LHRS07-2] *Lasse Lehmann, Tomas Hildebrandt, Christoph Rensing, Ralf Steinmetz;* ***Utilizing Lifecycle Information for Knowledge Document Management and Retrieval;*** Klaus Tochtermann, Hermann Maurer: Proceedings of the 7th International Conference on Knowledge Management, J. UCS, September 2007, S 192-199

Utilizing Lifecycle Information for Knowledge Document Management and Retrieval

Lasse Lehmann, Tomas Hildebrandt, Christoph Rensing and Ralf Steinmetz

KOM Multimedia Communications Lab
Technische Universität Darmstadt
Merckstr. 25, 64283 Darmstadt, Germany

{lasse.lehmann, christoph.rensing, ralf.steinmetz}@kom.tu-darmstadt.de

Abstract: Classical approaches to document management do not cope with the demands knowledge documents make. Knowledge documents or so-called "living documents" have a far more complex lifecycle than general documents. They are usually used, edited und utilized by several people and many different versions, revisions and variants exist. Today the multitude of information that these processes generate are not captured or used to provide a better management or retrieval for this kind of documents. Our approach shows that the capturing of this lifecycle information can help in the retrieval as well as usage and management of those documents.

Keywords: Document Management, Lifecycle Information, Document Lifecycle, Retrieval
Categories: H.3.1, H.3.2, H.3.4

1 Introduction and Motivation

Imagine the following scenario: A lecturer at a university wants to make slides for his new lecture. Thus he searches for related material and finds lectures from other professors and lecturers that cope with similar topics. He copies some of the retrieved slides directly into his new presentation, while other parts are edited or reviewed before being put into the new lecture slides. He removes parts from the retrieved slides and adds new slides. When he is done he puts an acknowledgement at the end of his slides to give credits to those he got the original slides from. This is a very common scenario, so that it can be assumed that the professors our lecturer got his slides from proceeded in a similar way during the creation of their slides. There are several things the authors of the other slides would appreciate to know. To name a few of them:

- Who is reusing my slides?
- Who has changed the slides in which way?
- In which context are the slides re-used?
- How often are the slides re-used?
- Are there lecturers teaching similar topics and do my slides match theirs?
- How popular are my slides?
- ...

If the lifecycle information for all these documents would have been collected during the aforementioned processes, these questions could be answered easily. On top of that there would be a whole network of information and relations connecting the various versions and variants of the different lectures and slides for the different authors.

Similar scenarios can be found in a corporate context. Here, knowledge workers correspond mainly with other employees of the same enterprise. However, the lifecycle information that can be gained can be of great help here, too. E.g. if one employee has a specific task to do which is new to him, he searches for related material, which he can learn from. Often, the knowledge he is searching for is distributed over a wide range of different documents, some of them in places, where the employee might not have thought of looking at. The lifecycle information helps him to find and structure the information and knowledge he needs and allows him to browse the network of relations connecting the documents.

We will show an approach to collect the lifecycle information for knowledge documents and provide possibilities and concepts for the utilization of the collected information. At first we will define our concept of knowledge documents and analyse their lifecycle and the information that can be collected herein in section 2. Section 3 deals with the capturing and utilization of lifecycle information. Section 4 shows a concept and architecture for a system that implements our approach while section 5 deals with related work. Section 6 concludes this article and gives an outlook at future work.

2 Knowledge Documents and Their Lifecycle

First of all, we need to make clear how we comprehend the concept *knowledge document*. We do not want to constitute an exact definition for knowledge documents. In our understanding it is more like we found a group of documents with certain attributes and tried to find a concept to name this type of documents. Our definition is based on the well known DIKW model - e.g. [Davenport, 99], which states that simple characters, data, information, knowledge and wisdom stand in a defined hierarchical order with accumulating constraints. The exact constraints are perceived differently in existing literature; in our definition, characters become data when syntax is added, data becomes information when context is added and information becomes knowledge when the cross-linking of information is added. Starting there from, a knowledge document is a document with the following attributes:

