Managing Knowledge for Assessing Spreadsheet-based Models

Alexandru Toader[1] Michael Kohlhase[1] Andrea Kohlhase[2]

[1] Computer Science, Jacobs University Bremen
[2] University of Applied Sciences, Neu-Ulm

Abstract. Spreadsheets are tools to model personal, business and scientific data and to support decision-making based on such models. Moreover, spreadsheets are used for communicating data, models, and decisions, which makes the interpretability of spreadsheet intents an important consideration in the design of spreadsheet interaction.
In this paper, we show how MKM methods can be used to support users in the problem of model assessment, i.e., the process of passing judgements on a situation based on modeled spreadsheet data. We introduce a framework for specifying, documenting and personalizing assessments semi-formally and a set of plugins for the Semantic Alliance Architecture.

1 Introduction: Spreadsheets as Decision-Support Systems

Spreadsheets are used to model quantitative real life scenarios and entire decision making processes have been built around them. In [BSH02] they are even defined as "*software products that help users apply analytical and scientific methods to decision making*". Here we take up this view and consider them special **decision support systems (DSSs)**. So far though, spreadsheet functionality does not directly support this quality, it rather assists indirectly by realizing calculations and enabling reports necessary for analyzing a situation and making decisions built on these. Thus, extending spreadsheet software to directly support its DSS quality would lower the hurdle and make DSS available to even more users, businesses, and institutions. We call systems that use spreadsheets for modeling and support the decision process **spreadsheet-based decision support systems (SSDSS)**, see [ŞA08; TR14]. In a nutshell, an SSDSS assists users with two tasks: *i*) deciding whether a particular aspect of a model is favorable to the business goal, and *ii*) finding remedies if it is not. We call the first task **assessment** and we will concentrate on supporting it in SSDSS in this paper.

We can see assessment in spreadsheet documents as just another facet of the interpretation process of tabular data with respect to the situation modeled by the electronic ledger. Interpretation is a source of errors[3] that has been often neglected although it is of great importance in the case of long-lived spreadsheets. This is especially virulent in cases where the reader and the creator of a

[3] For general studies of spreadsheet errors see[TT97; RCK08].

spreadsheet are different individuals, the reader is unfamiliar with the domain of the spreadsheet, and not all the knowledge of the problem domain is transferred along with the document – a standard situation, when spreadsheets are used as a means of communication:

An indication of this is given in a study that categorizes the help queries of users of a financial controlling spreadsheet in a large public/private research institution (see Figure 1). [KK13] reports that assessment-related questions make up 28% of the required explanation types.

This paper aims to reduce the occurrence of interpretation errors by enhancing the spreadsheet user's assessment capabilities.

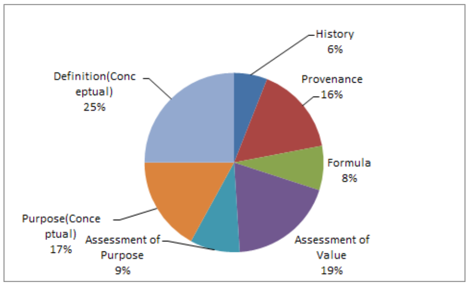


Fig. 1. Types of Help Requests

We build on the theoretical analysis in [KK13] and report on a concrete implementation by the first author in the scope of his B.Sc thesis [Toa14] based on the "Semantic Alliance Framework" developed in our research group [Dav+12; Juc14]. This gives us a DSS that combines the ease and simplicity of modeling business data in spreadsheets with a knowledge-based assessment service.

In Section 2 we briefly review the concept "assessment", in Section 3 the Semantic Alliance Framework, and in Section 4 we present the assessment service itself. Section **??** evaluates the services via a use case, and Section 6 concludes the paper and discusses future work.

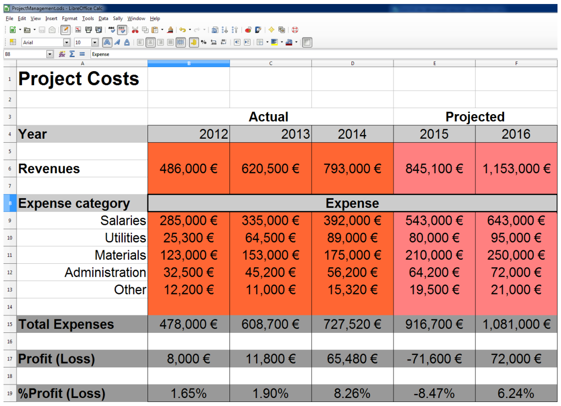
2 Assessment in Spreadsheets

Before devising an assessment service for spreadsheets, it is important to understand what assessment of spreadsheet data is and how it fits into the user's workflow so that we can derive requirements for our solution. To ease reading we will not point out any longer that we are specifically concerned with assessment of spreadsheet data and in a spreadsheet context.

