Modeling Task Experience in User Assistance Systems

Andrea Kohlhase
DFKI (German Center for Artificial Intelligence)
Enrique-Schmidt-Str. 5
28359 Bremen, Germany
andrea.kohlhase@dfki.de

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

ABSTRACT

One of the major issues for user assistance systems consists of "providing help at an appropriate level". In this paper we analyze the problem of modeling task experience — a prerequisite for provisioning adequate help. In contrast to level-based approaches we propose an ontology-based model, which allows fine-grained modeling of task experience using the concepts of the task domain as granules. The model is semantic in the sense that it allows to take advantage of the relations between concepts to provide novel semantic services and interactions. We present the SACHS (Semantic Annotations for a Controlling Help System, a semantic help system for a spreadsheet-based financial controlling system) software as an exemplary application of the proposed task experience model.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: [User Interfaces — Evaluation/methodology, training, help, and documentation]; H.5.2 [**Information Systems**]: Information Interfaces and Presentation—*Interaction styles*; H.5.2 [**Information Systems**]: Information Interfaces and Presentation—*User-centered design*

General Terms

Documentation, Interaction, Human Factors, Theory

Keywords

User Assistance, Task Experience, Spreadsheets

1. INTRODUCTION

One of the major issues for user assistance systems consists of "providing help at an appropriate level" (e.g., [17, 18]). A closer look reveals that there really are two issues at work here: the problem of *finding* help and the question what *appropriateness* of help for a specific object really means. Here, we want to focus on latter. To keep the discussion general but concise, we assume the existence of some kind of help system $\mathcal{H}$ with localization features that are aligned with an object $\mathcal{O}$ it provides help for. Note that $\mathcal{O}$ does not need to be a software system itself, it may as well be an object like an active document.

In a first approach in [1] Andrade and Novick understood appropriateness in terms of user competence. Concretely, they suggest the "**TAU model**" based on Kearsley's user competence factors to model appropriateness (see [8]). We use this as a starting point in this paper. The TAU model distinguishes three experience dimensions: the Task, the Application, and the so-called User experience. For the purposes of this paper, we take **task experience** to comprise that part of experience that refers to the intention of using or purpose of $\mathcal{O}$, whereas **application experience** refers to experience with $\mathcal{O}$ itself. Finally, **user experience** stands for experience with the technology that accommodates $\mathcal{O}$. Unfortunately, 'user experience' is nowadays used in a very different sense (see e.g., [20]), so a term like 'technology experience' would be a more apt name. Our notions are a generalization of the ones used in [1], hence include theirs. To get an intuition for the different dimensions let us have a look at the concrete example given in [1]. There, $\mathcal{O}$ is the MS Excel software and $\mathcal{H}$ is MS Excel's own help system. Then the T-dimension covers the experience of authoring Excel documents, the A-dimension the experience with respect to the Excel player, and the U-dimension the experience either with the MS Office Suite technology or even more general with handling a computer.

Andrade and Novick used the simple, traditional scale of "Novice", "Intermediate", and "Expert" for rating experience in the A- and U- dimension. Concretely, they suggest writing different help texts for each of the nine points in the resulting AU plane for each help topic (if sensible), each of which they consider a help *level*. As the 'level'-based approach seems unsuitable for the typically complex task dimension, it is left for future work.

Here is where our work comes in: We propose to model task experience *semantically*. In particular, we model it in a semi-formal ontology, where concepts about the help object $\mathcal{O}$ are reified and their relations explicitly represented so that the help system $\mathcal{H}$ can make use of them.

