*Re*examining the MKM Value Proposition: From Math Web Search to Math Web *Re*Search

Andrea Kohlhase and Michael Kohlhase

Computer Science, Jacobs University Bremen
{a,m}.kohlhase@iu-bremen.de

Abstract. The interest of the field of Mathematical Knowledge Management is predicated on the assumption that by investing into markup or formalization of mathematical knowledge, we can reap benefits in managing (creating, classifying, reusing, verifying, and finding) mathematical theories, statements, and objects. This global value proposition has been used to motivate the pursuit of technologies that can add machine support to these knowledge management tasks. But this (rather naive) technology-centered motivation takes a view merely from the global (macro) perspective, and almost totally disregards the user's point of view and motivations for using it, the local (micro) perspective.

In this paper we go a first step into a more principled analysis of the MKM value proposition by focusing on motivations for mathematical search engines from the micro perspective. We will use a table-based method called the "Added-Value Analysis" (AVA) developed by one of the authors. Even though we apply the AVA only to mathematical search engines, the method quickly leads to value considerations that are relevant for the whole field of MKM.

1 Introduction

Mathematical knowledge management (MKM) is a field at the intersection of document management, knowledge representation, and meta-mathematics. Like the first (and unlike the third), it is driven by practical motivations, i.e. by the desire to create technologies that help researchers, scholars, students, and engineers in dealing with mathematical knowledge.

The field has been inspired by experiences in *Formal Methods*: formalizing the intended behavior of programs in logic-based specification languages and employing semi-automated deduction technologies to verify programs against these by using formal proofs can indeed lead to safety and security assurances that are considered so valuable in some high-risk domains that they justify the extremely high costs involved. However, most applications of mathematics are less safety-critical, and with peer-review the mathematical community has a sufficient (much less cost-intensive) instrument for verifying their results. Therefore, MKM concentrates on management tasks for mathematical knowledge that involve higher volumes of information, but possibly less formal depth of representation. The intuition is that mathematical practices like notational adaptation,

M. Kauers et al. (Eds.): MKM/Calculemus 2007, LNAI 4573, pp. 313–326, 2007.

document aggregation and translation, semantic search and navigation, classification of mathematical objects, refactoring of mathematical theories, or error spotting can be based on lightweight formal annotations and content markup techniques for mathematics.

In this respect, the MKM approach is similar to the much-hyped *Semantic Web*, and suffers from the same problem: before the inference-based techniques of the field can pay off, a large volume of data (mathematical documents for MKM and web content for the Semantic Web) must be semantically annotated. Both fields also agree on how this should be achieved: rather than waiting for artificial intelligence methods that can automate this, we rely on content authors or volunteers to supply the annotations. Here, we have a very important difference to the case of Formal Methods, where the connection of the sacrifices incurred in program verification are directly linked to the expected benefits, usually a radically reduced risk of liability or reduced insurance payments. In MKM and the Semantic Web, the benefits lie with the "readers", while the sacrifices remain with the "authors" — creating what we call the *"authoring dilemma of MKM"* in [KK04] or is referred to as MKM's "chicken-and-egg problem"[1].

In [Koh06a] the authoring dilemma was traced back to differing perspectives on the problem: the **micro perspective**, a (local) view from within, and the **macro perspective**, a (global) view from without. The direct link between benefits and sacrifices for a user in Formal Methods consists in the alignment of the micro and macro perspectives, whereas the differentiation between authors and readers in MKM and Semantic Web is an expression of its drifting apart. From a macro perspective authors and readers are just roles of a single user, who will change dynamically between them. In contrast, from the micro perspective of that user, the benefits lie in the far future. MKM technology like most other designed systems usually takes the macro standpoint and almost totally disregards the user's point of view and motivations for using it in the micro perspective.

The MKM community sometimes calls for a "business plan" for MKM, but to our knowledge it has never seriously been attempted as a business plan must take the designer's macro perspective *and* the users' micro perspectives into consideration, so that great visionary technology gets actually used. In particular, we believe that MKM's value proposition is lop-sided and must be *re*examined, e.g. marketing experts like Normann and Ramirez argue: *"Like a portrait which over time dictates to those who see it how the person portrayed actually looked, models also tend to transform what they model, constraining that reality within the limits of the model's logic"* [NR98, xvi].

