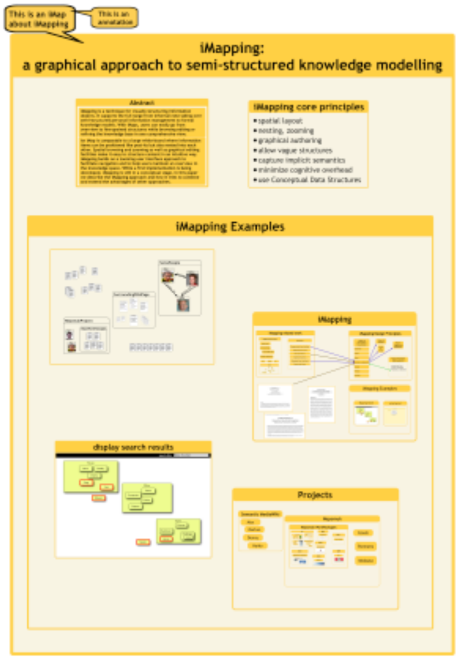
iMapping—
a Graphical Approach
to Semi-Structured Knowledge Modelling

Heiko Haller

Forschungszentrum Informatik,
Universität Karlsruhe (TH), Germany
hhaller@fzi.de
http://www.fzi.de/



iMapping— a Graphical Approach to Semi-Structured Knowledge Modelling

Heiko Haller

Forschungszentrum Informatik (FZI), Universität Karlsruhe (TH), Germany
heiko.haller@fzi.de,
http://www.fzi.de/ipe/

Abstract. iMapping is a technique for visually structuring information objects. It supports the full range from informal note taking over semi-structured personal information management to formal knowledge models. With iMaps, users can easily go from overview to fine-grained structures while browsing, editing or refining the knowledge base in one comprehensive view.

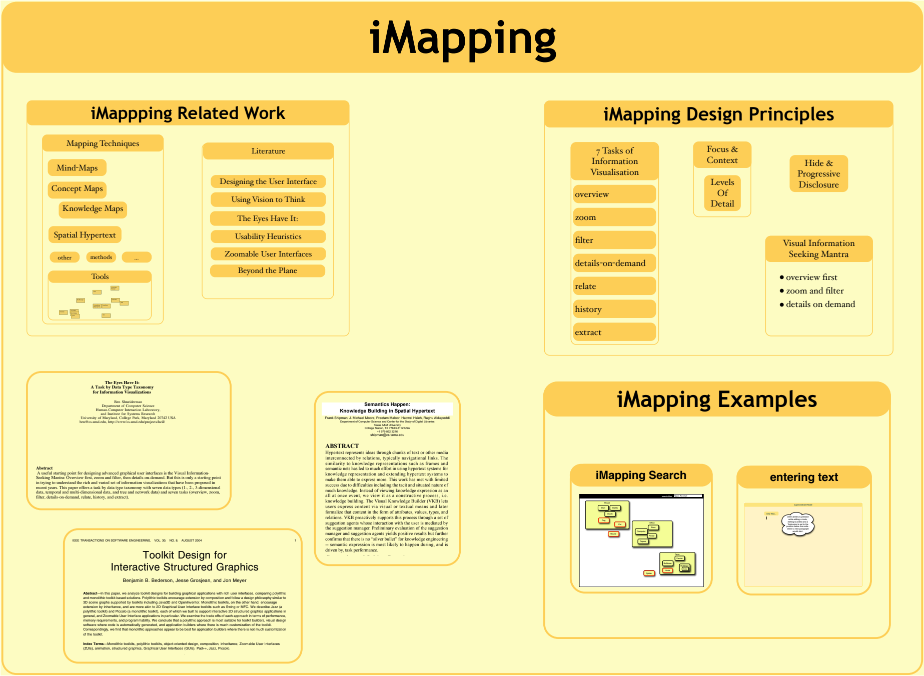
An iMap is comparable to a large white-board where information items can be positioned like post-its but also nested into each other. Spatial browsing and zooming as well as graphical editing facilities make it easy to structure content in an intuitive way. iMapping builds on a zooming user interface approach to facilitate navigation and to help users maintain an overview in the knowledge space. While a first implementation is being developed, iMapping is still in a conceptual stage. In this paper we describe the iMapping approach and how it tries to combine and extend the advantages of other approaches.

