REQUIREMENTS FOR PERSONAL KNOWLEDGE MANAGEMENT TOOLS

Max Völkel and Andreas Abecker
FZI Forschungszentrum Informatik Karlsruhe, Haid-und-Neu-Str. 10-14, Karlsruhe, Germany

Keywords: Personal Knowledge Management, Requirements, Literature.

Abstract: *Personal* knowledge management (PKM) is a crucial element as well as complement of *enterprise* knowledge management (EKM) which has been largely neglected by Enterprise Information Systems, up to now. This paper collects requirements for a specific class of PKM software, which supports personal note taking and the idea of extending the human memory by information management. It introduces the *knowledge-cue life cycle* which describes how *information* artefacts can be used for helping to denote, remember, use, and further develop *knowledge* embodied in people's heads. Based on this life cycle and on a literature study, this paper derives a comprehensive requirements catalogue to be fulfilled by knowledge articulation tools used in PKM. This requirements list can be used as a design specification and research agenda for PKM tool builders, and to assess the suitability of existing tools for PKM.

1 INTRODUCTION

As our globalized economy becomes increasingly more knowledge-based, Enterprise Knowledge Management (KM, c.f. Probst et al., 2006), (Nonaka and Takeuchi, 1995), (Abecker, 2004) has become a settled management discipline in the recent 15 years.

The most important area of knowledge creation and processing is still the most underdeveloped area of KM: namely that of the individual knowledge workers' personal knowledge.

Practically, the main goal of Personal Knowledge Management (PKM, c.f. Davenport, 2005), is to make the individual more productive - and thereby also the organisation as a whole. PKM investigates the use of methods and tools to amplify the abilities of the individual to work better with knowledge (Völkel, 2010). Typical use-cases in PKM are, e.g., note-taking, document creation, argumentation, idea management, or managing the personal social network. Many PKM tools have been built and even more tools are (ab-)used for PKM tasks. Hence the question arises how an ideal PKM system look like.

This paper considers the sub-class of PKM support systems which deal with articulating (parts of) the individual knowledge in the form of electronic artefacts.

2 PROCESSES IN PKM

Creating (semi-)formal knowledge is the act of *modelling*. This paper uses the term "**personal knowledge model**" to denote an artefact which represents a set of "knowledge cues". The knowledge cues can vary in size, structuredness and degree of formality.

A **knowledge cue** is either (a) a piece of content, containing plain text, semi-structured text, or arbitrary binary content such as images or desktop objects, or (b) a connection between other knowledge cues. Such connections can be unspecified relations, directed hyperlinks and formal statements.

This section introduces a novel process model for the management of knowledge cues. In PKM, organizer and retriever is the same person. Different from an organisational context, there is a personal motivation to organize knowledge. The user can freely trade efforts of authoring with efforts of retrieval.

An in-depth analysis of economic factors in PKM has been published by (Völkel and Abecker, 2008). Building on this work, a novel PKM process model, the *knowledge cue life-cycle*, has been created (c. f. Fig. 1). It describes ways in which an individual interacts with his personal knowledge model. It consists of ten processes:

1. *Create knowledge* mentally;

Völkel M. and Abecker A. (2010).
REQUIREMENTS FOR PERSONAL KNOWLEDGE MANAGEMENT TOOLS.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 332-337
DOI: 10.5220/0002969303320337

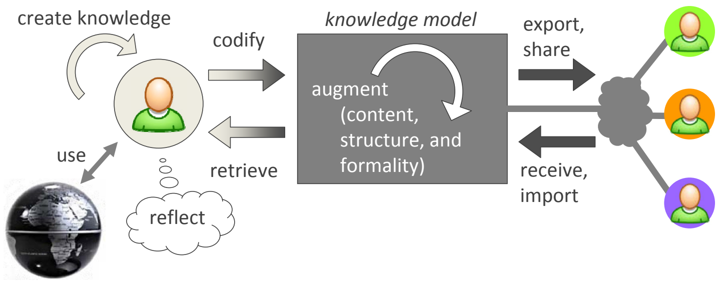

Figure 1: Knowledge Cue Life-Cycle.

2. *Codify* by creating initial knowledge cues;
3. *Augment* knowledge cues by adding more content, structure, or formality;
4. *Export* knowledge cues into other formats;
5. *Share* knowledge cues with others;
6. *Retrieve*[1] knowledge cues and evoke a previously experienced mind-state;
7. *Receive* knowledge cues into other formats and evaluate and filter them;
8. *Import* relevant and reviewed parts as knowledge cues;
9. *Use* knowledge in a real world situation;
10. *Reflect* on all knowledge cue processes.