In this paper we go a first step into a more principled analysis of the MKM value proposition. We will use a table-based method called the "**Added-Value Analysis (AVA)**" [KM07] developed by one of the authors to investigate the value constellations (core problems, their solutions, expected benefits, incurred sacrifices, and added values) that may activate people into users of

[1] It is often lamented that MKM technologies will only become useful, once we have a large corpus of semantically enhanced background material, but we can only realistically develop one once we have the MKM technologies.

mathematical search engines. Our motivation for this is twofold: first, we are interested in understanding the potential of our own search engine MathWeb-Searh [KŞ06, KŞ07] more thoroughly and secondly, we want to introduce the MKM community to the AVA method.

If we want to realize our dream of creating a viable technology and a semantic resource with universal coverage in mathematics [Far05], then we will have to convince many authors that there is value for them in doing so.

2 Added-Value Analysis and Easy-to-Fall-For Catches

The "Added-Value Analysis (AVA)" [KM07] was developed as an interaction design method, that allows and supports a view from micro perspectives, i.e. a theoretically unlimited number of micro perspectives. It differs from well-known interaction design methods because of its focus on the "here-and-now" of a user in the *process of using* the software.

The micro perspective influences a user's evaluation of a concrete context and thereby determines her action. With the Added-Value Analysis this contextual evaluation process can be (re)constructed. It makes use of the **double relativity of added-values**. On the one hand "value" is always relative to the individual (and not to the object under consideration), on the other hand only if this value is fixed, we can add *more* value. This implies that value *can* be manipulated. Moreover, we cannot think any longer in simple value chains, but have to consider rather complex value constellations [NR98], which implies that an object's value may change in the process of using it. In particular, there is no such thing like an added-value per se; its relativities create the basis for understanding taking-action based on underlying value constellations from a micro perspective.

How to Do an Added-Value Analysis? The starting point for the AVA is the understanding the term "Added-Value" based on [Grö97]. In particular, before we can fix the meaning of **"Added-Value"** of a software package, we have to state the *core problem* it tackles. Then we can speak of its (core) value, by evaluating the core problem with respect to the given solution. Afterwards — and only afterwards — we can determine how this value can be enhanced by adding other values. In order to get a handle on a value, the AVA method suggests to systematically split it into benefits received and sacrifices incurred [dChR00] *according to the core problem with the given solution*.

Concretely, we build up a table-like list with the AVA, that contains columns for the core problem, the considered solution, and its evaluation listed in form of lists of benefits and sacrifices, see the following table structure:

#	Pot	Trigger	Core Problem	Solution	Benefits	Sacrifices
0					•	•

The first step consists in determining the *"initial" problem* and a solution for it, then we make state the benefits and/or sacrifices incurred by them. Special care has to be taken that these benefits/sacrifices are such with respect to the core problem *and* the considered solution. Note that the quality of the AVA depends strongly on the precision of all formulations. For instance, if a general problem is addressed with a very specific solution, then a general benefit typically indicates the inappropriateness of the solution level, whereas a very specific benefit indicates the inappropriateness of the problem level.

Now, we **iterate this step** by using either any component of the previous line or any other association at this level as a trigger for a new core problem. For example, if a sacrifice is incurred on the user in Step #1, this can be taken as another core problem for which either a solution exists or is needed. If former, then we can consider it as added-value, if latter, then it becomes a *potential* added-value to the previous problem, which is marked as such in the second column of the table. If a row is triggered by another row in the AVA table, we label this process either by color or by a reference. We use both below, in particular a reader can use a **reference like "$\rightsquigarrow$ 5"** to follow the use process top-down, and the **color** to follow it bottom-up. Moreover, we explicitly note the trigger so that the association chain is enhanced and hence not easily broken. These streams can be reinterpreted as unfolding use processes, in which utility problems are not present up to a certain depth of already using the software. Naturally such cognitions can be exploited in interaction as well as interface design, but this is not the focus of this paper. For convenience, we use the abbreviation **"NN"** ("nomen nescio" = "I don't know") to indicate, that we either couldn't think of what to write down or that we intentionally do not follow this line of thought (for now).