1 Introduction

Many knowledge management systems, especially those who rely on highly structured information and metadata being entered and maintained by the users, fail because most users do not make this additional effort if they do not have to. This might be one of the reasons why semantic technologies have not found widespread use so far. For these technologies to be widely used, they have to provide immediate benefit to the single user and it is crucial that they are very easy to use and do not constrain the user in his work. This immediate benefit to the user is more likely to be experienced where users manage their every-day knowledge resources like personal notes, files, bookmarks etc. like in the setting of a semantic desktop environment (see sec. 2.3). The interoperable nature of semantics-based systems makes it likely that such a personal knowledge management system also benefits shared knowledge bases on a larger scale once is adopted by users and integrated in their every-day work-flow. Therefore, in this paper we focus on the perspective of a single user in a personal knowledge management setting.

When semantically formalised knowledge structures are being used, content is typically fine grained and highly structured. Such fine-grained content structures are typically more complex compared to plain text or classical hypertext structures. Even with the relatively simple structures in classical hypermedia, where we simply have interlinked information objects on the granularity level of whole pages or documents, hypertext research has shown that users often get "lost in hyperspace" when browsing without additional navigational help [1]. This stresses the need for user interfaces that facilitate navigation and authoring of such structures without losing orientation.

Human sense of orientation has developed over millions of years. It is highly optimized for orientation in a 3-dimensional world and on large plains and not for finding a way through complex hypertexts or abstract formal structures. Using graphical environments for structuring externalized knowledge enables the user to use his highly efficient sense of spatial orientation on his *knowledge space*.



The Eyes Have It:
A Task by Data Type Taxonomy
for Information Visualizations

Ben Shneiderman
Department of Computer Science
Human-Computer Interaction Laboratory,
and Institute for Systems Research
University of Maryland, College Park, Maryland 20742 USA
ben@cs.umd.edu, http://www/cs.umd.edu/projects/hcil/

Abstract
A useful starting point for designing advanced graphical user interfaces is the Visual Information-Seeking Mantra: Overview first, zoom and filter, then details-on-demand. But this is only a starting point in trying to understand the rich and varied set of information visualizations that have been proposed in recent years. This paper offers a task by data type taxonomy with seven data types (1-, 2-, 3-dimensional data, temporal and multi-dimensional data, and tree and network data) and seven tasks (overview, zoom, filter, details-on-demand, relate, history, and extract).

Semantics Happen:
Knowledge Building in Spatial Hypertext

Frank Shipman, J. Michael Moore, Preetam Maloor, Haowei Hsieh, Raghu Akkapeddi
Department of Computer Science and Center for the Study of Digital Libraries
Texas A&M University
College Station, TX 77843-3112 USA
+1 979 862 3216
shipman@cs.tamu.edu

ABSTRACT
Hypertext represents ideas through chunks of text or other media interconnected by relations, typically navigational links. The similarity to knowledge representations such as frames and semantic nets has led to much effort in using hypertext systems for knowledge representation and extending hypertext systems to make them able to express more. This work has met with limited success due to difficulties including the tacit and situated nature of much knowledge. Instead of viewing knowledge expression as an all at once event, we view it as a constructive process, i.e. knowledge building. The Visual Knowledge Builder (VKB) lets users express content via visual or textual means and later formalize that content in the form of attributes, values, types, and relations. VKB proactively supports this process through a set of suggestion agents whose interaction with the user is mediated by the suggestion manager. Preliminary evaluation of the suggestion manager and suggestion agents yields positive results but further confirms that there is no "silver bullet" for knowledge engineering -- semantic expression is most likely to happen during, and is driven by, task performance.