In order to showcase the (added) value of the semantic approach for the appropriateness of user assistance, we will elaborate it using the SACHS system (Semantic Annotation for a Controlling Help System) as an example. It is a help system for the financial controlling system at DFKI (German Cen-

SIGDOC '09 Bloomington, Il. USA

ter for Artificial Intelligence), see [10] for the underlying theory and a detailed description. DFKI's controlling system (**DCS**) is an application that uses MS Excel 2003 spreadsheets as a user interface for an accounting database. SACHS differs from other help systems in that it was developed from the basic assumption that users mainly need help in understanding the complex concepts behind the DCS and that the help system should expose the meaning of these to the user: it takes a *semantic perspective*. Consequently, the system does *not* focus on tasks the user might want to perform, but on the background knowledge needed to understand the DCS. In particular, SACHS as $\mathcal{H}$ focuses on what DCS as $\mathcal{O}$ is intended for. In other words, SACHS concentrates on the task experience dimension. Note that here the A-dimension in the TAU model pertains to the experience with the DCS and the U-dimension to that with MS Excel.

A prominent feature of SACHS will be the use of the notion of "**framing**": The practice of viewing a concept from various perspectives, particularly as an instance of a structure already understood. In contrast to the *presentation* variants predominantly offered in typical help systems framing enables *substance* variants.

In the next section we will detail the proposed model of task experience. Section 3 shows how it can be utilized in a concrete help system. Section 4 concludes the paper.

2. MODELING THE TASK EXPERIENCE

The first problem in modeling 'task experience' is the very meaning of the term 'task', which is often used interchangeably with 'activity' or 'process'. This makes the distinction between application and task experience non-trivial. To break this impasse, we observe that task experience is a snapshot of a status-quo. It is more concerned with having understood work *domains*, especially their objects and underlying relations, than with work *flows*. Therefore, we model task experience via the content and structure of the underlying domain knowledge of the help object $\mathcal{O}$. Concretely, we suggest using a **domain ontology** $\mathcal{D}$. It contains data about the domain in question *enhanced* by metadata, which contain at least information about its structure ("How is the object built up from or defined in terms of already known objects?") and its context ("What does the object's environment look like and what is its relation to other objects?"). While this design choice constitutes a different approach than the special markup infrastructure for tasks, goals and their sequencing and prerequisites in the DITA system (Darwin Information Typing Architecture, see its use for user assistance systems in [5]), this is only so by degree. Even though we did not need this so far, it would be possible to extend our ontology with a task/goal ontology equivalent to the one inscribed in DITA.

To recap the semantic technology of semi-formal ontologies used in this paper we use a slightly extended version of a spreadsheet used by Winograd in [22] as a running example for $\mathcal{O}$: The Excel document in Figure 1 presents a very simple controlling system, which details and explains actual and projected profits to a controller.

In Figure 2 we see a part of the formalized background knowledge for every cell in cell range [B17:F17] in our running example. Note that it not only contains additional knowledge like the name of the company "SemAnteX" for which it represents a controlling system, but also describes dependencies within the spreadsheet, its profit being dependent on its revenues and expenses, for example.

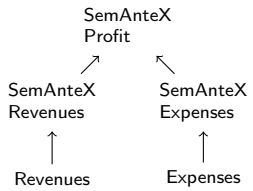


Figure 2: $\mathcal{D}$ for [B17]

Now, given a domain ontology $\mathcal{D}$ we can consider an individual user's task experience as a sub-ontology. In particular, we have natural docking points of the subjective task experience and all there is to know for $\mathcal{O}$. For instance, if a user doesn't know how to make sense of the number in cell [B17], then help for "Profit" is defined from the SemAnteX perspective in the domain ontology. As a consequence a help system $\mathcal{H}$ can make use of the sub-graph under the SemAnteX Profit node, for example by providing dependency links as in the shown graph.

To understand the full extent of the task experience model and the realization of the semantic services, we need to dive more deeply into the underlying semantic technology. We observe that the realization of an ontology is greatly influenced by the choice of semantic format, which ranges from formal ones optimized towards automated deduction (as in software verification systems) to informal/light-weight ones (like folksonomies [21]) that are geared towards consumption by humans. Both approaches have their merit, e.g., the former have to be tediously authored by humans but support the automated discovery of new knowledge while the latter allow a "*brilliantly lazy*" authoring process [16].