3 REQUIREMENTS

This section lists requirements gathered from a variety of sources analysing PKM. [2]

Of course, no such list can be considered "complete" – there might be even more requirements.[3]

• **R1: Knowledge Model should be a Super-set of existing Conceptual Models.** To re-use content residing in one kind of representation in another tool, it needs to be transformed. Transformations between data-models come not for free. A naive approach to convert between n formalisms would require writing n^2 transformations. However, if a common intermediate formalism can be used, the costs come down to $2n$. Learning a new tool requires a user to understand the conceptual model of the tool. It is easier to learn a tool with a familiar conceptual model. Hence, the formalism of a good PKM tool should be similar to those of other tools in use.

A knowledge model should be expressive enough to represent existing application data models to enable re-use of existing external structures. To save costs in content transformation, the conceptual knowledge model should therefore be a *super-set of the conceptual models of all other relevant PKM tools*.

• **R2: System should Run Queries Automatically.** req:autoqueryauto-query The best personal knowledge model is of no value if a user needs some knowledge and simply forgets to search his personal knowledge model. Therefore, a PKM system should run queries automatically (R2). (Cutrell et al., 2006) proposes to automatically start a search when certain triggers are encountered. Knowledge cues relevant to the current task context should be delivered pro-actively (Schmidt, 2009), relevant to the current business process (Abecker et al., 2001).

• **R3: Fast Entry.** In order to minimize distractions by one's own creative new ideas or external interruptions (phone calls, instant messenger, or colleague coming in), one needs to quickly take a note and continue working at the previous task.

• **R4: Informal Articulation.** Users needs a simple way to express content in an informal way, e.g., as plain text, formatted text or box-and-arrow diagrams (Oren, 2006) or "this is *nested within* that, but I can't say why".

(Blandford and Green, 2001) studied the use of short personal notes for task work and found that informal tools like paper and unstructured text files

[1] This process can also search other people's knowledge models, if their knowledge cues are shared.

[2] For this paper, many sources have been used in a non-systematic manner (papers emailed from colleagues, followed forward- and backward references, search engines, ...). The following additional sources have been used systematically: (a)Proceedings of the Annual ACM Conference on Research and Development in Information Retrieval from the years 2005, 2006, and 2007. (b) Proceedings of SIGCHI. All years up to 2008 have been queried for "personal", "knowledge", and "management". (c) Proceedings of workshops on personal information management from 2006 until 2009.

[3] Due to very restricted space, a longer version of this paper is available from http://pubs.xam.de

were sometimes preferred over traditional PIM applications because they supported more freeform input.

• **R5: Formal Articulation.** Formal reasoning can help to reduce retrieval costs, when from a set of explicitly stated formal statements further formal statements can be inferred *automatically*.

• **R6: Let the User decide on Granularity of Modelling.** req:granularitygranularity Content varies greatly in size and type. Polanyi (Polanyi, 1998): "... linguistic symbols must be of reasonable size, or more generally that they must consist of easily manageable objects. [...] Language can assist thought only to the extent to which its symbols can be reproduced, stored up, transported, re-arranged, and thus more easily pondered, than the things which they denote."

(Shneiderman, 1989) reports on a comparative study in which two groups of people had to locate answers to a series of questions in a *Hyperties* database. The group with more (46 instead of 5), shorter (4-83 lines instead of 104-150 lines) articles answered more questions correctly and took less time to answer the questions.

• **R7: Entities need to be Addressable.** To be able to link two entities, they must be addressable. To be able to model different versions of an entity, each version needs a kind of address. In a personal knowledge model, each entity that is shown to the user should be addressable.

• **R8: Simultaneous Use of Different Levels of Formality.** People need to be able to work at any level of formality (unstructured, structured and formal knowledge), and to freely mix such levels (Lethbridge, 1991). For textual content this means exploiting syntax, structures and semantics. E.g., in semantic wikis (cf. (Völkel and Schaffert, 2006)) all three levels can be used. While typing text, there is *syntax* for formatting (bold, italic), *structures* (headlines, lists) and *semantics* (typed links).

• **R9: Stepwise Changes from Informal to more Formal Structures.** The user should be able to migrate the knowledge into more formal structures, if desired (cf. (Völkel and Haller, 2006)). A very important characteristic of formal knowledge engineering in general is the *modelling process*. During modelling, a knowledge model might be in an inconsistent state. A tool should not simply prevent such inconsistent states, but rather inform the user about the consequences. The migration from one consistent formal state to another consistent formal state can be a complex operation which cannot or should not need to be carried out completely in the mind of the user. Instead, a sequence of modelling operations should allow producing inconsistent states, and then, step-by-step, move towards the desired target state. Inconsistent states should also be share-able, so that a resolution can be found collaboratively, if desired.