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 30, NO. 8, AUGUST 2004 1

Toolkit Design for
Interactive Structured Graphics

Benjamin B. Bederson, Jesse Grosjean, and Jon Meyer

Abstract—In this paper, we analyze toolkit designs for building graphical applications with rich user interfaces, comparing polylithic and monolithic toolkit-based solutions. Polylithic toolkits encourage extension by composition and follow a design philosophy similar to 3D scene graphs supported by toolkits including Java3D and OpenInventor. Monolithic toolkits, on the other hand, encourage extension by inheritance, and are more akin to 2D Graphical User Interface toolkits such as Swing or MFC. We describe Jazz (a polylithic toolkit) and Piccolo (a monolithic toolkit), each of which we built to support interactive 2D structured graphics applications in general, and Zoomable User Interface applications in particular. We examine the trade offs of each approach in terms of performance, memory requirements, and programmability. We conclude that a polylithic approach is most suitable for toolkit builders, visual design software where code is automatically generated, and application builders where there is much customization of the toolkit. Correspondingly, we find that monolithic approaches appear to be best for application builders where there is not much customization of the toolkit.

Index Terms—Monolithic toolkits, polylithic toolkits, object-oriented design, composition, inheritance, Zoomable User Interfaces (ZUIs), animation, structured graphics, Graphical User Interfaces (GUIs), Pad++, Jazz, Piccolo.

Fig. 1. Example of an iMap showing three expanded text pages and several sub-maps with collapsed items.

Allowing users to spatially arrange information items may enhance the link between their mental and external models because it enables the use of diagrammatic depictions whose obvious structure corresponds more closely to the structure of the content. This helps the user to intuitively grasp an overview of the subject matter. Unlike text, diagrammatic knowledge representations carry a structural analogy to the content they represent. In other words: A diagram's structure looks similar to the structure it is about. A flow-chart e. g. *depicts the structure* of a process. A text doesn't—It takes a longer way in the user's mind until it can be related to the user's mental model [3].

Research in cognitive and instructional psychology has shown that visual mapping techniques (like Mind-Maps, Concept Maps and others) provide easy ways to rather intuitively structure fine-grained information objects—for an overview see [2].

iMapping is a new visual mapping approach that tries to unite the strengths of several established mapping techniques. It combines these strengths with possibilities of modern IT like deep zooming and semantic technologies. iMapping is meant to support easy informal note taking as well as semi- and fully formalized knowledge engineering in the same powerful yet easy-to-use environment. The basic metaphor of an iMap is that of a large pin-board, where information items can be spatially arranged, enabling users to gain a visual overview over collections of items at once (see fig. 1). These items can represent bits of text as well as any kind of external resources like files, web pages, pictures or other maps. They can also be nested into each other and interlinked in various ways. Besides browsing by hyperlinks, users can navigate an iMap by zooming through it.

This article describes the iMapping approach. The following related work section (2) introduces design principles, their cognitivist background and other approaches, that iMapping is based on. The design section (3) describes iMapping in more detail and how it tries to unite advantages of related approaches. The article concludes (4) with a brief discussion of necessary future work and an outlook on the potential of iMapping.

2 Related Work

2.1 Design Principles

For the design and evaluation of visual information environments, Ben Shneiderman identified the following seven tasks [4]:

Overview Gain an overview of the entire collection
Zoom Zoom in on items of interest
Filter Filter out uninteresting items
Details-On-Demand Select an item or group and get details when needed
Relate View relationships among items
History Keep a history of actions to support undo, replay, and progressive refinement
Extract Allow extraction of sub-collections and of the query parameters

The four first points that also form Shneiderman's well-known "Information Seeking Mantra: Overview first, zoom and filter, then details-on-demand", are all targeted at reducing perceived complexity. This is so important, because human working memory is constrained to holding 7±2 items at a time [5]. This makes it crucial for tools used in knowledge intensive work, to impose as little cognitive overhead on the user as possible [6]. To deal with more than approximately seven items a time, we have to group items into clusters, which are then mentally represented as a single object [7]. This principle of abstraction is one of the most basic heuristics people use to deal with complex information [8], [9]. It is also inherent to context-and-detail techniques like the Levels-of-Detail approach. It means that objects in focus are shown in more detail whereas distant objects are shown in less detail in order to reduce complexity—be it for ease of computation or simply to not confuse the user with an overload of visual information.

