[LRS08-3] *Lasse Lehmann, Christoph Rensing, Ralf Steinmetz;* **Capture of Lifecycle Information in Proprietary Office Applications.** In: David Massart, Jean-Noël Colin and Frans Van Assche: Proceedings of the Second Second International Workshop on Search and Exchange of e-le@rning Materials (SE@M'08), CEUR-WS, September 2008. ISSN 1613-0073.

Capture of Lifecycle Information in Proprietary Office Applications

Lasse Lehmann, Christoph Rensing, Ralf Steinmetz

KOM - Multimedia Communications Lab
Technische Universität Darmstadt
Merckstrasse 25
64283 Darmstadt

{Lasse.Lehmann, Christoph.Rensing, Ralf.Steinmetz}@kom.tu-darmstadt.de

Abstract. Lifecycle information like e.g. relations resulting from re-use of Learning Resources and knowledge documents can be utilized to support search, retrieval and authoring of Learning Resources. We describe challenges that have to be faced when capturing lifecycle information in proprietary office applications like PowerPoint and show how we solved them in the implementation of ReCap.KOM add-ins for our LIS.KOM framework. Results of first evaluations are promising.

Keywords: Metadata, Capture, PowerPoint, Re-Use Tracking, Relations

1 Introduction

When creating Learning Resources authors do usually re-use or re-purpose existing documents or parts of them. This specifically applies to Learning Resources created with office applications like MS PowerPoint or Word. When creating slides most authors start with a search for existing presentations created by themselves or colleagues in order to re-use these. Processes like re-using, merging, adapting and re-creating Learning Resources provide for emergence of a multitude of information [2]. This information includes relations that connect Learning Resources, documents and media objects which have been involved in re-use processes. E.g. when an author takes slides from one presentation to create a new one both presentations are connected somehow. This information can be very helpful for retrieval of Learning Resources, e.g. by providing links to related resources or as input for ranking and recommendation methods. Additionally it can be used to support authors of Learning Resources, e.g. by notifying them when related resources are changed. Since this kind of information is bound to specific processes it can only be captured during these processes. However, most existing systems do not capture this kind of information and thus it is lost. We propose that this so called lifecycle information should be captured. In order to achieve this, corresponding processes, i.e. the users' actions in respective applications have to be monitored. This is not always an easy task to do since most widespread applications are usually proprietary. We designed and implemented a framework for the capture, management and utilization of lifecycle information [2]. The contribution of this paper is the identification and solution of challenges of lifecycle information capturing in proprietary applications by means of ReCap.KOM plug-ins for our LIS.KOM framework. The framework basically consists of a central server (the LIS.KOM Server), which collects and manages lifecycle information from various clients. On each client the collected information is cached and preprocessed by the LIS.KOM Client. The ReCap.KOM plug-ins do the actual capture of lifecycle information, specifically relation information, in office applications like PowerPoint, which will be the application scenario considered in this paper. First we describe what kinds of information can be captured in PowerPoint along with a quick look at closely related work (Section 2). After the description of the challenges we faced (Section 3) we will show how we solved them in our implementation (Section 4).

2 Lifecycle Information in Office Applications

In the following we shortly describe approaches related most to our contribution. For a more detailed discussion of related work in this area we refer to [2] and [3].

One part of the *ALOCOM* system [4] is a PowerPoint add-in that helps users to re-use slides by enabling an online search in several repositories. In addition to that it captures user behaviour (attention) in PowerPoint and provides it to the CAM (Contextualized Attention Metadata) framework [5]. When new documents are added to the system, ALOCOM decomposes them and captures re-use relations for the newly added objects. These relations are then used to e.g. rank search results within ALOCOM. However a monitoring of user actions outside of ALOCOM is not conducted. In *TeNDaX* [1], a system for collaborative creation and editing of documents, user actions are stored as transactions in a database. Thus it is possible to track copy and paste relations between documents. Nevertheless, to make this possible the TeNDAX editor has to be used.