(Oren, 2006) advises to focus on simply capturing and representing the things that the user wants to store, before doing any reasoning with it.

•t **R10: Knowledge Model Refactoring.** With stepwise formalization a user can gradually add more structure and formality to knowledge cues. Externalised personal knowledge artefacts are usually organised in a systematic manner, e.g., files are sorted in folders and sub-folders.

Unfortunately, a good structure today is not a good structure tomorrow, therefore personal organisation schemes change. Stuart K. Card in (Jones et al., 2006) sees this not only as a tedious maintenance task, but says "re-representation of information is a key to interpreting it". (Schreiber and Harbo, 2004) emphasises flexibility of knowledge models and the need for reorganisation.

• **R11: Versioning.** req:versioningversioning The cost of creating and manipulating knowledge cues is lower, if people have an easy way to undo their operations and revert to previous versions of a knowledge model.

• **R12: Capture the Context for Knowledge Cue Creation and Import** req:contextcontext Understand the notion of context, capture it together with the information and use it to enhance recall and understanding (Oren, 2006). Automatic context tracking should relive the user from maintaining bookkeeping data such as creation data of items or linking (R20) two items that are commonly used together (Graça Pimentel et al., 2000).

• **R13: Active Assistance in Maintenance Tasks.** Metadata about the usage of the knowledge cues by the system is required: How often did the knowledge cue appear in search results? How often has it been changed? When has the most recent statement been made about this knowledge cue? When was the last time this knowledge cue was used for inferencing? Such metadata can be used by the system to ask a user specifically and actively about the status of certain knowledge cues.

• **R14: Easy to Learn.** Each new tool has a learning curve that depends on the complexity of the underlying concepts and the user interface. A good user interface cannot compensate for an ill-designed underlying data model.

• **R15: Grouping of Items.** Users need composition for navigation (Frank, 1988). This allows, e.g., browsing and thereby narrowing down their view and allows discovering related, yet unexpected items.

It is important for a user to be able to group seemingly unrelated content together, so that retrieval of one item triggers retrieving of the others, too (Jones et al., 2005).

Grouping knowledge cues is also a pre-requisite for any kind of batch operations such as exporting, sharing, refactoring, delete and copy operations.

• **R16: Containment Relationship.** A containment relationship is a stronger form of grouping with additional semantics for operations. Delete and copy commands on containers trigger recursively the same command for all *contained* elements.

• **R17: Optional Naming of Knowledge Cues.** The data model should allow giving things human-usable names. A name is understood as a *unique name*. Names make linking much easier, as the link target name can simply be typed and one has not to select from a complex GUI. Names also allow direct navigation deep into a knowledge model.

But as users often have difficulties to find names (Boardman, 2004)[p. 105], (Frank, 1988) advises to not *require* a user to name all items.

• **R18: Alternative Names.** Many systems with unique names have also means to create additional *alias* names, which redirect to another unique name.

• **R19: Order Knowledge Cues** Ordering a collection of ideas or text snippets into a coherent flow is one of the main tasks of authoring (Esselborn-Krumbiegel, 2002). A user should be able to create order gradually and partially. Note how different this is from providing a list data-structure: A list can only represent a total order.

• **R20: Linking.** (Oren, 2006) finds “an under-utilization of the interlinked nature of the information”. Knowledge models should allow for precise and effective linking – and browsing (R27.

• **R21: Hierarchy.** (Shneiderman, 1996) emphasizes the need to get “Overview first, zoom and filter, then details-on-demand.”

Hierarchies of all kind are commonly used in user interfaces to let the user narrow down his interests step-by-step.

• **R22: Simultaneous Use of Multiple Levels of Detail.** Users need ways to see multiple levels of detail at once (Frank, 1988).

• **R23: Annotating Content.** When using documents, a field study of (O’Hara and Sellen, 1997) concludes that annotating documents is frequently a part of the document reading and understanding process.

• **R24: Tagging.** Tagging is the basic assignment of easy-to-type keywords to information artefacts. Tag names contain usually no whitespace and tend to be really short. A common representation is a *tag cloud*, showing all tag names at once, with a font size proportional to their usage frequency. People have problems in using strict hierarchies (Oren, 2006). Therefore less strict methods such as tagging (R24) and categories (R25) are required.