Assessment consists of value and purpose judgments passed on situations modeled by the spreadsheet document or parts of it. Indeed, [KK13] classifies assessment queries in Figure 1 into two categories:

- *Assessment of Value* refers to the ability of a user to make judgments on the spreadsheet data after having interpreted the concrete values.
- *Assessment of Purpose* refers to the ability of a user to draw the proper conclusions or perform the correct actions based on the content of a spreadsheet document.

Running Example To fortify our intuition about SSDSS consider the simple spreadsheet in Figure 2, which will serve as a running example for this paper. It is used to model the evolution of a small department of a multinational company over a number of years. It acts as a controlling tool on which all present and future decisions are based, thereby qualifying as a simple SSDSS.


ProjectManagement.ods - LibreOffice Calc
File Edit View Insert Format Tools Data Sally Window Help
Arial
10
B8
Expense
A
B
C
D
E
F

	A	B	C	D	E	F
1	Project Costs					
2						
3			Actual		Projected	
4	Year	2012	2013	2014	2015	2016
5						
6	Revenues	486,000 €	620,500 €	793,000 €	845,100 €	1,153,000 €
7						
8	Expense category			Expense		
9	Salaries	285,000 €	335,000 €	392,000 €	543,000 €	643,000 €
10	Utilities	25,300 €	64,500 €	89,000 €	80,000 €	95,000 €
11	Materials	123,000 €	153,000 €	175,000 €	210,000 €	250,000 €
12	Administration	32,500 €	45,200 €	56,200 €	64,200 €	72,000 €
13	Other	12,200 €	11,000 €	15,320 €	19,500 €	21,000 €
14						
15	Total Expenses	478,000 €	608,700 €	727,520 €	916,700 €	1,081,000 €
16						
17	Profit (Loss)	8,000 €	11,800 €	65,480 €	-71,600 €	72,000 €
18						
19	%Profit (Loss)	1.65%	1.90%	8.26%	-8.47%	6.24%

Fig. 2. A Project Management Spreadsheet

Let us use Figure 2 to introduce an important concept: We call a rectangular fragment of a spreadsheet a **functional block**, iff it corresponds to a function in the modeled situation. For instance, cells [B6:D6] model the expenses as a function on the years 2012-2014. Cells and functional blocks are the basic building blocks of spreadsheet models and therefore the natural domains of assessment.

Assessment is Intensional The following can be considered typical assessment statements in the context of Figure 2:

i) "*Row [6] looks good.*"
ii) "*The revenues look good.*"
iii) "*I like this* [points to cell [D17]] *but that* [points to cell [E17]] *is a disaster.*"
iv) "*I like the profit/loss in 2014 but of course not the expected one in 2015.*"

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 65,480 and -71,600 (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 2014/5" rather than the extension. Another way to view this is that the latter two assessments are about the argument/value pairs 2014/65,480 and 2015/-71,600.

We will make this view the basis of our treatment of assessment: We need to have a model of spreadsheets, which can capture the intensional level, so that we can assess it. In the next section (part 3.1) we will present spreadsheets with a structured background ontology that captures the spreadsheet intention, and in 4 we show how we can we can extend that for assessing spreadsheet values and functional properties.

Assessment is Personal But there is another aspect of assessment we need to model: assessments depend on context and who assesses the situation. Indeed example *v*) below shows a statement relating assessments by three stakeholders – arguably enabling the addressee of the statement to form their own assessment from that.

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.*"

In particular, we cannot tie assessments rigidly to particular cells and functional blocks, but need a setup, where assessments are objects in their own right that can be "applied" to cells and functional blocks. We consider these "assessment objects" as reified background knowledge about values and functions of the model expressed in the spreadsheet.

Assessment is Computational Finally, we note the obvious: An assessment is computed on the values in the respective cell or functional block – according to the specification in the "assessment object" we have stipulated above.

All of these requirements do not fit with the conventional approach to extending spreadsheets by macros in the scripting language of the application (e.g. .Net for Excel or Java for LibreOffice). Instead we make use of a semantic framework we have previously developed and which we will present now.

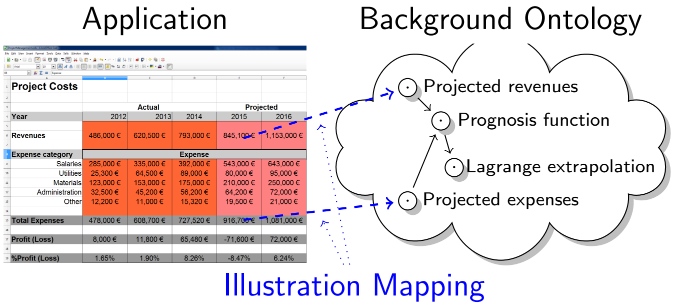
3 The Semantic Alliance Framework

The entry point for the Semantic Alliance Framework we use as a basis for our Assessment Service is the realization, that spreadsheets per se only represent the data and computation layers of a model, and leave the intended meaning and the background knowledge about the situation modeled in the spreadsheet implicit – usually hinted at in the "legend" cells – see [KK13] for details.