Added-Value Analysis' Related Work. To fortify our intuition of the AVA method, and especially for its reinterpretation as a value proposition analysis tool, we will compare it to related interaction design methods first and highlight its distinguishing features: In 1996 Batya Friedman coined the term "*Value-Sensitive Design*" [Fri96] with which human values like "autonomy" reenter interaction design. This concept meant a shift from a technology-centric approach towards a user-centric one. The AVA incorporates the value-sensitive design method by looking at the subjective values for the decision-taking for action or non-action from the micro and not only from the macro perspective. Within the "*Humanistic Research Strategy*" [Oul04] human values are considered as well, but the underlying reasoning for action is restricted to experience and culture, whereas the AVA takes the processuality of action into account. Again we might say, that the micro perspective distinguishes both design approaches from the AVA. Finally, we like to point out that the "*Contextual Design*" method [BH99] differs not by objective but by implementation as their method is based on contextual inquiry, ours is based on thought trajectories.

With these differentiations in mind, we will look now at possible applications for the Added-Value Analysis method.

What Is an Added-Value Analysis For? Note that the AVA can be used for any technology, that involves design and use processes. It is not relevant, whether the design process is done *before* the actual use process like in a typical software product or *during* it like writing a diary (in which what is written inspires what will be written and vice versa). But in the latter the processes are much more interwoven and therefore, we don't go into this spider's nest. In former, we can distinguish the time-line of a use of AVA from the designer's standpoint: it can either be applied in, inbetween, or after the design process as the view of micro perspectives is helpful in either one.

– When the AVA is used *in* a software design process, it functions as a *system analysis* tool with a focus on the evolving interaction design.
– *Within* the design process, the AVA helps to find potential added-value services and enables the designer to reevaluate the anticipated utility of the software component in question. As it tends to discover unforeseen user needs, in this phase, we can consider it as a *usability testing tool* as well as a *creativity tool*.
– If AVA is applied to an *already existent* product (which we do not consider extensible for the moment), we can reconstruct its interaction design as well as evaluating its strengths and weaknesses. If we start with a general initial problem in a specific field and are informed participants, we can *reconstruct* the underlying discourse dynamics and hence its *value proposition*.

Overall, the Added-Value Analysis modifies the understanding of the processes that take place when the designed object is actually used. Therefore, we can also see opportunities for using it from the user's perspective, for example for a new manual design or a user's transparency needs.

But the application of AVA holds some easy-to-fall-for catches, that influence many designs and which we will try to spell out in the following subsections.

Catch 0: The Quest for Objectivity. We are used to the fact that an analysis yields objective results. But as the AVA makes heavy use of thought trajectories, which are of subjective nature, and not "objective" logical deductions, its results are "valid" with respect to the distinguished worked-out AVA list — which can hardly be called objective. Contrary to expectations, the wanted "view of micro perspectives" for a better understanding of the processuality of use actions is an organized *quest for subjectivity* (and thereby more often than not exceeding expectations). In particular, the AVA method cannot be used for an objective evaluation of a software product but for a systematic exploration of the "subjective" micro perspectives: it uncovers value constellations; these are values from a micro perspective.

Catch 1: Knowing the Answer Before the Question. Downsizing from the macro perspective to the micro perspective is not easily achieved. First, the setting of the initial core problem is rather difficult, for an example look at the discussion further below. From a macro perspective we can and actually frequently start with rather fuzzy problems like "saving the world" with software "xyz". Then the list of benefits in the evaluation process is rather awkward like "xyz makes people happy". Therefore the hard question is what core problem that "xyz" might actually solve. Note that most people tend to know the explicit answer but not the explicit problem.[2] The underlying reason for this difficulty is, that a user's value considerations are not distributed any longer over a straight road, not even a curvy road: it is a *value landscape* with several dimensions.