- It contains knowledge (in contrast to containing data or information only)
- It is mainly text-based but can be a compound of different media and text
- It has a processible format (could be proprietary, like .doc or .ppt, too)
- It is often a result of a collaborative authoring process
- It is re-used or re-purposed (and undergoes the according processes)

Therefore knowledge documents are a subset of documents in general. In our understanding typical knowledge documents are, depending on the context, e.g.

lecture slides, technical reports, documentations, papers, scientific works or internal reports like shown in the aforementioned examples. Existing literature often refers to "living documents" [Berndt, 05] which is a quite similar concept from our point of view, but even vaguer than the term "knowledge document". Learning Objects can be knowledge documents as well, but due to their loose definition it is not possible to state that all learning objects are knowledge documents. We do not consider documents like bills, schedules, calendars or other documents often used in business processes as knowledge documents, since they contain mainly information.

2.1 Context Information

We distinguish two types of lifecycle information. **Context information** is related to one document. It covers the different contexts the document traverses during its history, e.g. applications that opened the document, users who accessed the document, the number of times a document was searched for, downloaded or retrieved respectively as well as the context of creation or usage (in terms of used applications durations of use etc.). Context information can be generated implicitly. This kind of information can be collected automatically by monitoring the used applications. On the other hand context information can also be user-generated, like e.g. feedback users give to the authors of certain documents, annotations to documents or parts of documents or even reviews or ratings.

2.2 Relation Information

The second type of lifecycle information we consider is **relation** information. Relations connect two or more knowledge documents and occur as result of explicit user actions where new instances of that knowledge documents emerge. While capturing these relations, we do not want to solely capture the fact that a relation exists, but also the type of relation connecting the two instances. We defined a specific set of relations for SCORM [ADL, 07] compliant learning objects [Lehmann, 07]. However, these relations can not be transferred to the domain of knowledge documents without modification. Thus, a future step will be the definition of types for relations emerging between knowledge documents in the course of their lifecycle. The lifecycle itself is analysed in the following.

2.3 Lifecycle Model

There are several existing models for the lifecycle of documents in general - most of them from the area of Information Lifecycle Management (ILM). As the name implies, do these models handle documents containing *information*. As stated before, we want to consider documents containing *knowledge*. Such, the prerequisites for lifecycle models are different. In the area of living documents or knowledge documents there are very few approaches for the modelling of the lifecycle. In [Ginsburg, 99] a waterfall model for the "Intranet Document Lifecycle" is presented. However, it does not focus on the systems and phases important for knowledge documents. In the area of e-Learning there are some approaches like [Brooks, 06] or [McCalla, 04] that at least state that the lifecycle has to be taken into account. A specific lifecycle model for learning objects is presented in [Collis, 04]. Since knowledge documents are a subset of documents in general, the lifecycle model for

knowledge documents should be compatible with the existing lifecycle models from ILM or digital libraries. However, it should provide a meaningful view on the relevant phases of the lifecycle from different perspectives. This enables us to analyse the information being generated or utilized during the different phases.

Figure 1 shows a schematic model that is suitable for the lifecycle of knowledge documents and stresses the phases where lifecycle information is relevant. The arrows show where lifecycle information is being generated and where it can be utilized. The dotted arrows represent relation information while the other ones mean context information. The lifecycle in our schema has no defined beginning or end with exception of the creation phase. Here, a **creation** of a new document "from scratch" is meant, where no relations to other documents exist, thus it is the starting point to get into the cycle. Nevertheless, this is quite seldom the case, since most new documents are created by re-using, aggregating, re-purposing or in any other manner modifying existing documents.

Therefore the most common starting point is the **Access / Retrieval** phase, where the documents being re-used are retrieved. This phase holds for a multitude of context information like occurrences of the document in search results, downloads / copies of the document or ratings if available. When a document is copied a relation between the two instances of this document emerges. In turn, this phase is suited well to use the gathered information to support the user in retrieving desired documents.