2.2 Visual Mapping Techniques

Visual mapping techniques (like mind-mapping, concept mapping and others) are methods to graphically represent knowledge structures. Most of them have been developed as paper-based techniques for outlining, brainstorming, learning facilitation or to elicit knowledge structures. Some of them have proven to be very useful in Personal Knowledge Management, especially for tasks like gathering and structuring ideas and acquiring an overview on a certain domain. For an overview on visual mapping techniques, their cognitive psychological background, related research and a comparative evaluation of some existing techniques and tools, see [2]. In brief, all of these mapping techniques are quite helpful for some purposes but have constrained paradigms that make them useless for others.

Mind-Maps [10] for example, provide an easy-to-understand tree-like structure useful for outlining a topic or sorting items. But they are not suitable to depict the relational structure between items because they are constrained to their hierarchical model. **Concept Maps** [11] on the other hand have a graph-based structure that emphasizes these relations. But they are not as easy to handle, because explicitly specifying all these relations is too laborious e. g. for simple note taking or brain storming.

“Spatial Hypertext” is a very simple approach. The basic idea is to view a self-contained hypertext from an overview perspective, spatially arranging single pages (which tend towards microcontent[1]). However the Spatial Hypertext paradigm expressly abandons the concept of explicitly stating relations between objects and uses spatial positioning as the basic structure. To fuzzily relate two objects, they are simply placed near to each other, but maybe not quite as near as to a third object. This allows for so-called “constructive ambiguity” [12] and is an intuitive way to deal with vague relations and orders. While Spatial Hypertext in its pure form is not suitable to author formal knowledge structures needed for semantic knowledge management, the general approach may well be used as a UI basis.

[1] The idea of microcontent is explained below.

2.3 Zooming User Interfaces

An early research prototype using a zooming approach was Pad and its successor Pad++, both developed in Maryland [2]. It has been used in various applications and also as a web browser capable of showing the viewed web pages and their link-structure from a bird's eye. In a study where participants had to perform browsing tasks in order to answer some questions, subjects using Pad++ were 23% faster than those using Netscape [13]. This shows that large zoomable information surfaces are well-suited hypertext front-ends. The work on Pad++ has later yielded its successors "Jazz" and finally "Piccolo", a toolkit in Java and .Net, that supports the development of 2D structured graphics programs, in general, and Zoomable User Interfaces, in particular [3]. A Semantic Desktop system whose UI is deeply based on zooming is MentalSky [4]. It uses machine-learning methods to semantically classify existing resources into clusters that can be browsed by zooming through and restructured with drag-and-drop interaction. MentalSky is currently in a prototype state of development. It differs significantly from an iMapping system in the respect, that it is constrained to managing external resources (like files, pictures, web-links, etc.) but is not made for authoring content neither in plain text nor in a formal way.

Semantic Desktops and Wikis One of the first semantic desktop systems, that lets the user freely specify semantic relations between typed information items on a topic maps basis, is DeepaMehta [15]. It provides a graph-based UI in a thin client. Once an item (or relation) has been specified (in a topic map), DeepaMehta keeps it in a background repository on the server independent from whether they are still part of an actual topic map. This separation between the structural model and visual model makes sense, also for iMapping, because it allows multiple (visual) instances of an item to be used in different contexts or locations—much like hard links in a Unix file system. For more information on semantic desktop systems in general, see [5].

Microcontent Some knowledge management tools, like e. g. the wiki engine SnipSnap [6], already allow including one or several wiki pages in another page so, instead of having to follow a link, the user can see the target page inline. This is a first step into the direction of so-called microcontent ("... content that conveys one primary idea or concept and is accessible through a single definitive URL... ") [7], which is useful to avoid redundancies, because most pieces of information are relevant in different contexts. In the same way, in an iMap items can be nested

[2] `http://www.cs.umd.edu/hcil/pad++/`

[3] `http://www.cs.umd.edu/hcil/piccolo/`

[4] `http://mentalsky.net`,
`http://cognitivetools.net/tiki-index.php?page=MentalSky`

[5] `http://semanticdesktop.org`

[6] `http://snipsnap.org/`

[7] `http://en.wikipedia.org/wiki/Microcontent`

into other items. Many information items will just be little snippets of text, while others will mainly contain such snippets or other resources, functioning as aggregators. This makes sense especially when items are reused in different contexts.