Catch 2: Solutions ARE the Benefits. In the process of the AVA, it often seems as if the solutions are already the benefits: they solve the problem and that alone feels like a benefit. This is usually a sign that the AVA analyst is not sufficiently independent from the software author's motivation (from the macro-perspective). On the one hand, if the user considers the solution among others from a micro-perspective, then her evaluation is independent of this perceived benefit as it doesn't deliver an advantage over another solution. On the other hand, if it is the only available solution, then the evaluation isn't necessary for taking action.

Catch 3: Problems Are Rhetoric Questions. A similar problem appears if the core problem is formulated in a way that the solution is not only evident, but the only reasonable possible solution. In that case, we cannot think of any benefits or sacrifices as their is no choice, therefore, the AVA would run into a dead end. Probably though, the problem can be formulated in a more general way, so the phrasing of the problem should be reconsidered in the AVA; this alone is an added value of the AVA at the conceptual level.

Catch 4: Values on Too Low Levels The evaluation takes place with respect to a concrete core problem and one concrete solution for it. Often benefits and sacrifices from a higher level are still considered on a lower level. Typically this happens as soon as benefits or sacrifices are repeatedly phrased very similarly in one process chain. Then the AV analyst better remembers that a user wouldn't be at that point (i.e. on that level) if she hadn't already taken into account all preliminary values. This exactly forms the specific value constellation from a micro perspective. Therefore a very precise evaluation is required and in the AVA the correct level has to be determined for an argument.

Catch 5: The Unfinishedness of AVA. With an AVA one cannot decide what is right and what is wrong in a design, but rather what its consequences may be. Each core problem by itself can be considered as a starting point for another AVA. It is dramatically subjective what the reason for using a software or not. Therefore, in our experience the AVA always feels unfinished, some termed it "the AVA infinity". This is a feature and not a bug of the AVA: if we as designers know of many of these micro perspectives, we might be able to offer manifold solutions e.g. by abstracting in the right direction. Moreover, the realization that the AVA is not a tree, but a graph, enables a complex interplay between several components.

[2] Think of the fame of Douglas Adams' number 42. This also resonates with the fact that MKM is a technology-driven community always on the search of a "killer application" much like a person with a hammer desperately looking for a nail.

3 Math Web *Re*search: An Added-Value Analysis of MathWebSearch

In the application of the AVA to the MathWebSearch (MWS) [KŞ06, KŞ07] system, we started with assuming that the initial problem for search engines for mathematics is *"finding content representations of formulae"* (see row 8 in the table below), and immediately fell victim to a combination of Catch 1 and Catch 3 mentioned above, i.e. "knowing the answers before the question" and "problems as rhetoric questions". In fact, finding formulae is just what a formula search engine does, and relying on content markup is the specific approach our system takes. Remember that we are doing this analysis, because we want to get a view of micro perspectives and not to simply reconfirm our design decisions so far. Therefore, we have to take the standpoint of a potential user. But which real user *starts* with the question "finding content representations of formulae"? It seems obvious, that such a user is probably a researcher of math web search in the first place. Hence, this level of argumentation is more at the level of the introduction of a research paper than it represents a starting point for a value analysis.

Suspiciously, this initial problem does not even cover competing designs of mathematical search engines, e.g. [MY03, LM06, MM06]. We interpret this as a sign, that we started on a rather deep level of the AVA and therefore should rephrase the initial problem in a *much* more general way. Hence, when we put ourselves in the user's shoes, we realize that we might rather want to find occurrences of mathematical objects (irrespective of their representations in mathematical formulae; see row 13, 12, and 5), and what we really want are answers to *"mathematical questions"*. Therefore, this is what we will take as the initial question (row 1) of the AVA in the table below.