For user assistance systems we need to take a middle road: the end-users are human, but support should be generated intelligently, which requires some formality in the underlying data. Therefore, we suggest making the best of both worlds by using a **semi-formal semantic format** like OMDoc (Open Mathematical Documents [12]) to realize task experience. Such a semi-formal ontology format represents ontologies in a structured manner that automated processes draw on (using the "formal" aspects) and allows the extraction of information for human users (using the "informal" aspects given in natural language); see [13] for details.

In an OMDoc ontology information about objects and their properties and relations are expressed as definitions, axioms, and assertions, etc. They are modularized into **theories** that are interconnected via **imports** relations. For instance, in our example ontology we have theories for SemAnteX Profit, SemAnteX Revenues, SemAnteX Expenses, Revenues, and Expenses. Since the SemAnteX Revenues theory imports the Revenues theory and is imported by the SemAnteX Profit theory, all the objects in the Revenues theory can be used in the SemAnteX Profit theory. In particular, on the concept level within SemAnteX Profit the profit is accrued by a company c called "SemAnteX" over a time interval t (say an accounting year). Hence, it is a function $\pi(c,t)$ of a given c in t and *defined* to be the difference of the revenue function $\rho(c,t)$ accommodated in Revenues and the expense function $\epsilon(c,t)$ harboured in Expenses. The visualization of theories with their imports relations in graph form is called a "**theory graph**".

Typically, a user assistance system provides the theory environment by having grouped information objects and its imports-relations by providing links. But a theory can contain more information than just definitions. For instance, SemAnteX Profit may contain the assertion

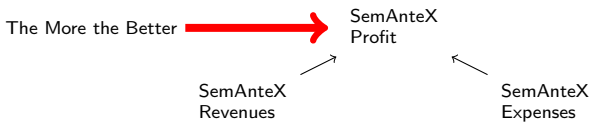
	A	B	C	D	E	F	G	H
1	**Profit and Loss Statement**							
2								
3	(in Millions)			Actual			Projected	
4		1984	1985	1986	1987	1988	1989	1990
5								
6	**Revenues**	3,865	4,992	5,803	5,441	4,124	4,617	5,223
7								
8	**Expenses**							
9	Salaries	0,285	0,337	0,506	0,617	0,705	0,805	0,919
10	Utilities	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: Our Running Example: A Simple Controlling System Using MS Excel after [22]

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

which is justified by an envisioned theory Income lower down in the theory graph. 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 $(*)$. So we have the following situation, where the red, thick link is a view:



Note that all links 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). On the one hand this allows a very efficient reuse of information in a theory graph; on the other hand it allows for multiple explanations in user assistance systems. In our example, we could now also 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 [6].

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 a theory morphism from a source theory (the **framing theory**) into the theory representing the problem (the **framed theory**). The theory morphism itself is called a **frame**. This is justified because the term "frame" has been used in Communication Research as "*schemata of interpretation that enable individuals to locate, perceive, identify and label occurrences within their life space and the world at large.*" [19]. Hence, a frame is understood as a scaffolding of concepts that influence the understanding of situations. In situations where there is a unique morphism from a theory S to T, we will also say that S is a frame for T in a slight abuse of terminology. Note that for every theory S, the identity is a theory morphism, we call it the **natural frame** for S. In our running example the theory The More the Better represents a frame for Profit as well as the theories Revenues and Expenses as transitive closure in the theory graph. The view explanation makes use of framing to anchor the explanation of profit in the user's previous experience.

Finally, if a framing theory 'frames' distinct theories, then we call the set of such framed theories **frame variants**. Consider for instance an Excel controlling system for another company "SemCom", that also uses the theory Expenses in its underlying domain ontology. Where SemAnteX uses the ϵ function from Expenses directly, SemCom defines it to be $\epsilon_{SemCom}(c,t) = 1.1 * \epsilon(c,t)$ in their theory SemCom Expenses to account for general costs. Then the theories SemAnteX Expenses and SemCom Expenses are frame variants with respect to the frame Expenses.