• **R25: Classifying into Categories.** Categories differ from tags: Categories tend to have longer, encyclopedia-like names. In most category-systems, there is a weak hierarchy, i.e., categories can often be nested into (several) other categories. In practice, the boundaries between tags (short, easy-to-type, not nested) and categories (nice to read, nested) are blurred sometimes.

• **R26: Queries.** Besides browsing a user also needs the ability to search and query the data (Frank, 1988). A number of different queries are required (fulltext queries, structured queries, aggregating queries, metadata queries, and formal queries).

• **R27: Following Links and Browsing Collections.** Following links is one of the three core strategies of information retrieval described by (Bates, 2002). There is a fundamental difference between search (where you know what you are looking for) and browsing (where you find things that you placed there) (Jones et al., 2005).

• **R28: Inverse Relations.** Many wikis allow traversing hyperlinks not only forward, but also in backward mode. For each page they list all pages linking to the page. In most semantic GUIs incoming links are rendered different from outgoing links. Therefore it makes a difference for browsing whether a user stated ([SAP], [employs], [Dirk]) or ([Dirk], [works for], [SAP]). For the user, this is often an artificial distinction. It is desirable that link types have labels for both directions, e.g., “works for” and “employs”.

• **R29: Flexible Schema.** The survey paper of (Oren, 2006) states a requirement for flexible schemas: Leave users their freedom and do not constrain them into rigid schemas. This is also relevant for importing from other data models to be able to represent as much of the given structure and formality as possible.

• **R30: Transclusion.** Embedding a reference and rendering the content is called *transclusion*. The need for transclusion is further explained by (Ludwig, 2005), (Nelson, 1995) and in the evaluation of Popcorn described by (Davies et al., 2006).

• **R31: Meta-modelling.** If knowledge cues become old, but not outdated, they become just harder to understand. The meaning of terms shifts. It is therefore required to let the user describe and annotate all aspects of knowledge cues. Even annotation on annotations, statements and relations are sometime required. This allows a user to create a more self-describing knowledge model. The data model must allow annotating (and therefore addressing) all of its elements, in order not to limit expressivity.

4 CONCLUSIONS

This paper investigated requirements for Personal Knowledge Management tools. A knowledge-cue life cycle was introduced, which describes how knowledge workers use tools to create information artefacts that help them to express, remind, share, discuss, use, and further develop their personal knowledge. Based on the ten processes, an exhaustive requirements list has been compiled from existing literature.

Some conclusions and observations can be drawn from this list of requirements: (1) Ultimately, a PKM tool must be a general purpose modelling tool: It must have a rather generic (i.e., not restricted to a particular domain, R29) data model (R1) with different levels of formality (R4, R5, R8) and granularity (R6). It should allow a user to model a number of conceptual constructs, namely order (R19), hyperlinks (R20), hierarchy (R21), and annotations (R23).

(2) Even more advanced modelling features such as inverse relations (R28) and meta-modelling (R31) are desirable. (3) Some requirements 2, 10, 13, and 22 can only be realised in a tool, i.e., a model by itself cannot 'run queries automatically' (R2), only a tool can actually *run* something actively.

The derived list of requirements can be used: (1) to guide the design of future PKM systems as well as the underlying research; as well as for (2) evaluating the adequateness of existing tools for PKM.

Along the requirements described in this paper, an initial web-based prototype system has been developed and evaluated. It is described in an upcoming dissertation by (Völkel, 2010). A desktop-based system, *iMapping*, tackling the same requirements is currently being developed by (Haller, 2006).

REFERENCES

Abecker, A. (2004). *Business-Process Oriented Knowledge Management: Concepts, Methods and Tools*. PhD thesis, Universität Karlsruhe (TH), Institut AIFB, D-76131 Karlsruhe.

Abecker, A., Bernardi, A., Ntioudis, S., Mentzas, G., Herterich, R., Houy, C., Müller, S., and Legal, M. (2001). The decor toolbox for workflow-embedded organizational memory access. In *ICEIS (1)*, pages 225–232.

Bates, M. (2002). Speculations on browsing, directed searching, and linking in relation to the bradford distribution. In *Emerging frameworks and methods: Proceedings of the Fourth International Conference on Conceptions of Library and Information Science (CoLIS 4)*, pages 137–150, Greenwood Village, CO. Libraries Unlimited.

Blandford, A. E. and Green, T. R. G. (2001). Group and individual time management tools: What you get is not what you need. *Personal Ubiquitous Comput.*, 5(4):213–230.

Boardman, R. (2004). *Improving Tool Support for Personal Information Management.* PhD thesis, Imperial College, London.

Cutrell, E., Dumais, S. T., and Teevan, J. (2006). Searching to eliminate personal information management. *Communications of the ACM*, 49(1):58–64.