#	Pot	Trigger	Core Problem	Solution	Benefits	Sacrifices
1			math questions	finding math answers in documents	• documents as reified knowledge ⇝ 3 • standing on the shoulders of giants	• answer space restricted to available documents • judging document credibility ⇝ 4
2			— dito —	ask experts	• get more than you asked for	• finding people who know ⇝ 20 • get more than you asked for
3		reified math knowledge	kinds of math knowledge	math practice: classification into formulae, statements, theories	• provenience of math assertions • epistemic status of document fragments	• three-level distinction enough?⇝ 5
4	P	judging document credibility		NN	• NN	• NN

Note in particular, the identification of the underlying initial question allows us to see design alternatives of our system from a higher-level perspective (e.g. the MKM one). For instance, if we want answers to mathematical questions, we may ask people who know — a time-tried method in mathematics, which immediately begs the question of how to identify experts.

The other hidden assumption unearthed here, is that MKM search restricts itself to search in documents (in the widest sense) as sources for reified knowledge, and that (of course) we can only find answers in mathematical documents, if they are machine-readable and accessible to the search engine. Once we have found a suitable document, we have to analyze the provenience of the information, i.e. the credibility of a searched document has to be judged. If the documents discern e.g. the three levels of mathematical practice (formulae, statements, and theories), that is if they are suitably marked up (e.g. in a document format like OMDoc [Koh06b] with explicit structural annotations), then the reader can obtain machine-support in principle. In particular, then we can be interested in whether a found or looked-for math statement like an assertion is assumed, conjectured, or proven, and if so, what assumptions the proof is based on, as its provenience is known as well as the epistemic status of this document fragment.

Now, there are various ways of finding relevant mathematical documents, including using metadata like the Mathematical Subject Classification (MSC), by author or keyword search using standard search engines, or (again) by asking experts (see Catch 5). Here, we will only concentrate on formula search (see row 5), as we use the three-level distinction as a trigger to only look for formulae.

#	Pot	Trigger	Core Problem	Solution	Benefits	Sacrifices
5		three-level distinction enough?	finding math formulae	formula search engine	• finding *references* to formulae	• thinking in formulae • crafting queries $\leadsto$ 6, $\leadsto$ 7 • no differentiation between function and form $\leadsto$ 8,$\leadsto$ 12 • media change $\leadsto$ 7
6		crafting queries	typing correct queries	structural query editor	• GUI-like	• restricted query subset
7	P	crafting queries, media change	typing correct queries	invasive integration	• no learning curve • no media change	• development costs

The main hidden assumption uncovered here is that it is a prerequisite of math search engines that the user has to think in formulae to use them. Obtaining the formula to search for may be a big part of the problem — indeed we teach this to our children as “math word problems” or “algebra story problems”. Currently, the consensus in the MKM field seems to be that the pre-formulaic stage should not or cannot be supported by our methods.

Now that we have drilled down and motivated formula search as a (derived) core problem, we can look at the sacrifices incurred: obviously we have to learn how to craft queries, and we have to decide whether we want to find formulae by their form or their function. We consider the latter a sacrifice, since it requires a decision and mental activity by the user and therefore constitutes a hurdle for using MKM technologies. Furthermore, almost all math search engines require a change of medium: they require the user to enter a formula representation into the input form of a special web page (Mihai Şucan's browser plugin for MathWebSearch [Mat07c] being a partial exception). To support the user in crafting correct queries, we see two alleys, one is supplying a structural input editor like MathWebSearch [Mat07b] or Mathdex [Mat07a] do, or to integrate the search functionality into the editor that users are familiar with and use for math document development anyways (see [KK04] for an introduction to invasive authoring).