3. SACHS: A SEMANTIC HELP SYSTEM FOR MS EXCEL SPREADSHEETS

Even though MS Excel spreadsheets serve well as an interface for a financial controlling system, they are often too complex in practice. In SACHS we made use of the fact that spreadsheets are *active documents* whose surface structure can adapt to the environment and user input to address this usability problem. In our terminology above, spreadsheets are a help object $\mathcal{O}$ for which SACHS is a help system $\mathcal{H}$.

In [10] we analyzed spreadsheets as semantic documents, where the formula representation is the computational part of the semantic relations about how values were obtained. To compensate the diagnosed computational bias we proposed to augment the two existing semantic layers of a spreadsheet — the surface structure and the formulae by one that makes the intention of the spreadsheet author explicit. Hence, we can use the SACHS system now to demonstrate the advantages for a user assistance system of modeling task experience as a domain ontology that describes the author's or community's intention. In particular, as we do not focus on *finding* appropriate help automatically, we enable the

user to find *what* she needs. We let the user select her own "*product-adoption criteria*" [7]. Unfortunately, even though SACHS' interaction design is elaborate, another consequence of the above focus is, that its user interface is not (yet).

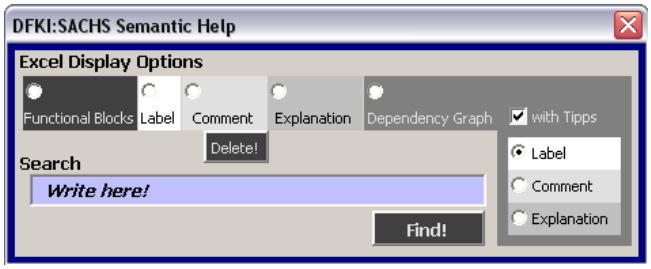


Figure 3: The SACHS Panel

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, has grown and was even called the "*future for software help*" [3]. 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 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.

In particular, for each concept in the underlying OMDoc domain ontology there are three text slots for its description that vary in its granularity. Accordingly, the SACHS panel shown in Figure 3 offers the choice of getting "**labels**" (a title), "**comments**" (a short description), or "**explanations**" (a detailed description). The generated help texts are enhanced by showing the concrete value instances of its parameters. See for instance the generated label for cell [H9] in Figure 7 where the time interval is instantiated as the year 1990.

In the first design of SACHS we had the traditional scale of "Novice", "Intermediate", and "Expert" for aligning content to the user's task experience in mind, which we have withdrawn by now. This form survived in the panel as it still seems to be useful for letting the user decide what kind of *attention* she wants to pay the concept, so by now it is a feature independent of task experience.

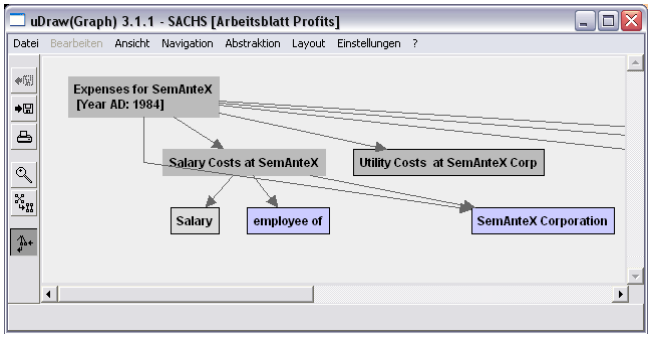


Figure 4: Dependency Graph for Cell [B15]

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 4 are generated. If the user wants to elaborate on a specific concept, then a click on the corresponding node expands it by another level. 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).