Davies, S., Allen, S., Raphaelson, J., Meng, E., Engleman, J., King, R., and Lewis, C. (2006). Popcorn: the personal knowledge base. In *Conference on Designing Interactive Systems*, pages 150–159.

Esselborn-Krumbiegel, H. (2002). *Von der Idee zum Text. Eine Anleitung zum wissenschaftlichen Schreiben.* Utb. 2. Auflage.

Frank, G. H. (1988). Reflections on notecards: seven issues for the next generation of hypermedia systems. *Communications of the ACM*, 31(7):836–852.

Graça Pimentel, M. d., Abowd, G. D., and Ishiguro, Y. (2000). Linking by interacting: a paradigm for authoring hypertext. In *HYPERTEXT '00: Proceedings of the eleventh ACM on Hypertext and hypermedia*, pages 39–48, New York, NY, USA. ACM Press.

Haller, H. (2006). iMapping – a graphical approach to semi-structured knowledge modelling. In Rutledge, L., editor, *Proceedings of the The 3rd International Semantic Web User Interaction Workshop (SWUI2006)*. Poster presented at the The 3rd International Semantic Web User Interaction Workshop.

Jones, W., Phuwanartnurak, A. J., Gill, R., and Bruce, H. (2005). Don't take my folders away!: organizing personal information to get things done. In *CHI Extended Abstracts*, pages 1505–1508.

Jones, W., Pirolli, P., Card, S. K., Fidel, R., Gershon, N. D., Morville, P., Nardi, B. A., and Russell, D. M. (2006). "it's about the information stupid!": why we need a separate field of human-information interaction. In *CHI Extended Abstracts*, pages 65–68.

Lethbridge, T. C. (1991). A model for informality in knowledge representation and acquisition (an extended abstract). In *Workshop on Informal Computing, Santa Cruz, Incremental Systems*, pages 175–177.

Ludwig, L. (2005). Semantic personal knowledge management. Technical Report D11.01_v0.01, DERI Galway, University Road, Galway, Ireland.

Nelson, T. H. (1995). The heart of connection: hypermedia unified by transclusion. *Communications of the ACM*, 38(8):31–33.

Nonaka, I. and Takeuchi, H. (1995). *The Knowledge-Creating Company : How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press.

O'Hara, K. and Sellen, A. (1997). A comparison of reading paper and on-line documents. In *CHI*, pages 335–342.

Oren, E. (2006). An overview of information management and knowledge work studies: Lessons for the semantic desktop. In Decker, S., Park, J., Sauermann, L., Auer, S., and Handschuh, S., editors, *Proc. of the Semantic Desktop and Social Semantic Collaboration Workshop*, volume 202 of *CEUR Workshop Proceedings ISSN 1613-0073*. CEUR-ws.org.

Polanyi, M. (1998). *Personal Knowledge*. Routledge.

Probst, G., Raub, S., and Romhardt, K. (2006). *Wissen Managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen*. Gabler Verlag, 5 edition.

Schmidt, A. (2009). *Situationsbewusste Informationsdienste für das arbeitsbegleitende Lernen*. Karlsruhe Institute of Technology / Universität Karlsruhe (TH), Karlsruhe, Germany. http://digbib.ubka.uni-karlsruhe.de/volltexte/1000012939.

Schreiber, T. and Harbo, K. (2004). Information literacy and personal knowledge management. In *12th Nordic Conference on Information and Documentation*, pages 106–114.

Shneiderman, B. (1989). *The Society of Text*, chapter Reflections on authoring, editing, and managing hypertext, pages 115–131. MIT Press, Cambridge, MA, USA.

Shneiderman, B. (1996). 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.

Völkel, M. (2010). *Personal Knowledge Models with Semantic Technologies*. PhD thesis, Fakultät für Wirtschaftswissenschaften, Institut AIFB. To appear.

Völkel, M. and Abecker, A. (2008). Cost-benefit analysis for the design of personal knowledge management systems. In *ICEIS 2008 - Proceedings of the Tenth International Conference on Enterprise Information Systems, Volume AIDSS, Barcelona, Spain, June 12-16, 2008*, pages 95–105.

Völkel, M. and Haller, H. (2006). 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.

Völkel, M. and Schaffert, S., editors (2006). *Proceedings of the First Workshop on Semantic Wikis at ESWC, Budva, Montenegro.*, volume 206 of *CEUR Workshop Proceedings ISSN 1613-0073*. FZI Forschungszentrum Informatik Karlsruhe, CEUR-ws.org. http://CEUR-WS.org/Vol-206 (accessed 06.01.2010).