The querying problem is much simpler to deal with, if we want to search formulae by their form since presentation formats for formulae are in much wider use. Quite generally, there seems to be an *equivalence principle between queries and intended results*: We type keywords into Google to find pages containing these words, and we issue formulae as queries to math search engines. The next block of the AVA table will be concerned with the problem of finding formulae by their form:

#	Pot	Trigger	Core Problem	Solution	Benefits	Sacrifices
8		content / form differentiation	finding formulae by form	formula presentation search engine	• finding formulae by visual cues	• thinking in formula layout $\rightsquigarrow$ 9,$\rightsquigarrow$ 10
9		thinking in layout	prioritizing visual cues $\rightsquigarrow$ 11	weighting by glyph composition	• higher visual recall & precision	• assumption: uniform glyph meaning
10	P	— dito —	low mathematical recall	semantic query expansion	• higher math. recall	• assumption: uniform glyph meaning
11	P	prioritizing visual cues	low-priority sub-layouts	glyph similarity search	• targeted fuzzy search • higher visual recall	• lower mathematical precision

It turns out that the main sacrifice about finding formulae by their form is, that we have to think about them by their layout, i.e. we have to know what they look like, which tends to miss relevant occurrences if we completely specify this. Therefore, some presentation formula engines offer some ways of broadening the search by weighting formula parts (which partly recovers a semantical flavor). We can see this as a measure of prioritizing some visual cues over others, and we can (potentially) take this to its logical conclusion by deemphasizing some low-priority parts of the layout to allow for similarity search on these. Further semantics (i.e. presentation-independence) could be added to search by adapting a technique from information retrieval: Query expansion adds semantically similar concepts (e.g. alternative presentations) to a query to obtain better coverage (see e.g. [QF93] for details).

For the problem of finding formulae by function (finally coming to the initial starting point of our AVA), we can state the general prerequisite that users must be able to think about formulae in terms of their function. If they do, and are able to formulate their problem as an instance problem (this is the only query type that MathWebSearch can answer, i.e. given a query formula

q find an occurrence of a subterm t, such that $\sigma(q) = t$ for some substitution σ), then the system can efficiently find formula occurrences, retrieve and display them. One intended application here is to remember forgotten (i.e. only partially remembered) formulae [KŞ06] (see also row 14), so MathWebSearch can act as a memory trigger, leaving the user with the task of judging whether the formulae returned are adequate. Here, system support is given by displaying the formula and associated substitutions.

#	Pot	Trigger	Core Problem	Solution	Benefits	Sacrifices
12		content / form differentiation	finding formulae by function	formula content search engine	• finding formulae independent of presentation $\rightsquigarrow$ 13	• thinking in formula function
13		finding formulae independent of presentation	finding content formulae	MWS	• finding URL of OM/MathML instance • Retrieve URL	• formulate as instance problem $\rightsquigarrow$ 14
14		formulate as instance problem	finding forgotten formulae	MWS	• memory trigger	• judge precision

Another (largely unexplored) application is that we can use instance queries to find applications of general mathematical results (which can be expressed as universally quantified formulae): here we can just use the quantifier-free formula body as a query term (see row 15 and note Catch 2 "benefits are the solutions"). Similarly, we can find counter-examples by negating the body (see 16 and note Catch 4 "values on too low levels"). We do not pursue the encountered catches here, but they imply that there is more work to do. If we apply these two techniques aggressively (and speculatively) over large bodies of mathematics, then we can conceivably even extend this to a (weak) form of formula induction, where we conjecture theorems from known results.

#	Pot	Trigger	Core Problem	Solution	Benefits	Sacrifices
15	P		finding applications	MWS	• finding applications	• formulate input (theorem body) as formula
16	P		finding counterexamples	MWS	• finding counterexamples	• formulate input (theorem body) as formula, then negate
17	P		formula induction		• NN	• NN

Note that all of these applications rely on MathWebSearch being able to answer instance queries efficiently, using term indexing techniques developed for automated deduction. But there are other indexing methods that allow to index for other kinds of queries: for instance generalization queries. Here, the query consists of a query formula q, and a generalization search engine, that efficiently finds occurrences t of subformulae such that $\sigma(t) = q$. With this kind of query, we can find theorems applicable to a given formula (e.g. when we are stuck in a proof). Another immediate application is to use this technology to determine novelty of research results or to settle priority claims.