The actual creation as it takes place in most of the cases is represented by the **Edit / Re-Use / Aggregate / Re-Purpose** phase. While this phase also holds for the emergence of context information, like applications used or persons who worked on a document, this is the main phase for the generation of relation information. Through the different authoring processes different kinds of relations emerge.

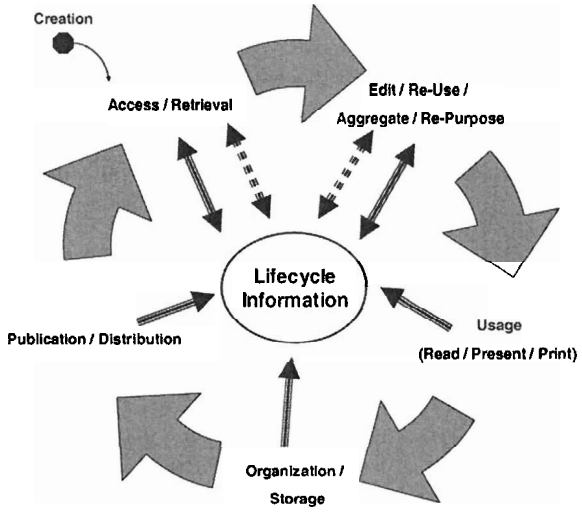

Figure 1: Schematic Knowledge Document Lifecycle

The **Usage** phase depicts the actual use of the document, where the document serves the desired purpose. Context information like e.g. the time a presentation took can be gotten here.

During the **Organization / Storage** phase, the document is stored. Context information can e.g. be gotten from the directory tree. The harvesting of e-mail context would be done in this phase, too.

During the **Publication / Distribution** phase, the document is made available to other persons. Naturally, the labelling of the document would take place during this phase, providing context information. We did not consider a destruction or archiving phase, because we think that this is not relevant for our purpose. The phases shown in Figure 1 may be mostly traversed in the given order; however, in reality interconnections between the different phases do exist but are left out for the sake of clarity.

3 Capturing and Utilization of Lifecycle Information

Our intention is to collect the information at the moment when it emerges. Therefore it can only be captured within the systems and applications that are part of a knowledge document's lifecycle. For the **capturing** there are normally two possibilities: If we have full control over the application, because the application is developed by us or it is an open source application, the capturing can take place from within the source code of the application. Otherwise we have to rely on event-handlers, which are called when different user actions are taken within the application, and accordantly capture the collectable information. Examples for information that can be captured:

Number of downloads from a document storage
Explicit feedback information from other users

- Version- and instance relations
- Father-/child relations
- Part-of relations
- Structural relations (permutation, successor, predecessor, etc.)

For the **utilization** of lifecycle information, similar things hold true. We want to present the right parts of processed lifecycle information to the user in moments when he or she can benefit from it the most. Therefore the information has to be integrated into the application the user currently works with. This will require in most cases plug-ins, too. In [Lehmann, 07] we presented a system where we implemented the utilization of lifecycle information for the ranking of search results in an authoring-by-aggregation environment for Learning Resources as well as the recommendation of related resources. For knowledge documents, we plan - among others - the following ways of utilization:

Ranking (by number of downloads, re-usability, ...) - *retrieval phase*

- Recommendations of related documents - *retrieval/authoring phase*
- Search without query concepts (via relations) - *retrieval phase*

- Visualisation of a the existing document network - *retrieval phase*
- Tracking of a document's influence on other documents - *retrieval/authoring phase*
- Provision of feedback from other users - *authoring phase*
- Notifications (changes, updates, new additions to the document network) - *authoring phase*
- Automatic generation of acknowledgements - *authoring phase*

4 Concept and Architecture

Figure 2 shows the architecture for a system implementing the aforementioned approach. It is a client / server based system. The **Lifecycle Management System (LMS)** on the server side stores and processes the collected lifecycle information. It stores the collected information with reference to the accordant documents stored in the backend. It provides a web service API that the **Capture- and Utilization Plug-ins** (CP & UP) within the applications make use of to get the processed information from the LMS or to send the collected information back respectively. The **Backend** is hold flexible. Different kinds of document stores can be connected to the LMS that distributes the IDs and holds the lifecycle information for every document.