3.1 Semantic Illustration

To remedy this shortcoming, we give the spreadsheet creators the opportunity to document the background knowledge – not in a human-readable "manual", but in a machine-actionable **structured background ontology**[4]. The background ontology concepts themselves can be linked to the meaning-carrying fragments of the spreadsheets. Figure 3 shows the approach.

[4] We will use the term "ontology" in its broad meaning, where it stands for "a representation of the objects in a given situation, and their relations to each other".



Project Costs					
		Actual		Projected	
Year	2012	2013	2014	2015	2016
Revenues	486,000 €	620,500 €	793,000 €	845,100 €	1,153,000 €
Expense category			Expense		
Salaries	285,000 €	335,000 €	392,000 €	543,000 €	643,000 €
Utilities	25,300 €	64,500 €	89,000 €	80,000 €	95,000 €
Materials	123,000 €	153,000 €	175,000 €	210,000 €	250,000 €
Administration	32,500 €	45,200 €	56,200 €	64,200 €	72,000 €
Other	12,200 €	11,000 €	15,320 €	19,500 €	21,000 €
Total Expenses	478,000 €	608,700 €	727,520 €	916,700 €	1,081,000 €
Profit (Loss)	8,000 €	11,800 €	65,480 €	-71,600 €	72,000 €
%Profit (Loss)	1.65%	1.90%	8.26%	-8.47%	6.24%

Fig. 3. The Semantic Illustration Architecture

In our system the background ontology is represented in the OMDoc [Koh06] format, a markup format and data model for semi-formal, structured document collections. OMDoc provides a strong, logically sound module system based on structured theory graphs [RK13]. In contrast to other ontology modeling languages like OWL, the OMDoc format does not commit to a formal logical language. Therefore, it lacks a native concept of inference, but also does not force the author to fully formalize the ontology and to work around the expressivity limitations of the underlying logical system. Instead, OMDoc allows to locally formalize elements - and thus provide partial inference - with whatever formal system is most suitable.

After choosing a representation format for the ontology, the biggest challenge left is how to integrate it with the document and make it usable by the associated services. In [Eri07] Eriksson suggested to combine documents and ontologies by "*adding annotations to electronic-documents formats and including the ontologies in electronic documents*" yielding semantic documents. This approach limits the reusability of the ontology for multiple documents that contain the same concepts and poses a problem for maintaining and enhancing the information stored in the ontology. Enhancing the spreadsheet application with functionality to use a local ontology is prohibitively expensive. Therefore, an external ontology that can be used by multiple services and documents seems more sensible and is the approach taken by the Semantic Alliance Framework which will support the Assessment Service. This approach was materialized as the *Semantic Illustration Architecture* [KK13].

We will now give a brief overview of the Semantic Alliance Framework and explain how it supports semantic services like the Assessment Service.

3.2 Semantic Alliance Framework

The Semantic Alliance Framework is an architecture and software framework for semantic allies [5]. It combines an invasive approach that allows users to profit

[5] Semantic allies are semantic systems that complement existing software applications with semantic serviices and interactions based on a background ontology.

from semantic technology without leaving their accustomed workflows and tools with an application-independent way of extending applications with knowledge management technologies.

The Semantic Alliance Framework provides the foundation for the development of the Assessment Service. As we will see later on, it offers functionality to tap into the user's interactions with the document and to provide meaningful services in the appropriate context while giving the user the impression of (almost) perfect integration with the host application. The Semantic Alliance Framework works by mashing up the GUIs of the knowledge management systems and applications themselves. The Semantic Alliance Framework has three components (see Figure 4):

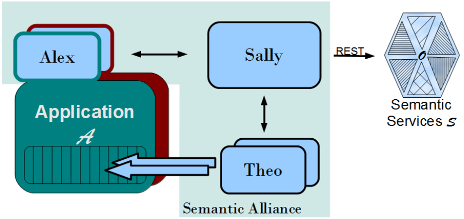


Fig. 4. The Semantic Alliance Framework

- **Sally:** The main component of the mashup enabler, integrates the functionality of the application $\mathcal{A}$ (e.g., spreadsheet application) and the semantic services $\mathcal{S}$ (e.g., the Assessment Service) into a joint user interface and interaction model.
- **Alex:** The application $\mathcal{A}$ is extended by a slim API (Application Programming Interface) "Alex" that reports and executes relevant user interactions within $\mathcal{A}$ (e.g., cell clicks in spreadsheets) to and from **Sally**.
- **Theo:** A screen-area manager that enables invasive design. Content supplied by $\mathcal{S}$ is embedded as a pop-up into the GUI of $\mathcal{A}$.
- **Semantic Services:** are independent components which merge and process information from multiple sources including the machine-actionable ontology, the application $\mathcal{A}$ and external sources. They return the result to the user through **Sally** (e.g., The Definition Service, which is offered through the Semantic Alliance Framework, when invoked returns the definition of a concept associated with a spreadsheet cell. For instance, when the user asks for the definition of cell [E15], he gets the definition of the concept *Projected Expenses*.)