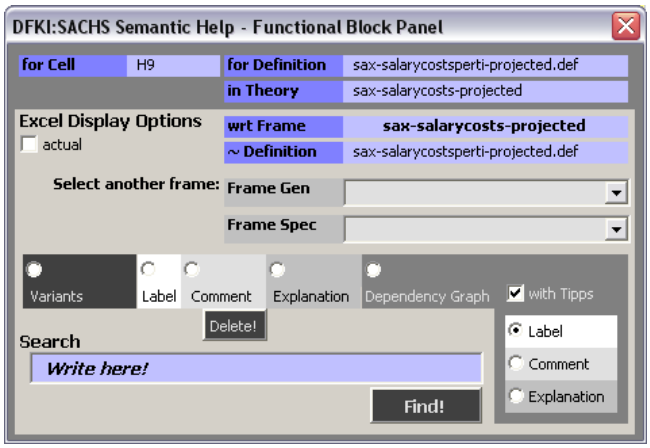


Figure 5: The SACHS Functional Block Panel

Our analysis of the knowledge structure behind spreadsheets revealed that the main significance of cells is that they are part of **functional blocks**: rectangular regions in the grid whose cells can be interpreted as input/output pairs of a specific function. For instance, the cell range [B17:F17] in Figure 1 (highlighted with the selection of [B17] by a borderline) is a functional block, since the cells represent profits as a function π of time. As the functions induced by the functional blocks in a spreadsheet constitute meaning, it is important to make the SACHS interface semantically transparent (see [11]). Therefore, in the SACHS panel (Figure 3) the user can also opt for entering into the "**functional block mode**". Then the functional block of the selected cell is highlighted (as shown in Figure 1) and the panel changes to the one in Figure 5.

The change of the semantically transparent object implies a change of semantic services as well. For instance, now that a user's reference point is a functional block, the process of framing can be supported. The object 'functional block' has a natural frame: the theory the respective concept belongs to. In our example in Figure 5 we can see that the cell [H9] representing the salaries in 1990 (which was selected by the user) is assigned to a definition named sax-salarycostsperti-

projected.def in the background ontology. This definition is located in the theory sax-salarycosts-projected, which also serves as natural frame.

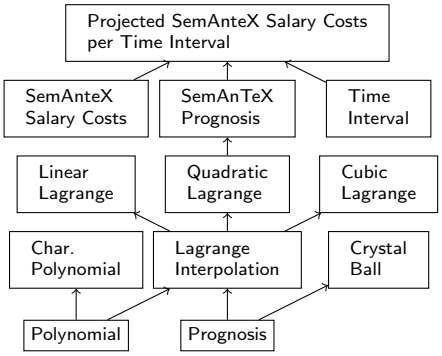


Figure 6: Domain Ontology behind Cell [H9]

Now, if we want to change the frame, we need to select another framing. In Figure 6 we find a part of the underlying domain ontology associated with cell [H9]. It tells us that the value "0.919" was computed *a*) using a prognosis function adapted to SemAnteX, that is *b*) based on the Quadratic Lagrange Extrapolation function that is *c*) a Lagrange Interpolation that is *d*) a function used for prognosis.

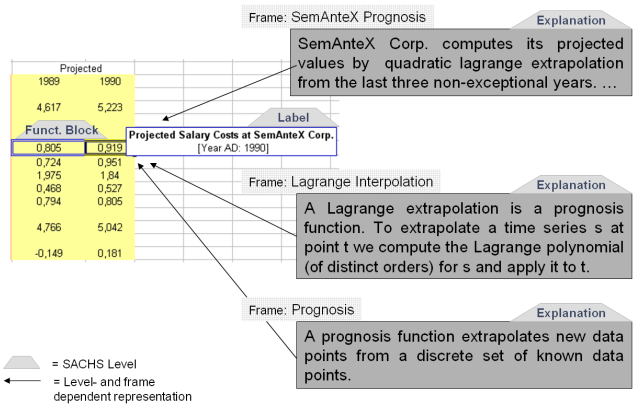


Figure 7: Variant Help Texts for Cell [H9]

The theory in the background knowledge concerned with SemAnteX Prognosis provides a link to the definition sax-salarycostsperti-projected.def and can hence be used as a **frame generalization**. In particular this means that the help content changes. If we look at Figure 7, we see a label that is generated based on the natural frame and three distinct explanations that are generated by SACHS based on subsequent frame generalizations.