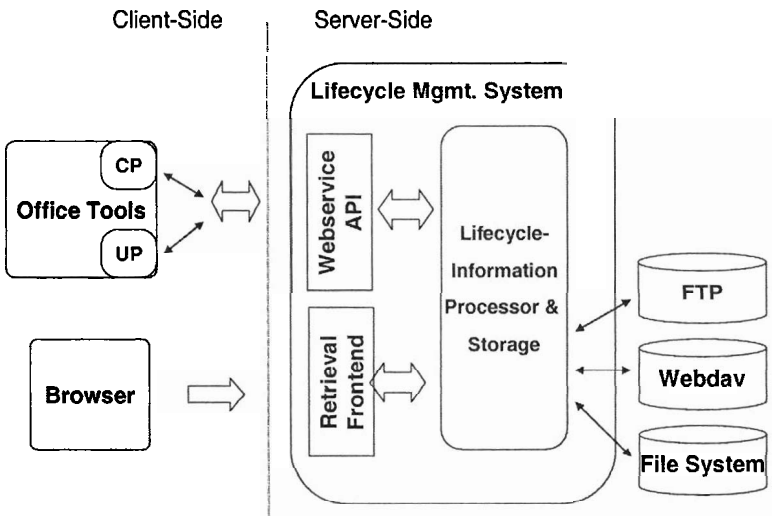

Figure 2: Conceptual Architecture

We plan to integrate a web front-end into the LMS that enables the users to retrieve documents from the connected backend solutions. In such a case we have the full control over the retrieval service and thus can fully support the user with additional information. The processing of the collected lifecycle information includes, among others, the distribution of new information to involved documents, checking of relations to avoid dead links and the enrichment of relations via algebraic rules.

In [Lehmann, 07b] we propose an architecture for the management of lifecycle information for Learning Resources. In contrast to that, this one copes with the special demands, knowledge documents make.

Besides the here described conceptual architecture there are several existing applications and frameworks where lifecycle information would be a nice feature. Examples are so called Semantic Desktops, which are basically desktop search and retrieval application where users can describe and structure their content and documents, like described in [Sauermann, 05]. Personal Learning Environments [Wilson, 06] might benefit from the capturing and utilization of lifecycle information, too. A P2P version of the approach, where users can share knowledge documents and the collected lifecycle information and thus create huge networks of interconnected documents, is planned, too.

5 Related Work

There are several approaches, where the utilization of lifecycle information is partly conducted. This applies especially for context information. In [Najjar, 06] the Attention Metadata system is described, where context information and user behaviour is tracked to gain information about learning objects. In [Ochoa, 06] an approach is proposed, how this information can be used to rank and recommend search results for learning objects. These approaches mainly deal with learning resources and do not take the relations that connect different instances and variants of knowledge documents or learning resources into account. In [Chirita, 05] context metadata is used to get search results for desktop search. Here the context information is gotten from the e-Mail context, file system information and the browser cache. Again, relation information is neglected. Known commercial platforms like eBay or Amazon utilize special kinds of context information to rank and recommend items. This has been a focus of research several years ago, e.g. in [Good, 99].

Systems and approaches that utilize the relations between knowledge documents are rather sparse. In [Mueller, 06] a system for a management of change for collaborative authoring of structured and unstructured documents is proposed. This is somehow build upon relations but mostly relies on semantic, intra-document relations that are inevitable for a consistent management of change. Nevertheless this might be an interesting addition to the lifecycle information gathered by us.