#	Pot	Trigger	Core Problem	Solution	Benefits	Sacrifices
18	P		finding applicable theorems	MWS(G)	• finding universal generalization • document as context • creative trigger	• judge context compatibility
19	P		determining novelty/priority		• NN	• NN

We can use (presentation or content) formula search engines to find experts — a leftover task from the beginning of our AVA, when we were examining ways to obtain math answers — by their publications. So, if we can find documents dealing with mathematical objects we are interested in, we can usually find experts for these by looking up their authors, acknowledgments, or citations. Finally, we can use formula search engines to search for data: If we can fit the data (e.g. time series) by closed-form expressions (i.e. formulae), then we can search them via formula search engines. The advantage here is that this approach avoids having to develop direct indexing techniques for data, and we have automatic similarity search via the approximation during formula fitting.

#	Pot	Trigger	Core Problem	Solution	Benefits	Sacrifices
20	P	finding people	finding people	by math objects in publications $\rightsquigarrow$ 5	• authors are experts	• bibliographic metadata
21	P		finding similar data	finding fitted formulae	• prescribed	• finding fitting formulae $\rightsquigarrow$ 22 • loss of precision by approximation
22	P	finding fitting formulae	fitting formulae to data		• NN	• NN

4 Conclusion

We have presented the Added-Value Analysis as an associative method for exploring a user's value constellations from a micro perspective and applied it to an emerging subfield of MKM: formula search engines. Even though we initially applied the AVA to the specific mathematical search engine MathWebSearch, the method quickly led to value considerations that are relevant for the whole field of MKM and to new potential services that could be explored in the future, e.g. semantic query expansion or a glyph similarity search service (see rows 10, 11).

In particular, consider the value propositions of "thinking in formulae" and the pros and cons for "precision/recall of mathematical queries". The first indicates a discrepancy between intuitions about anticipated and intended users for the math search solutions. On the one hand, formula search engines assume the user's capability of thinking in formulae to use them, on the other hand, they dream of either layman users (that want to pose math questions) or professional mathematicians. For the first a GUI (row 6) is developed, but for the latter who typically still prefer pen and paper above editors, OCR methods would be best as input technology — an opportunity for [SSWX05]?

We were also reminded that math web search — as it is handled now — *requires* "formulae competence" as a quality of the user. In particular, a user gets references as results and has to decide whether the URL is relevant herself. The underlying assumptions again imply math-knowledgeable users, for instance mathematicians. One problem might be the above mentioned equivalence principle between query and search result: the query has to be made associative, lightweight, and *not* verifiable, whereas the search result needs to be trustworthy and verifiable.

Another problem we make out consists in the prediction, that the above anticipations for math web search users indicate, that these users are also the ones we are aiming for as authors for data in our systems. If we assume that, then more work has to be done to support such users within their natural "habitat", which we believe strengthens our case for semantic, invasive editors like [Koh05a, Koh05b] or [Koh06c].

Other parts of MKM can also profit from such an analysis, and as the AVA is subjective in the choices made, even a re-analysis of math search might lead to additional insights. Note that the AVA does not directly lead to a recipe for motivating authors to take action and contribute semantically enhanced materials, but it does help to uncover added-value situations, i.e. such situations in which users can obtain value without incurring sacrifices additional to those already amortized by solving their core problem. Such AVA tables not only provide "cheat sheets" for a real MKM business plan, they enhance added-values that can help tip the scale towards using them.

References

[BH99] Beyer, H., Holtzblatt, K.: Contextual Design. ACM interactions 6(1), 32–42 (1999)

[dChR00] de Chernatony, L., Harris, F., Riley, F.D.: Added value: Its nature, roles and sustainability. European Journal of marketing 34(1/2), 39–56 (2000)

[Far05] Farmer, W.M.: Mathematical Knowledge Management. In: Schwartz, D.G.(ed.) Mathematical Knowledge Management, pp. 599–604. Idea Group Reference (2005)

[Fri96] Friedman, B.: Value-Sensitive Design. interactions, ACM, pp. 16–23 (November and December 1996)

[Grö97] Grönross, C.: Value-driven relational marketing: from products to resources and competencies. Journal of Marketing Management 13, 407–419 (1997)