Note that the user can usually only get the information with respect to the help authors' choice of framing as the created OMDoc document is fixed and consequently the imports-relation for any theory. Another author might have chosen to associate the Lagrange Interpolation theory directly with cell [H9], or she might have opted for different import relations. Here, the SACHS panel broadens the user's opportunities and takes back the rigor and subjectivity of the authors' choice of framing.

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

Another interesting service SACHS can offer in the functional block mode is the display of **variants**. These variants are the result of on-the-fly application of formula alternatives in the active spreadsheet. This can be done, since the concrete framing assumption reified in the Excel formula for a cell can be changed. In our example, we have three theories specializing Lagrange Interpolation with concrete Lagrange interpolations of different order. From these we can derive spreadsheet formulae, which in turn can be entered into the spreadsheet automatically. In the example in Figure 8, we are looking for variants for the '∼Definition' lagrangeinterpolation.def in the framing theory for the definition sax-salarycostsperti-projected.def assigned by the author to cell [H9]. Concretely, selecting the option "Variants" in the SACHS panel shown in Figure 5 leads to the opening of the "Variants Panel" demonstrated in Figure 8.

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 used as the SemAnteX prognosis function, this is marked by the arrow in front of this variant in the lower right hand side of Figure 8. 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 grey row 10 in Figure 8) enabling the presentation of the variant values for the entire functional block. The according variant formula (in the Excel formula box at the top of Figure 8) is evaluated.

Mehlenbacher points out in [15, p. 140] that a "*cognitive information-processing model of learning involves [...] critical information-human interactions*". In particular, he lists

- "*Information + Comprehension*",
- "*Representation + Integration with existing and available knowledge structures*",
- "*Retrieval + Development of new connections between the new information and the existing state of understanding*", and
- "*Construction + Elaboration toward a richer understanding of the subject matter*".

Observe that the SACHS system deals with all four of them, only notably missing an editor for help text construction for the last point. Thus, SACHS draws on the interdependence of learning, cognition, and user assistance systems.

4. CONCLUSION AND FURTHER WORK

In this paper we have analyzed the problem of what "appropriate help" in the task experience dimension might mean in user assistance systems. We propose to go beyond the simple "level"-based approaches and use a domain ontology as a task experience model. In particular, where the level approach mainly deals with *how* the information is presented (presentation variants), the semantic approach offers additionally a user's interaction with *what* is explained (substance variants) with regard to a specific topic. We argued that such a model allows to better appropriate help, since it is:

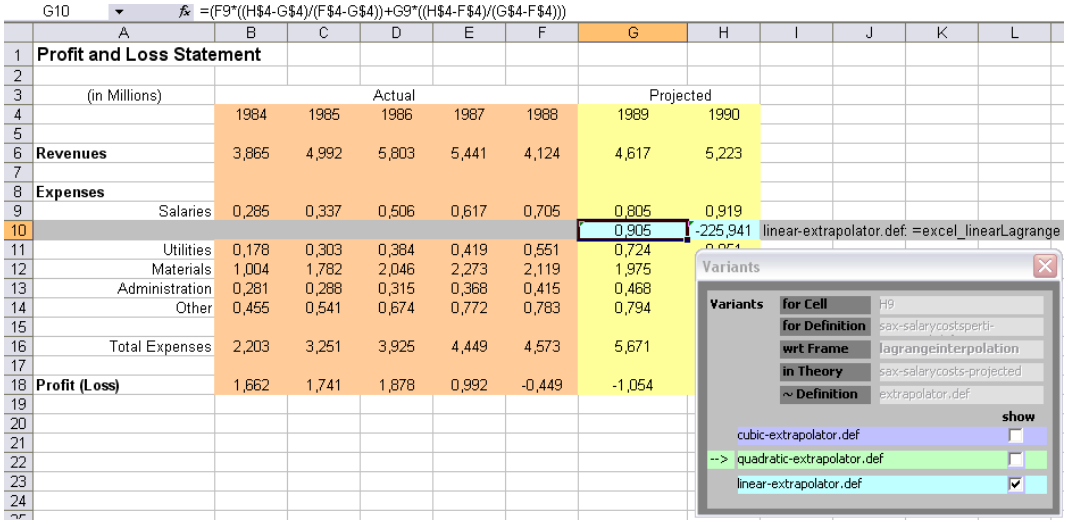


Figure 8: Frame-Based Variant for Cell Range [G9:H9] with SACHS' Variants Panel

fine-granular Instead of having a task experience dimension with few levels, we have as many dimensions as the domain offers concepts. In the SACHS project we explicitly modeled the domain ontology for DCS. From this experience we estimate that such an ontology will contain between 100 and 1000 concepts.

semantic By establishing a domain ontology for the background knowledge, we are really suggesting an n-dimensional space for each topic in the T-dimension in addition to the AU-dimensions of the TAU model. But instead of assuming the n dimensions to be independent, we make use of the fact that ontologies allow to model semantic relations between the objects and concepts they describe. Therefore, this space is usually heavily interconnected. This not only allows a user assistance system $\mathcal{H}$ to offer *semantic services* and *semantic interactions*, but also to redefine appropriateness of help in terms of content and not form only. In [9] we elaborated on this feature of semantic data. In particular, we observed that each knowledge object includes implicit formalizations (content) and explicit realizations (form) and that this set of variants can be structured by notions of equality we called *substance equivalences*, which represent meaning-conserving relations.

Note that the investment in ontology-driven help systems is not necessarily greater than in a conventional multi-level help system, since the amount of text to be written only depends on the distinguishable competence levels. On the contrary, many help texts that are not atomic (i.e., describing more than a single concept) can be aggregated from the simpler components in the ontology representation (see e.g., [4]).

A referee correctly pointed out that our approach shifts the problem of dealing with task experience to relying on humans to build ontologies. We claim that this is a much simpler task. Moreover, background ontologies can be shared across systems, for instance, all financial controlling systems are based on the same (or at least similar) accounting principles. Thus, a standard accounting textbook converted to OMDoc will go a long way in providing task experience based help for such systems. Note that sharing background ontologies across help systems may even aid system migration problems.

We used the SACHS system to showcase various semantic services and innovative interactions based on this semantic approach. Analogous services can be exploited in all help systems based on the suggested task experience model.

In [14] it is argued that different information formats should be integrated with an underlying (system) ontology. In particular, they suggest using an ontology format well-suited for topic maps, so that a "*conceptual navigation*" among all the concerned documents is enabled. Note that our "semantic navigation" based on the OMDoc format does that just as well, because it only draws on the semantic background document and the concrete interpretations in the spreadsheet. Actually, our navigation is much more fine-granular because of the interpretation function.

We believe, that there is no reason why the ontology-based approach should be restricted to the task experience dimension. In particular, the division into task/application/user experience is problematic in the first place as we have already remarked in the introduction. We conjecture that the ontology-based approach to appropriate help can be extended to the AU-dimensions as well by developing domain ontologies for them. Then, the semantic interactions we developed in SACHS could be applied to them. Hence, we would reap the same advantages of fine-grained and semantic modeling. But what is more, the unified approach promises to do away the need to erect arbitrary borders between "experience classes" and moreover would allow to share ontologies for help systems across applications. We will leave an exploration of this idea to further work.

Another direction of research would be concerned with *finding* the appropriate help. In particular, the problem of actually determining or estimating the 'right' sub-ontologies that correspond to the user's task experience automatically. Here, user models could draw on our task experience model.