6 Conclusions and Future Work

This article shows that lifecycle information can be of great importance for the retrieval and management of knowledge documents. A schematic lifecycle model for this kind of documents was proposed and the described architecture has shown how lifecycle information of knowledge documents can be managed. The next step is the refinement of the proposed schema along with the analysis of the different types of context and relation information, before the system can be implemented. In further works, a schema for the storage and management of the gathered information has to be found. We think that lifecycle information can improve the management and retrieval of living documents significantly.

References

[ADL, 07] Advanced Distributed Learning, SCORM - Shareable Content Object Reference Model, http://www.adlnet.gov/scorm/index.aspx

[Berndt, 05] Berndt, O.; Biffar, J. & Zoeller, B. (2005), DMS – vom elektronischen Archiv zum Enterprise-Content-Management aus VOI: Code of Practice - Dokumenten-Management, VOI, chapter 1.

[Brooks, 06] Brooks, C. & McCalla, G. (2006), Towards Flexible Learning Object Metadata, *in* 'In Proceedings of Int. J. Cont. Engineering Education and Lifelong Learning, Vol16, Nos 1/2'.

[Chirita, 05] Chirita, P.; Gavriloaie, R.; Ghita, S.; Nejdl, W. & Paiu, R. (2005), Activity Based Metadata for Semantic Desktop Search, in 'Proceedings of the 2nd European Semantic Web Conference'.

[Collis, 04] Collis, B. & Strijker, A. (2004), 'Technology and Human Issues in Reusing Learning Objects', *Journal of Interactive Media in Education* **4**.

[Davenport, 99] Davenport, T.H. & Prusak, L. (1999), *Working Knowledge: How Organizations Manage What They Know*, Harvard Business School Press.

[Ginsburg, 99] Ginsburg, M. (1999), 'An Agent Framework for Intranet Document Management', *Autonomous Agents and Multi-Agent Systems* **2**, 271 - 286.

[Good, 99] Good, N., Schafer, J.B., Konstan, J.A., Borchers, A., Sarwar, B., Herlocker, J. & Riedl, J. (1999), Combining Collaborative Filtering with Personal Agents for Better Recommendations, *in Proceedings of the 'AAAI/IAAI.*

[Lehmann, 07] Lehmann, L.; Rensing, C. & Steinmetz, R. (2007), Capturing, Management and Utilization of Lifecycle Information for Learning Resources, *accepted at* 'EC-TEL 2007'.

[Lehmann, 07b] Lehmann, L.; Hildebrandt, T.; Rensing, C. & Steinmetz, R. (2007), Lifecycle Information Management and Utilization in an Authoring by Aggregation Environment, *in* 'Proceedings of the Edmedia 2007'.

[McCalla, 04] McCalla, G. (2004), 'The Ecological Approach to the Design of E-Learning Environments: Purpose-based Capture and Use of Information About Learners', Journal of Interactive Media in Education 7.

[Mueller, 06] Mueller, N. (2006), An Ontology-Driven Management of Change, in 'In Wissens- und Erfahrungsmanagement, LWA (Lernen, Wissensentdeckung, Adaptivitaet) conference proceedings'.

[Najjar, 06] Najjar, J.; Wolpers, M. & Duval, E. (2006), Attention Metadata: Collection and Management, in 'Proceedings of the WWW2006 workshop on Logging Traces of Web Activity: The Mechanics of Data Collection, Edinburgh, Scotland'.

[Ochoa, 06] Ochoa, X. & Duval, E. (2006), Use of Contextualized Attention Metadata for Ranking and Recommending Learning Objects, *in* 'Proceedings of the CAMA 2006'.

[Sauermann, 05] Sauermann, L.; Bernardi, A. & Dengel, A. (2005), Overview and Outlook on the Semantic Desktop, *in* 'Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference'.

[Wilson, 06] Wilson, S.; Liber, O.; Johnson, M.; Beauvoir, P.; Sharples, P. & Milligan, C. (2006), Personal Learning Environments: Challenging the dominant design of educational systems, *in* 'Proceedings of EC-TEL'.