Conceptual Data Structures CDS is a lightweight top-level ontology about relations that naturally occur in common knowledge artefacts. It is designed to bridge the gap between unstructured content like informal notes and formal semantics like ontologies by allowing the use of vague semantics and by subsuming arbitrary relation types under more general ones. By that it is suitable for representing knowledge in various degrees of formalisation in a uniform fashion, allowing gradual elaboration.

CDS serves two purposes: First, as a guideline for future personal knowledge Management tools, providing a set of crucial structural primitives. Second, the RDF-based representation of CDS can serve as a knowledge exchange format. It is able to represent vague or even inconsistent knowledge structures.

The core top-level relations in CDS are *order* (before, after, etc.), *hierarchy* (corresponding to item nesting in iMapping and subsuming semantic relations like *is_a* and *part_of*), *linking* (subsuming hyperlinks as well as any other freely specified relation carrying formal semantics or not) and *annotation* (subsuming freeform notes as well as tags and types).

In fact our first implementation of an iMapping environment will be based on the CDS data model. For more information about CDS see [14].

3 Design

iMapping tries to combine the advantages of all the above approaches:

- visual knowledge representations with structural analogy to content
- easy hierarchical overall topology
- facility for graph-based relation mapping
- basic wiki functionality (collaborative editing, easy linking, backlinks etc.)
- support for formal semantic statements
- allowing constructive ambiguity
- providing overview by integrating context and detail through zooming

Basic Hierarchy The basis of the iMapping paradigm is a large two-dimensional surface, where items can be freely placed. For typical knowledge management scenarios, these items will usually be short text passages. The size can vary from just a keyword to a short note to whole paragraphs. Because these items can contain other items, it is recommended to use microcontent rather than long unstructured text passages. Longer texts can be represented by a sub-map containing a sequence of smaller text-items thus allowing to deal with the structure on a more fine grained level. Whereas in mind-maps or other tree-like diagrams lower hierarchies branch towards the outside from a central point, in an iMap hierarchy goes down into deeply nested nodes that can be zoomed into (see Fig. 1). Like explained above, there can be multiple visual instances of one and the same information object, because it may be relevant in different contexts.

Content Adding content to an iMap should be as simple as clicking anywhere in the map and typing some text. Instead of a text-snippet, an information item could also be a picture drag-and-dropped into the map, a webpage or a document.

It is important to note, that it should always be possible to add completely vague and unstructured content. Although allowing for semantic knowledge management, an iMapping system should never force the user to specify any semantics. Otherwise the system will not be used in daily work—which it aims to be. Content can later be refined and formalised incrementally.

Levels of Detail Some information objects esp. text, are rather hard to recognise when they are scaled down to thumbnail size. So composite items should have at least two possible states: open and closed, which could also be seen as expanded/collapsed or being inside/outside the node. Switching between these states is done either manually per click or can take place automatically, depending on how large the object is displayed. This method is also sometimes referred to as "semantic zooming". A longer text-item could be represented e. g. by a keyword in collapsed state and with its content in expanded mode. A more structured article could from a distance only show its title, when zoomed larger also some additional information like authors and date, then the table of contents and when zoomed to reasonable size, fade over to the full content.

Link Structure There are three structurally different ways of interrelating items in an iMapping environment:

- classical hyperlinks (like in e. g. wikis) (linking from a particular text position to another item)
- nesting items into another (including the link target inline at a specified position)
- linking on an item level (stating a relation between two objects)

Each of these can be mere navigational links or carry formal semantics, if specified.

On one hand, enabling an overview of the structure and relations between information items is one of the main goals of iMapping. On the other hand, visualising every relation between all items would result in a great mess also known as the "spaghetti syndrome". The idea in iMapping is, to not show any relations by default, and only make them visible on demand (see Fig 2). This could be a subtle interaction like mouse-over or something more explicit—depending on user settings or a mode. Graphically drawing Links between objects in a concept map (node-and-link-) style, is a common and well-evaluated technique that is useful for depicting and authoring relations between information items. Such explicitly visible drawn links stay permanently visible. In the same way, items can be semantically interrelated by simply drawing links between them, which can then be typed. If this is done using auto-completion, reuse of existing relation types is fostered.