Note that the Semantic Alliance Framework relies on the Semantic Illustration architecture – the machine-actionable background ontology is the basis for the semantic services and the illustration mapping (the green dashed arrows in Figure 3) show which parts apply at a given point in the spreadsheet. But from the point of view of the Semantic Alliance Framework, the background ontology is encapsulated into the semantic (web) services reducing the knowledge integration problem to a mashup of (web) services (this is what the Sally component does).

4 The Value Assessment Service

In this section we will give a brief overview of how the Assessment Service is integrated into the Semantic Alliance Framework, describe the interaction between the user and the service, and discuss about the representation and computation of assessments with theory graphs. An Assessment Service differs from – say – a Definition Lookup Service which only relays and aggregates parts of the background ontology, in that it has to *compute* a verdict about possibly changing spreadsheet data based on the background ontology and feed it back to the user on the fly in an intuitive way.

Representing/Computing Assessments in Theory Graphs The Semantic Alliance Framework organizes the background ontology into **theories** – collections of concept definitions together with statements of properties of and relations between the objects denoted by these concepts. Theories are connected by **views**, that is, meaning-preserving mappings that allow to interpret objects from the source theory as objects of the target theory. The **theory-graph** paradigm in OMDoc/MMT [Koh06; RK13] gives us a powerful, modular/object-oriented representation framework for formal and informal mathematical knowledge. Crucially, OMDoc/MMT allows for **parametric theories** and embedded computation in the MMT web service [Rab13]: Representing assessment knowledge in (collections of) theories that can be connected to cells and functional blocks in spreadsheets meets the requirement of "object-like" assessment knowledge from subsection 2. The object-oriented/parametric nature allows us to make general theories of "assessment patterns" and specialize them in the illustration mapping as the Sally component that operationalizes it also contains an "abstract spreadsheet model" that can store concrete parameters. Details of the background ontology used for the running example can be found in Section 5.

Integration into the Semantic Alliance Framework The Assessment Service is just one of the many semantic services [Dav+12] offered through the Semantic Alliance Framework. For the purpose of this example, we will use LibreOffice Calc as our host spreadsheet application and the document illustrated in Figure 2 as our running example.

The service registers itself with the framework at initialization and is available to the user each time he tries to assess a semantic object that is linked to an assessment rule.

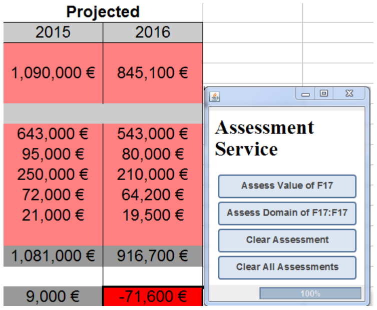
When the Assessment Service is invoked on a cell range, it collects information from the spreadsheet application regarding the selected object, merges it with knowledge from the ontology, and it runs the assessment for the concrete values in the document as described above.

User Interaction The Assessment Service addresses both assessment needs: to be able to assess a value and the purpose. The former refers to an interaction on cell level, the latter requires an interaction concerning a functional block. If the service visualizes the assessment outcome, then the original layout should be regathered with a simple click. From the user's point of view, therefore, the Assessment Service has to support three operations:

- Assess a value,
- Assess a domain (a group of cells), and
- Clear assessment results (i.e., revive the original spreadsheet state).

Value Assessment To assess the value of cell [F17], the user right-clicks on the cell and selects the Assessment Service "Assess value of F17" from the list of services offered by the framework for that particular object (see Figure 5). At this point, the service mashes-up information about the selected object from the spreadsheet and the knowledge base, and returns the result of the associated assessment theory evaluated for the object.

The result of an assessment is shown to the user by highlighting the assessed cell or cell ranges in bright green for a positive result, and in bright red for a negative result. At this point, LibreOffice locks down the sheets containing the colored cells so that no modification is possible. We can see the result of assessing the projected profit for the year 2016 in Figure 5.



Projected	
2015	2016
1,090,000 €	845,100 €
643,000 €	543,000 €
95,000 €	80,000 €
250,000 €	210,000 €
72,000 €	64,200 €
21,000 €	19,500 €
1,081,000 €	916,700 €
9,000 €	-71,600 €

Fig. 5. Assessed Projected Profit

Domain Assessment The process of assessing an entire functional block is almost identical. After, for example, the range [B17:D17] was selected and the Assessment Service "Assess Domain of B17:D17" was invoked, the system checks if the selected domain represents one homogeneous functional block. If there are cells in the block that are associated with a different concept, i.e., a different intended function, we cannot assess the block. If this sanity check passes, the process continues as in the case of the assessment of value scenario.

After performing an assessment the spreadsheet cannot be edited by the user. This locking is enforced to offer the user the possibility of returning the spreadsheet document to a pre-assessment state. In particular, this enables the "Clear Assessment" and "Clear All Assessments" options of the Assessment Service (see Figure 5). After clearing all the results, the user can edit the document as before the Assessment Service was invoked.