5. REFERENCES

[1] O. D. Andrade and D. G. Novick. Expressing help at appropriate levels. In C. J. Costa, A. Protopsaltis, M. Aparicio, and H. O'Neill, editors, *Proceedings of*

the 26th annual ACM international conference on Design of communication, pages 125–130. ACM Special Interest Group for Design of Communication, ACM Press, 2008.

[2] J. Carette, L. Dixon, C. Sacerdoti Coen, and S. M. Watt, editors. *MKM/Calculemus 2009 Proceedings*, number 5625 in LNAI. Springer Verlag, 2009.

[3] M. Ellison. Embedded user assistance: The future for software help? *interactions*, 14(1):30–31, 2007.

[4] A. Faulhaber and E. Melis. An efficient student model based on student performance and metadata. In M. Ghallab, C. D. Spyropoulos, N. Fakotakis, and N. M. Avouris, editors, *ECAI*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 276–280. IOS Press, 2008.

[5] M. Hughes. Architecting user assistance topics for reuse: Case examples in DITA. Online publication (`http://www.uxmatters.com/mt/archives/2009/05/architecting-user-assistance-topics-for-reuse-case-examples-in-dita.php`), May 2008. Accessed on 2009-05-28.

[6] 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.

[7] M. Hughes. Progressive user adoption. Online publication (`http://www.uxmatters.com/mt/archives/2009/03/progressive-user-adoption.php`), Mar. 2009. Accessed on 2009-05-15.

[8] G. Kearsley. *Online help systems: Design and implementation*. Intellect Press, Ablex, 1988.

[9] A. Kohlhase and M. Kohlhase. An exploration into the mathematical knowledge space. In M. Kohlhase, editor, *Mathematical Knowledge Management 2005 (MKM'05)*, number 3863 in LNAI. Springer Verlag, 2005.

[10] A. Kohlhase and M. Kohlhase. Compensating the computational bias of spreadsheets with MKM techniques. In Carette et al. [2], pages 357–372.

[11] A. Kohlhase and M. Kohlhase. Semantic transparency in user assistance systems. In *Proceedings of the 27th annual ACM international conference on Design of communication*. ACM Special Interest Group for Design of Communication, ACM Press, 2009.

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

[13] C. Lange and M. Kohlhase. A mathematical approach to ontology authoring and documentation. In Carette et al. [2], pages 389–404.

[14] G. R. Librelotto, J. C. Ramalho, and P. R. Henriques. A framework to specify, extract and manage topic maps driven by ontology. In *SIGDOC '08: Proceedings of the 26th annual ACM international conference on Design of communication*, pages 155–162, New York, NY, USA, 2008. ACM.

[15] B. Mehlenbacher. Communication design and theories of learning. In *SIGDOC '08: Proceedings of the 26th annual ACM international conference on Design of communication*, pages 139–146, New York, NY, USA, 2008. ACM.

[16] K. Miezkowski. Steal this bookmark! Available at `http://dir.salon.com/story/tech/feature/2005/02/08/tagging/index_np.html`, February 2005. Accessed on 2006-9-15.

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

[18] S. A. Selber, J. Johnson-Eilola, and B. Mehlenbacher. Online support systems. *ACM Comput. Surv.*, 28(1):197–200, 1996.

[19] D. A. Snow, E. B. Rochford, S. K. Worden, and R. D. Benford. Frame alignment processes, micromobilization, and movement participation. *American Sociological Review*, 51(4):464–481, 1986.

[20] UX matters: Insights and inspiration for the user experience community. `http://www.uxmatters.com/`, 2005.

[21] T. V. Wal. Folksonomies for IA. Available at `http://s3.amazonaws.com/2006presentations/OZIA/Folksonomy_for_IA.pdf`, September 2006. Accessed on 2006-10-15.

[22] T. Winograd. The spreadsheet. In T. Winograd, J. Bennett, L. de Young, and B. Hartfield, editors, *Bringing Design to Software*, pages 228–231. Addison-Wesley, 1996 (2006).