[KK04] Kohlhase, A., Kohlhase, M.: CPoint. In: Asperti, A., Bancerek, G., Trybulec, A. (eds.) MKM 2004. LNCS, vol. 3119, pp. 175–189. Springer, Heidelberg (2004)

[KM07] Kohlhase, A., Müller, N.: Added-Value: Getting People into Semantic Work Environments (submitted 2007)

[Koh05a] Kohlhase, A. (2005) `http://kwarc.eecs.iu-bremen.de/projects/CPoint/`

[Koh05b] Kohlhase, A.: Overcoming Proprietary Hurdles: CPoint as Invasive Editor. In: de Vries, F., Attwell, G., Elferink, R., Tödt, A. (eds.) Open Source for Education in Europe: Research and Practise, pp. 51–56. Open Universiteit of the Netherlands, Heerlen (2005)

[Koh06a] Kohlhase, A.: The User as Prisoner: How the Dilemma Might Dissolve. In: Memmel, M., Ras, E., Weibelzahl, S. (eds.) 2nd Workshop on Learner Oriented Knowledge Management & KM Oriented e-Learning. pp. 26–31 (2006) Online Proceedings at `http://cnm.open.ac.uk/projects/ectel06/pdfs/ECTEL06WS68d.pdf`

[Koh06b] Kohlhase, M.: OMDoc – An Open Markup Format for Mathematical Documents [version 1.2]. LNCS (LNAI), vol. 4180. Springer, Heidelberg (2006)

[Koh06c] Kohlhase, M.: stex: Semantic markup in $\TeX$ / $\LaTeX$ Self-documenting $\LaTeX$ pacakge (2006) available at `https://svn.kwarc.info/repos/stex/sty/stex.pdf`

[KŞ06] Kohlhase, M., Şucan, I.: A search engine for mathematical formulae. In: Calmet, J., Ida, T., Wang, D. (eds.) AISC 2006. LNCS (LNAI), vol. 4120, pp. 241–253. Springer, Heidelberg (2006)

[KŞ07] Kohlhase, M., Şucan, I.: System Description: MathWebSearch 0.3, A Semantic Search Engine. submitted to CADE 21 (2007)

[LM06] Libbrecht, P., Melis, E.: Methods for Access and Retrieval of Mathematical Content in ActiveMath. In: Iglesias, A., Takayama, N. (eds.) ICMS 2006. LNCS, vol. 4151, Springer, Heidelberg (2006), `http://www.activemath.org/publications/Libbrecht-Melis-Access-and-Retrieval-ActiveMath-ICMS-2006.pdf`

[Mat07a] Mathdex (seen Mar 2007) web page at `http://www.mathdex.com`

[Mat07b] Math Web Search (seen April 2007) web page at `http://kwarc.info/projects/mws/`

[Mat07c] Math Web Search Plugin. (seen April 2007) `http://kwarc.info/projects/mws/plugin.html`

[MM06] Munavalli, R., Miner, R.: MathFind: a Math-Aware Search Engine. In: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 735–735. ACM Press, New York (2006)

[MY03] Miller, B.R., Youssef, A.: Technical Aspects of the Digital Library of Mathematical Functions. Annals of Mathematics and Artificial Intelligence 38(1-3), 121–136 (2003)

[NR98] Normann, R., Ramirez, R.: Designing Interactive Strategy. From Value Chain to Value Constellation. Wiley and Sons, Chichester (1998)

[Oul04] Oulasvirta, A.: Finding Meaningful Uses for Context-Aware Technologies: The Humanistic Research Strategy. In: Late Breaking Result Papers (April 2004) ISBN 1-58113-703-6

[QF93] Qiu, Y., Frei, H.-P.: Concept based query expansion. In: Korfhage, R., Rasmussen, E.M., Willett, P. (eds.) Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, ACM, pp. 160–169 (1993)

[SSWX05] Smirnova, E., So, C., Watt, S., Xie, X.: Components for pen-based mathematical interfaces. In: ACA 2005 Special Session on Pen-Based Mathematical Computing (2005)