified `SACHS` system can account for 50%. Even though this is certainly an improvement, it leaves much more to be desired than we anticipated. In particular, we draw the conclusion that background knowledge that 'only' contains a domain ontology is simply not enough.

We will try to remedy parts of this in the remainder of this paper. In particular, 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 Modelling Assessment

A first-hand approach of complementing spreadsheets with assessment knowledge could be the inclusion of Assessment of Value information into the definition text itself. In the concrete `SACHS` ontology we felt that we had no other choice in order to convey as much knowledge as possible, it is ontologically speaking a very impure approach (hence wrong) 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.

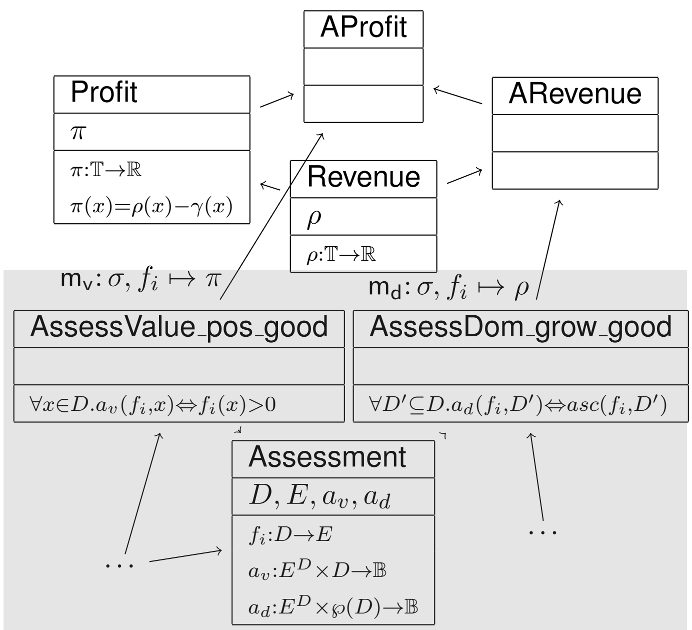
So before we address the question of how to model assessment, first we have to take a closer look at assessment itself: What is it about? Assessments consist of value judgements passed on situations modeled by (parts of) spreadsheets. As such, we claim that assessments are deeply in the semantic realm. 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, KK09d] 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.

4.1 Assessment via Theories and Morphisms

Consider the partial theory graph in figure 5, 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.



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 5: 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 judgemental 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.

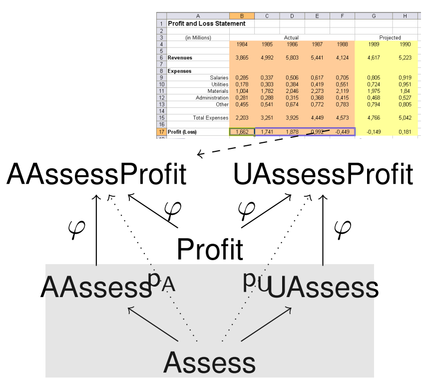
In our example we have restricted ourselves to unary functions, but of course it is very simple to provide assessment theories for any arity that occurs in practice. Moreover, we have only used assessment theories that only refer to inherent properties of the intended functions (e.g. being monotonically 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.

4.2 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 Example V 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 6, 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.



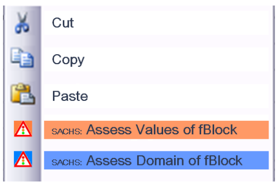
$\varphi := \sigma, f \mapsto \pi$ and σ as in Figure 5

Figure 6: Multi-Context Assessment

In the framing-based user interface put forward in [KK09d] we use theory morphisms 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 $\mathsf{p_A}$ 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 The Envisioned Assessment Extension in SACHS



We will now show how assessments 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 left 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.

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.

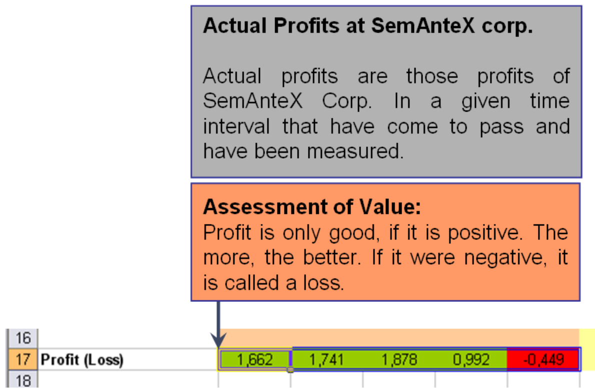


Figure 7: Assess the *Values*

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 7 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 8.