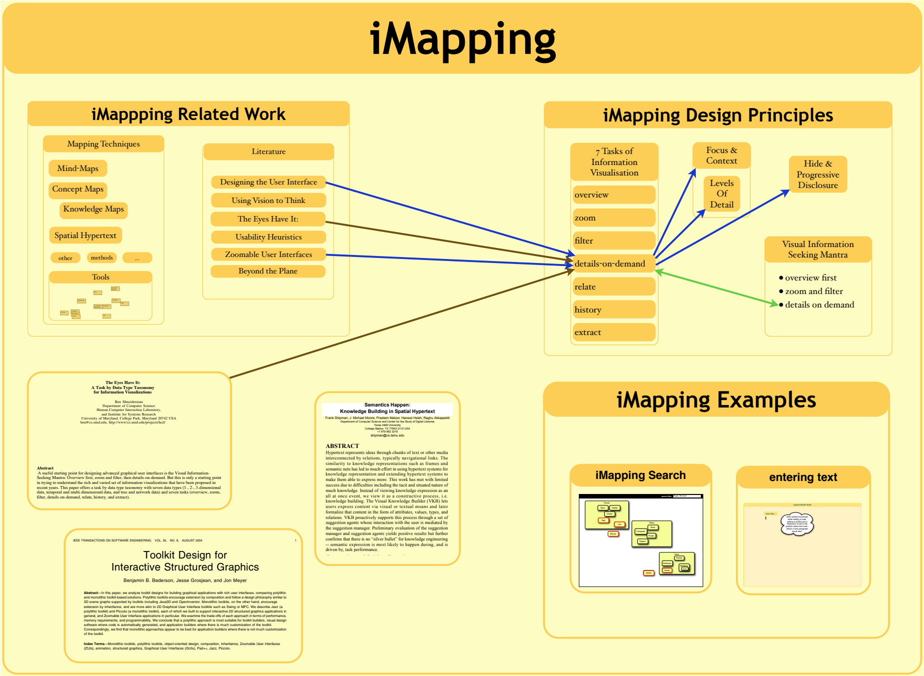


Fig. 2. The same iMap but with link-structure of one item made visible.

3.1 Interaction

A main problem with today's semantics enabled authoring environments is that articulating knowledge structures in a formalized way usually takes a lot of effort—both cognitive and in terms of interaction.

These efforts, can be minimised in several ways:

a) Leaving the user free to specify structures only to the degree of formalization currently needed
b) Reducing the number of interactions by carefully eliminating all unnecessary steps
c) Reducing the number of interactions even further by guessing and proposing probable steps

d) Presenting opportunities to elaborate and formalise structures at the right moment: when the least additional cognitive effort is needed to specify additional semantics
e) Extracting structures implicit in the spatial layout

While a) is mainly an issue of the data format used to represent the content, and is covered by CDS (see 2.3) and b) is a very general principle of interaction design that we will not detail here, c) and d) are specifically addressed by the interaction paradigm that we propose:

With *"mark-up as you go"* we allow the user to formalise her knowledge structures incrementally. E. g. she could start by simply entering plain text keywords and later, while rearranging them, she could assign the Keywords types and tags and draw relations between them.

It is always a certain cognitive effort to later elaborate on already existing structures because the user has to recall the structures and relations between items that were implicit at design time prior to later formalising them. Therefore the best moment to let the user specify formal structures is when she is interacting with them anyway. For instance: when an item is dragged into another item, thus implicitly assigning it a new parent, at this moment the user is probably aware of the type of relation these two items have on a detailed level. So if she can specify the type of the relation between the two items she touched just before e. g. by simply typing a name for the relation without any other interaction steps in between, then this is probably less cognitive effort than doing this as a separate task.

While she is typing she can be further supported in two ways:

1. by auto-completion, which saves interaction costs and at the same time fosters reuse of existing relation types. When an unknown relation type is entered, the user gets feedback that no matching relation is found. If the user still confirms it, a new relation type will be created.
2. When the types of the items to be related are known to the system, relation types can be proposed to the user based on prior knowledge of domains and ranges of applicable relation. E. g. when an person-item is dragged into an event-item, the system could propose probable relations like *participates in* or *organises*.

4 Conclusion

4.1 Discussion

Since iMapping is still in its conceptual phase, and the first implementation is still under development, no user tests have been conducted yet. Whether the iMapping approach will be successful, user studies and time will have to tell.