Our Assessment Service provides a simple, intuitive interface that hides the formal theory structure and the evaluation of arbitrarily complex assessment rules. The theory graph that is the basis of the service can be easily extended with new assessment rules with a minimal implementation cost while incrementally increasing the value of the entire system.

5 Facilitating Modular Assessments via Theory Graphs

We now present the core of the modular assessment algorithm promised in Section 4: we want to compute a binary "good/bad" assessment for cell [E17] from Figure 2.

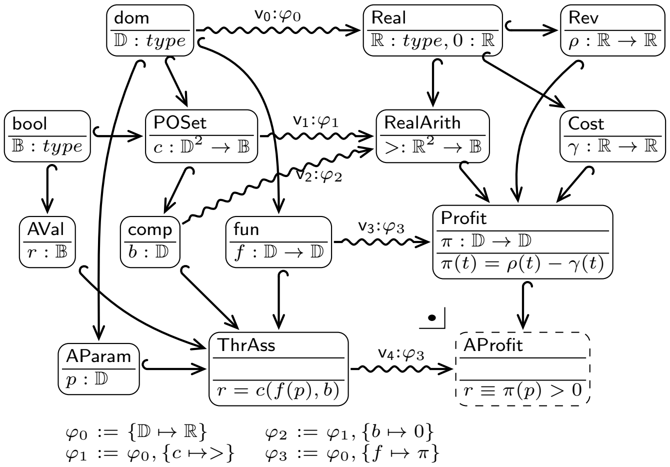

dom
$\mathbb{D} : type$
$v_0 : \varphi_0$
Real
$\mathbb{R} : type, 0 : \mathbb{R}$
Rev
$\rho : \mathbb{R} \to \mathbb{R}$
bool
$\mathbb{B} : type$
POSet
$c : \mathbb{D}^2 \to \mathbb{B}$
$v_1 : \varphi_1$
RealArith
$> : \mathbb{R}^2 \to \mathbb{B}$
Cost
$\gamma : \mathbb{R} \to \mathbb{R}$
$v_2 : \varphi_2$
AVal
$r : \mathbb{B}$
comp
$b : \mathbb{D}$
fun
$f : \mathbb{D} \to \mathbb{D}$
$v_3 : \varphi_3$
Profit
$\pi : \mathbb{D} \to \mathbb{D}$
$\pi(t) = \rho(t) - \gamma(t)$
$\bullet\rfloor$
AParam
$p : \mathbb{D}$
ThrAss
$r \equiv c(f(p), b)$
$v_4 : \varphi_3$
AProfit
$r \equiv \pi(p) > 0$
$\varphi_0 := \{\mathbb{D} \mapsto \mathbb{R}\}$
$\varphi_1 := \varphi_0, \{c \mapsto >\}$
$\varphi_2 := \varphi_1, \{b \mapsto 0\}$
$\varphi_3 := \varphi_0, \{f \mapsto \pi\}$

Fig. 6. A Theory Graph for Value Assessments

5.1 A Theory Graph for Assessment

In Figure 6, we see a theory graph of assessment theories. The upper two rows are occupied by a standard development of partially ordered sets (POSet) over a domain $\mathbb{D}$, that is instantiated by the real numbers via a view v_i with assignment φ_1. In lower left hand side we work towards a "threshold-based assessment theory" ThrAss which combines generic symbols b for a basis of comparison, a function f to be assessed a parameter p where it is. Here the symbol r introduced by the theory Aval ("value of Assessment") is used for recording the assessment result. Theory ThrAss specifies the assessment intuition that we can assess the value of a function f at a point p by comparing it to a threshold b in the axiom $r \equiv c(f(p), b)$.

Finally, on the far upper right, we define a theory Profit, which introduces a profit function π as the pointwise difference of the revenues function ρ and the costs function γ[6].

[6] In this exposition we have glossed over many details in the model, for instance we use the abstract domain $\mathbb{D}$, even though spreadsheet values can be strings, numbers, booleans, quantities, ... Similarly, we make π, ρ, and γ real functions, even though they should go from time intervals to (monetary) quantities. This simplification is only for expository reasons and does not preclude the central mechanism.

The point of the whole construction is the theory AProfit ("assessment of profit") in the lower right hand corner. It is the category-theoretic pushout (indicated by $\lrcorner$ in the diagram) of the views φ_1, φ_2, and φ_3. Such pushouts always exist in MMT and AProfit can be computed as the union of Profit and the image of ThrAss under the morphism $\varphi_4 := \varphi_1 \cup \varphi_2 \cup \varphi_3$. In MMT, we can specify this by the theory expression declaration (see [Rab15] for details)

AProfit $=$ Profit with ThrAss[φ_1,φ_2,φ_3]

Thus AProfit gives access to the definition of π as well as the computational definition $r \equiv \varphi_4(c(f(p), b)) \equiv \pi(p) > 0$, which is at the heart of the assessment of profit. Note that the construction of AProfit is parametric in the choice of parameters in the assessment: We use

1. a threshold-based assessment (ThrAss),
2. a lower threshold (φ_1 assigns c to $>$),
3. the real number 0 (as specified by φ_2) as a threshold

The morphism v_3:$f \mapsto \pi$ with source is given determined by the illustration mapping which interprets the functional block of cell [D17] as a profit by linking it to the theory Profit.