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.

Note that this analysis is anchored to the cell: Figure 9 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 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.

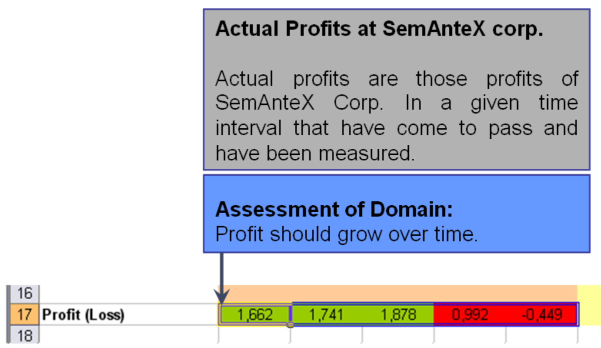


Figure 8: Assess the *Domain*

6 Conclusion and Further Work

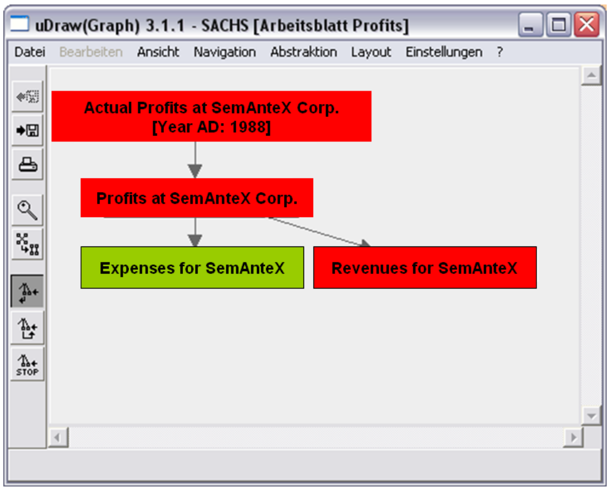


Figure 9: Assess *all* Values

In this paper we have reported an evaluation of the SACHS system, a semantic help system for a financial controlling system, via a (post-facto) "Wizard of Oz" experiment. The immediate results of this are twofold. The experiment basically validates the Semantic Illustration approach implemented in the SACHS system: The availability of explicitly represented background knowledge resulted in a dramatic increase of the explanations that could be delivered by the help system. But the experiment also revealed that significant categories of explanations are still systematically missing from the current 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 seems somewhat unnatural, intractable and largely empty. 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 in this paper. It seems that we can model Purpose and Assessment of Purpose explanations with a similar construction as the one proposed in section 4: We start out with a base assessment theory which provides an assessment function like a_v, which acts on a generic intended function f_i of the functional block in question, but instead of mapping into Boolean values, it 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.

But there is also another avenue for further research: We have not made full use of the data from the "Wizard of Oz" experiment in section 3. In figure 10 we compute the correlations between the explanation types. The co-occurrences seem particularly interesting: as Definition is the dominating type, then the others occur relatively infrequently (from 17.9% to 50%) in the first group and the bar

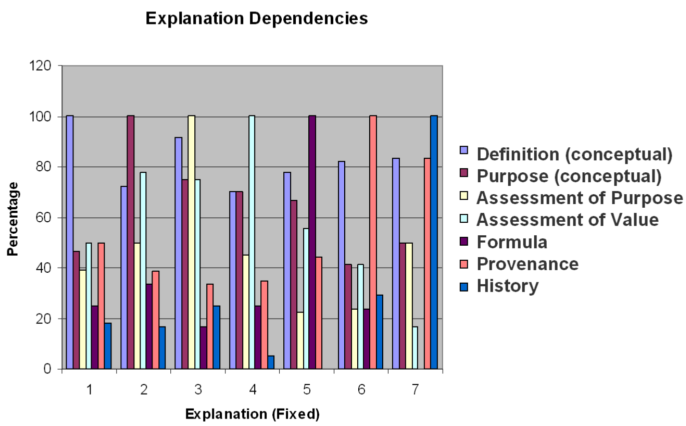


Figure 10: Explanation Dependencies

for Definition is relatively constant in the other clusters. The only exception to this is in the Assessment of Purpose cluster, where the co-occurrence is unusually high. Another interesting observation is that for all explanation types the co-occurrence with the Definition level is highest — except for the Purpose level. Here, the Assessment of Value statements appear more frequently than the ones of type Definition.