Since the iMapping approach was initially designed for personal use, there might be unforeseen difficulties when used in collaborative settings. For example, there could be dissent on how items should be spatially arranged. But hopefully, like it is common in wiki culture, over time layouts will converge to a structure that finds consensus. Another approach would be to use personal profiles to let users make their personalised spatial arrangements of the content. The better the content and its structure represented using defined semantics, the easier it is to separate it from its visual appearance and to syndicate it to other applications.

4.2 Outlook

A first java based iMapping environment is under development using the Piccolo framework (see sec. 2.3). It is part of the open source Social-Semantic-Desktop project Nepomuk [8].

Semantic knowledge management technologies might in the future be able to significantly increase interoperability. However, if knowledge management does not start on a personal level, providing immediate benefit to the single knowledge worker, where should the formalised external knowledge come from? Focussing on user interaction and cognitive ergonomics will be an important point, if semantic knowledge management systems are to become widely used whether collaboratively or for personal knowledge management. iMapping-based user interfaces could be a step to make highly structured knowledge management in different levels of formalisation easy and intuitive.

Acknowledgments:

Research reported in this paper has been partially financed by the EU in the Social Semantic Desktop project NEPOMUK (IST-FP6-027705).

References

1. Tergan, S.O.: Hypertext und Multimedia: Konzeption, Lernmöglichkeiten, Lernprobleme und Perspektiven. In: Information und Lernen mit Multimedia und Internet. third, completely revised edn. Beltz, PVU (2002)
2. Haller, H.: Mappingverfahren zur Wissensorganisation (2003) Knowledge Board Europe. Availlable online at http://heikohaller.de/literatur/diplomarbeit/.
3. Schnotz, W.: Wissenserwerb mit Texten, Bildern und Diagrammen. In: Information und Lernen mit Multimedia und Internet. third, completely revised edn. Beltz, PVU, Weinheim (2002)
4. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: VL '96: Proceedings of the 1996 IEEE Symposium on Visual Languages, Washington, DC, USA, IEEE Computer Society (1996)
5. Miller, G.A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review **63** (1956) 81–97
6. Conklin, J.: Hypertext: an introduction and survey. Computer **20**(9) (1987) 17–41
7. Anderson, J.R.: Cognitive Psychology and Its Implications. 6th edn. Worth Publishers (2005)
8. Vester, F.: Die Kunst vernetzt zu denken. Dtv (2002) Report to the Club of Rome.

[8] http://nepomuk.semanticdesktop.org

9. Dörner, D.: Die Logik des Mißlingens. Strategisches Denken in komplexen Situationen. Rowohlt Tb. (2003)
10. Buzan, T., Buzan, B.: The Mind Map Book: How to Use Radiant Thinking to Maximize Your Brain's Untapped Potential. Plume (1996)
11. Novak, J.D., Gowin, D.B.: Learning how to learn. Cambridge University Press, New York (1984) For a crisp overview on "The Theory Underlying Concept Maps and How To Construct Them" by J.D. Novak, see `http://cmap.coginst.uwf.edu/info/`.
12. Shipman, F., Marshall, C.: Spatial hypertext: An alternative to navigational and semantic links. ACM Comp. Surveys **31**(4) (1999)
13. Bederson, B.B., Hollan, J.D., Stewart, J., Rogers, D., Vick, D.: A Zooming Web Browser. In: Human Factors and Web Development, Second Edition. LEA, Inc. (2002)
14. Völkel, M., Haller, H.: Conceptual data structures (cds) – towards an ontology for semi-formal articulation of personal knowledge. In: Proc. of the 14th International Conference on Conceptual Structures 2006, Aalborg University - Denmark (2006)
15. Richter, J., Völkel, M., Haller, H.: Deepamehta - a semantic desktop. In Decker, S., Park, J., Quan, D., Sauermann, L., eds.: Proceedings of the 1st Workshop on The Semantic Desktop. 4th International Semantic Web Conference (Galway, Ireland). Volume 175., CEUR-WS (2005)
16. Furnas, G.W.: Generalized fisheye views. In: Human Factors in Computing Systems CHI '86. (1986) 16–23