Note that we have established a theory in which assessments can be computed, but we have not actually assessed cell [E17] yet, since the profit function π and the point p of assessment are still generic. We can iterate the construction above with the MMT declaration

AE17 $=$ AProfit[π:=Π,p:=2015]

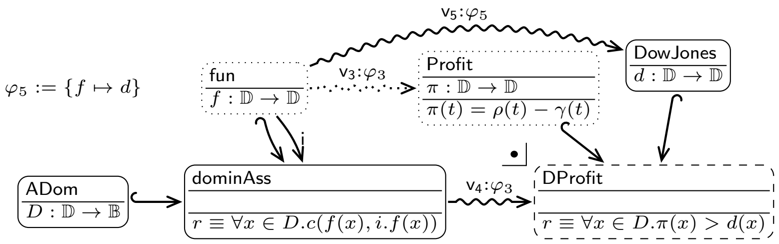
where Π is the concrete function $\{2015 \mapsto -71,000\text{€}, \ldots\}$ from the functional block of [E17]. We are making use of a recent extension of MMT by literal values (here machine numbers [Rab]). For computation with finite functions, we introduce a straightforward representation theory and introduce special rewrite rule for the application of finite functions, which the MMT rewriter can simplify.

As all of the information needed for this declaration are in the spreadsheet (and can be supplied by the spreadsheet Alex), we can generate it on the fly and avoid polluting the theory graph with spreadsheet data. In the theory AE17 we can now compute $r \equiv \Pi(2015) > 0 \equiv -71,000 > 0 \equiv \mathsf{F}$. So we can assess the value of cell [E17] as "bad". This justifies the red color in Figure 5.

The actual computation for the assessment is shared between the MMT API [Rab13], which computes the formulae in AE17 and the Universal OpenMath Machine [KMR13], which implements arithmetic simplifications.

5.2 Other Assessments

Other threshold assessments can reached by selecting different parameters, e.g. if the company of our running example were a tax evasion scheme, we would assess profits as "good", if they are sufficiently negative by adding a morphism $\varphi_5 := \{c \mapsto <, b = 50,000\}$ to the theory graph in Figure 6 and using that instead of φ_2.



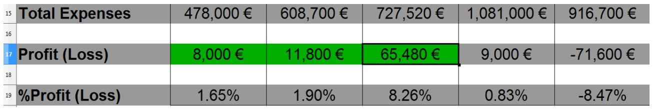
$\mathsf{v}_5 \colon \varphi_5$
$\varphi_5 := \{f \mapsto d\}$
fun
$f : \mathbb{D} \to \mathbb{D}$
$\mathsf{v}_3 \colon \varphi_3$
Profit
$\pi : \mathbb{D} \to \mathbb{D}$
$\pi(t) = \rho(t) - \gamma(t)$
DowJones
$d : \mathbb{D} \to \mathbb{D}$
i
ADom
$D : \mathbb{D} \to \mathbb{B}$
dominAss
$r \equiv \forall x \in D.c(f(x), i.f(x))$
$\mathsf{v}_4 \colon \varphi_3$
DProfit
$r \equiv \forall x \in D.\pi(x) > d(x)$

Fig. 7. Figure 6 + Domain Assessment

But with a little more effort we can also build assessments that are not threshold assessments. Say we want to assess a profit if it grows stronger than the Dow Jones index, then we can extend the theory graph in Figure 6 with the one in Figure 7, where fun, Profit, and φ_3 are the theory from Figure 6 and drawn dotted. Theory DowJones introduces a function d for the Dow Jones index and we get the assessment theory DProfit as

$$\mathsf{DProfit} = \mathsf{Profit\ with\ domainAss}[\varphi_1,\varphi_2,\varphi_5]$$

As above we can assess the the profit over a whole range of years by further instantiating D and visualizing via the red/green semaphore metaphor as above. If we choose $D := \{2012, 2013, 2014\}$ we obtain the situation in Figure 8.



15	Total Expenses	478,000 €	608,700 €	727,520 €	1,081,000 €	916,700 €
16						
17	Profit (Loss)	8,000 €	11,800 €	65,480 €	9,000 €	-71,600 €
18						
19	%Profit (Loss)	1.65%	1.90%	8.26%	0.83%	-8.47%

Fig. 8. Assessed Actual Profit Block

A strong point of our theory-graph-based approach is that it can handle different assessments – even adversarial ones – simply by associating different theories and views; composition and computation are supplied uniformly by the OMDoc/MMT framework.