It seems that the distribution in figure 10 might tell us something about aggregation of explanation types in help systems. To make progress on this we might try to ask: *"Given an explanation on some level, then what else knowledge is needed or useful (according to an expert)?"*. In the absence of a criterion for differentiating between necessary knowledge and voluntarily supplied knowledge in our experiment, we might take the fact that a co-occurrence above 60% seems to be an obvious critical amplitude in this tabulation as an indicator that two explanation types are 'needed or useful' for each other.

We plan to study these relationships further; if these can be corroborated in other studies and other spreadsheet-based applications, then we will fine-tune our text aggregation algorithm for the dependency graph interface in figure 2 to volunteer the experimentally correlated explanation types.

Finally, we observe that the Semantic Illustration paradigm is neither restricted to the system Excel nor to the financial controlling domain. Unfortunately, the discussion and its consequences are beyond the scope of this paper, but was carried out elsewhere.

References

[Abr05] Robin Abraham. Template parsing with user feedback. In *VL/HCC*, pages 323–324. IEEE Computer Society, 2005.

[BP08] Richard Brath and Michael Peters. Spreadsheet validation and analysis through content visualization. *CoRR*, abs/0803.0166, 2008.

[CDSCW09] Jacques Carette, Lucas Dixon, Claudio Sacerdoti Coen, and Stephen M. Watt, editors. *MKM/Calculemus Proceedings*, number 5625 in LNAI. Springer Verlag, July 2009.

[Din09] Matthew Dinmore. Documenting problem-solving knowledge: Proposed annotation design guidelines and their application to spreadsheet tools. *CoRR*, abs/0908.1192, 2009.

[HM08] Karin Hodnigg and Roland T. Mittermeir. Metrics-based spreadsheet visualization: Support for focused maintenance. *CoRR*, abs/0809.3009, 2008.

[KK06] Andrea Kohlhase and Michael Kohlhase. Communities of Practice in MKM: An Extensional Model. In Jon Borwein and William M. Farmer, editors, *Mathematical Knowledge Management (MKM)*, number 4108 in LNAI, pages 179–193. Springer Verlag, 2006.

[KK09a] Andrea Kohlhase and Michael Kohlhase. Compensating the computational bias of spreadsheets with MKM techniques. In Carette et al. [CDSCW09], pages 357–372.

[KK09b] Andrea Kohlhase and Michael Kohlhase. Modeling task experience in user assistance systems. In Mehlenbacher et al. [MPWS09], pages 135–142.

[KK09c] Andrea Kohlhase and Michael Kohlhase. Semantic transparency in user assistance systems. In Mehlenbacher et al. [MPWS09], pages 89–96.

[KK09d] Andrea Kohlhase and Michael Kohlhase. Spreadsheet interaction with frames: Exploring a mathematical practice. In Carette et al. [CDSCW09], pages 341–356.

[MGM+08] Luc Moreau, Paul Groth, Simon Miles, Javier Vazquez, John Ibbotson, Sheng Jiang, Steve Munroe, Omer Rana, Andreas Schreiber, Victor Tan, and Laszlo Varga. The provenance of electronic data. *Communications of the ACM*, 51(4):52–58, 2008.

[Mic89] Microsoft. Ms excel. Online (`http://office.microsoft.com/en-us/excel/default.aspx`), 1989. Accessed on 2009-07-27.

[MPWS09] Brad Mehlenbacher, Aristidis Protopsaltis, Ashley Williams, and Shaun Slatterey, editors. *Proceedings of the 27th annual ACM international conference on Design of communication (SIGDOC)*, New York, NY, USA, 2009. ACM Special Interest Group for Design of Communication, ACM Press.

[Mur08] Simon Murphy. Spreadsheet hell. *CoRR*, abs/0801.3118, 2008.

[NW06] David G. Novick and Karen Ward. What users say they want in documentation. In *SIGDOC'06 Conference Proceedings*, pages 84–91. ACM, 2006.

[Pan00] Raymond 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.

[Wik09] Wikipedia. Wizard of oz experiment — wikipedia, the free encyclopedia, 2009. [Online; accessed 20-May-2009].

[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, 2006.