5.3 Towards Knowledge Management for Assessments

The theory graph developments above show that the knowledge behind assessments can be expressed succinctly, declaratively, and modularly in form of formal/computational MMT theories. And the setup scales well with the addition of new assessment methods: inspection of the graphs in Figures 6 and 7 reveals that the theories involved are needed for different purposes:

1. general mathematical background (bool, POSet, Real, RealArith)
2. domain knowledge (Rev, Cost, Profit)
3. the basic setup of the assessment method (dom, AVal, fun),

4. the value/domain assessment aspects (AParam/ADom)
5. threshold assessment (comp, ThrAss)
6. domain assessment (domainAss)

The first four categories come with the assessment approach and can be reused for new *general assessment methods*, which comprise one theory with the respective computation formula and their parameter theories. The *concrete assessment theories* (AProfit, DProfit) can be obtained from these by establishing a handful of views and one line theory definitions.

Note that even so, we cannot assume that the theories of the general assessment methods are contributed by spreadsheet authors and users. Like the domain/math knowledge, they have to be set up by knowledge engineers who are well versed in theory graph authoring. But in our (admittedly limited) experience assessment is governed by a tractable number of general assessment methods that correspond to a limited number of assessment stereotypes.

Given these, the spreadsheet author/user only needs to bind a functional block to a general assessment method and instantiate that with the appropriate views. This task can be supported by a suitable user interface: We can make use of the fact that as a flexiformal format OMDoc/MMT theories can include their own documentation – informal/natural language counterparts to the formal/-computational content, which can be presented to the user by a **Theo**. Once a general assessment method AM is chosen, we can prompt the user for values of the parameters in AM, to establish the necessary views. This again amounts to a choice from the background ontology, which can be restricted – and thus made more tractable – by semantic (e.g. type/domain) constraints.

With such a knowledge management interface for assessments, we can hide all the theory graph mechanics from the spreadsheet author/user, who only needs knowledge about the domain and the spreadsheet itself (obviously) to create new assessments.

6 Conclusions & Future Work

We have presented a knowledge-based approach to assessment in spreadsheets and reported on our experiences of implementing an assessment service in the Semantic Alliance Framework. Given a small set of carefully constructed "assessment theories" (constructed by OMDoc/MMT specialists) and a larger set of theories that specify the background knowledge about the domain modeled by the spreadsheet (created by domain specialists), individual assessment views can be set by linking to specific theories via the illustration mapping (set up by each user). From these the MMT web service can compute assessments and visualize them to the spreadsheet user. Extended by assessment knowledge, spreadsheet-based models can in turn act as decision support systems.

We have a prototypical implementation of the system and background ontologies for a small number of spreadsheets. To scale the system beyond an initial prototype much remains to be done. Work is currently under way to support authoring of OMDoc ontologies and better understand knowledge management issues involved. Once those issues are solved, we can conduct user evaluations. Instead of detailing this let us assess the potential by exploring what added-value services can be built on assessment ontologies.

One important part of assessment we have not addressed is the answer to the multifaceted question "*Why is this assessment positive/negative?*". One the one hand, it is concerned with how the assessment theory is defined, but on the other hand, it is concerned with the factors that determined the assessment to have a positive or negative result. The following two envisioned services will attempt to tackle both issues.

Assessment Explanation To explain an assessment result the user has to understand the assessment theory behind it. As the computation in the MMT web service is inference-based, we can generate explanations of every step of the computation and the assessment. In fact, it should be relatively simple to generate assessment explanations specific to the particular cell or functional block, since the MMT API has access to all parameter instances.

Assessment of Dependencies Recall the two tasks of a DSS mentioned in the introduction. With the assessment service presented in this paper, we have only addressed the first one. Finding remedies based on the assessments still remains unaddressed.

It is conceivable that the information in the assessment can be used for pinpointing causes of negative assessments and thus problems the remedies might attack.

Consider for instance the cell [F17] in the running example, which represents the Projected Profit for the year 2016. The Projected Profit is defined as "*Projected Revenues - Projected Total Expenses*". It makes sense to trace the assessments through the dependency tree in an attempt to gather more insight about the assessment's result.

In Figure 9 we can see the definition graph for the negatively assessed cell [F17] for the Projected Profit in year 2016. The node associated with Projected Revenue for the year 2016 is colored in red, meaning that it was also negatively assessed, so the problem might be with the revenues.

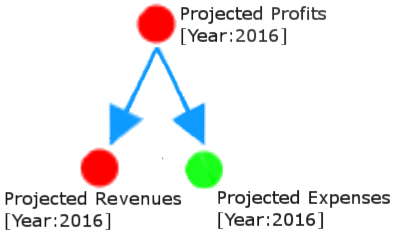


Fig. 9. Assessment of Dependencies

The JavaScript InfoVis Toolkit [Jit] provides all the necessary functionality to visualize the dependency graph and the Assessment Service already provides functionality for assessing the individual cells and functional blocks. At this point, the only missing functionality to implement this service is a Formula Parser that would transform spreadsheet formulae into a tree structure, which is left for further research.

Acknowledgments We gratefully acknowledge that fruitful discussions with Mihnea Iancu and Florian Rabe about pushouts and theory expressions in MMT, and their extensions of the MMT API that made the modular computation of assessments possible. This work has partially been supported by the German Research Council under Grants KO 2428/11-1 and KO 2428/13-1.

References

[BSH02] Hemant K. Bhargava, Suresh Sridhar, and Craig Herrick. "Beyond Spreadsheets: Tools for Building Decision Support Systems." In: *IEEE Computer* 32.3 (Jan. 3, 2002), pp. 31–39. URL: http://dblp.uni-trier.de/db/journals/computer/computer32.html#BhargavaSH99.

[Car+13] Jacques Carette et al., eds. *Intelligent Computer Mathematics.* Lecture Notes in Computer Science 7961. Springer, 2013. DOI: 10.1007/978-3-642-39320-4.

[Dav+12] Catalin David et al. "Semantic Alliance: A Framework for Semantic Allies". In: *Intelligent Computer Mathematics.* Ed. by Johan Jeuring et al. LNAI 7362. Berlin and Heidelberg: Springer Verlag, 2012, pp. 49–64. URL: http://kwarc.info/kohlhase/papers/mkm12-SAlly.pdf.

[Eri07] Henrik Eriksson. "The semantic-document approach to combining documents and ontologies". In: *International Journal of Human-Computer Studies* 65.7 (2007). Knowledge representation with ontologies: Present challenges - Future possibilities, pp. 624 –639. DOI: 10.1016/j.ijhcs.2007.03.008.

[Jit] *JavaScript InfoVis Toolkit.* http://philogb.github.io/jit/. [Accessed on 21-April-2014].

[Juc14] Constantin Jucovschi. "Towards an Interaction-based Integration of MKM Services into End-User Applications". In: *Intelligent Computer Mathematics 2014.* Ed. by Stephan Watt et al. LNCS 8543. Springer, 2014, pp. 344–356. URL: http://arxiv.org/abs/1403.5058.

[KK13] Andrea Kohlhase and Michael Kohlhase. "Spreadsheets with a Semantic Layer". In: *Electronic Communications of the EASST: Specification, Transformation, Navigation – Special Issue dedicated to Bernd Krieg-Brückner on the Occasion of his 60th Birthday* 62 (2013). Ed. by Till Mossakowski, Markus Roggenbach, and Lutz Schröder, pp. 1–20. URL: http://journal.ub.tu-berlin.de/eceasst/article/view/870.

[KMR13] Michael Kohlhase, Felix Mance, and Florian Rabe. "A Universal Machine for Biform Theory Graphs". In: *Intelligent Computer Mathematics.* Ed. by Jacques Carette et al. Lecture Notes in Computer Science 7961. Springer, 2013. DOI: 10.1007/978-3-642-39320-4.

[Koh06] Michael Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. LNAI 4180. Springer Verlag, Aug. 2006. URL: http://omdoc.org/pubs/omdoc1.2.pdf.

[Rab] Florian Rabe. "Generic Literals". URL: http://kwarc.info/frabe/Research/rabe_literals_14.pdf.

[Rab13] Florian Rabe. "The MMT API: A Generic MKM System". In: *Intelligent Computer Mathematics*. Ed. by Jacques Carette et al. Lecture Notes in Computer Science 7961. Springer, 2013, pp. 339–343. DOI: 10.1007/978-3-642-39320-4.

[Rab15] Florian Rabe. "Theory Expressions (a Survey)". submitted. 2015.

[RCK08] Kamalasen Rajalingham, David R. Chadwick, and Brian Knight. "Classification of Spreadsheet Errors". In: *CoRR* abs/0805.4224 (2008).

[RK13] Florian Rabe and Michael Kohlhase. "A Scalable Module System". In: *Information & Computation* 0.230 (2013), pp. 1–54. URL: http://kwarc.info/frabe/Research/mmt.pdf.

[Toa14] Alexandru Valentin Toader. *Assessment Service for Spreadsheet Documents*. Bachelor's Thesis, Jacobs University. 2014.

[TR14] Valentin Toader and Veronica Rozalia Rus. *Spreadsheet-based decision support systems*. 2014. URL: http://steconomiceuoradea.ro/anale/volume/2008/v4-management-marketing/275.pdf (visited on 01/26/2014).

[TT97] Teo Thompson and Margaret Tan. "Quantitative and Qualitative Errors in Spreadsheet Development". In: *Proceedings of the 30th Hawaii International Conference on System Sciences*. HICSS '97. Washington, DC, USA: IEEE Computer Society, 1997, pp. 149–. URL: http://dl.acm.org/citation.cfm?id=938435.938681.

[ŞA08] Michelle M.H. Şeref and Wayne L. Ahuja Ravindra K. Winston. "Spreadsheet-Based Decision Support Systems". In: *Handbook on Decision Support Systems 1*. Springer, 2008, pp. 277–298. DOI: 10.1007/978-3-540-48713-5_14.