Slide presentations, which our focus is on here, have naturally a high potential of re-use. The basic data model of presentations is modular due to the segmentation of a presentation into single slides and the segmentation of slides into shapes and textboxes. Thus, re-use relations can be tracked with a very fine granularity. It can be assumed - and our recently done first evaluation [3] has given evidence that this is true - that the creation of many presentations starts with searching for existing presentations and taking slides out of them, or using tools like ALOCOM [4] to otherwise find existing slides to re-use. Selected slides are then put together, adapted and joined with newly created content and media objects like pictures to form a new presentation. Lifecycle information that can be captured during the creation of PowerPoint presentations includes:

- Relations resulting from PowerPoint internal re-use of slides, shapes or text
- Relations resulting from external re-use of content from other documents (e.g. Word documents) or websites.
- Relations resulting from re-use of media objects (like e.g. pictures), including aggregation relations connecting the presentation with the media

objects themselves (e.g. by "isPartOf") and "secondary relations" between two presentations (re-)using the same object.

- Variant or identity relations connecting different presentation instances (e.g. when a presentation is saved under a different name)
- Client information (user information etc.) and time stamps

If this information is captured it can be made available to be utilize. For this purpose there are two main scenarios imaginable. The first possibility is utilization of lifecycle information within the respective application. This could in case of PowerPoint be an add-in that gives access to every presentation related to the one currently opened. The other possibility is a stand-alone application. This could be a search tool or explorer, which visualizes relations between documents and uses them to recommend documents or rank search results. Naturally these examples are not exhaustive.

3 Challenges

On our way towards integrative and transparent capture of lifecycle information in PowerPoint we had to cope with five main challenges.

Get all relevant events that provide for the emergence of relations: It is a quite complex task to do information capturing within a proprietary application without access to its source code. However, because of the high coverage of PowerPoint compared to open source solutions we decided to give it a try. To achieve our goals we make use of add-ins that are plugged into PowerPoint and utilize the API provided by Microsoft. This API is quite powerful, at least with respect to manipulating a presentation programmatically. Nevertheless the events which can be caught with the API are far from sufficient for our purposes. There is for instance an event thrown when a new slide is added, but not when a slide is deleted.

Get all information needed to depict a relation properly: It has to be guaranteed that enough information is available to construct a valid relation. In some cases, e.g. when an external object is re-used in PowerPoint the API does not provide enough information about this object to depict the relation properly. Thus other possibilities have to be found.

Handle changes done by users without the respective add-ins: Another challenge is to keep the state of captured information consistent with the state of the presentation, e.g. when a re-used slide is deleted from a presentation the respective relation should be deleted, too. This is particularly complex when *undo* and *redo* actions come into play.

Keep the state of the captured information synchronous to the actual presentation state: We have to cope with changes that occur when PowerPoint is not running, since PowerPoint add-ins naturally only work while PowerPoint is running. This includes changes in the file system, like renaming, deletion or movement of involved files.

Handle events and changes occurring while PowerPoint is not running: We can not always assume a closed scenario where every user has the required add-ins installed. Thus we have to handle changes done to the content of a document, which we have not been able to monitor at least in a way that collected information is kept valid.

4 Implementation

The implementation presented here is part of the LIS.KOM framework. Figure 1 depicts the components responsible for capture of information in different applications. The ReCap.KOM add-in represents the main component for the capture of information. In order to receive enough events, standard PowerPoint events are combined with low-level mouse and keyboard hooks as well as monitoring of Clipboard activity. From PowerPoint we get - besides standard events like "Saved", "Opened" or "Closed" - events when specific buttons or menu items are clicked (e.g. "Insert", "Copy", "Delete", "Undo", etc.). The keyboard hook triggers method calls when specific keys or combinations ("CTRL+C", "CTRL+V" etc.) are pressed and the Clipboard monitor notifies us of changes on the Clipboard and provides us with the actual Clipboard content. With a combination of these three methods it is possible to monitor almost all actions relevant for the capture of re-use relations.

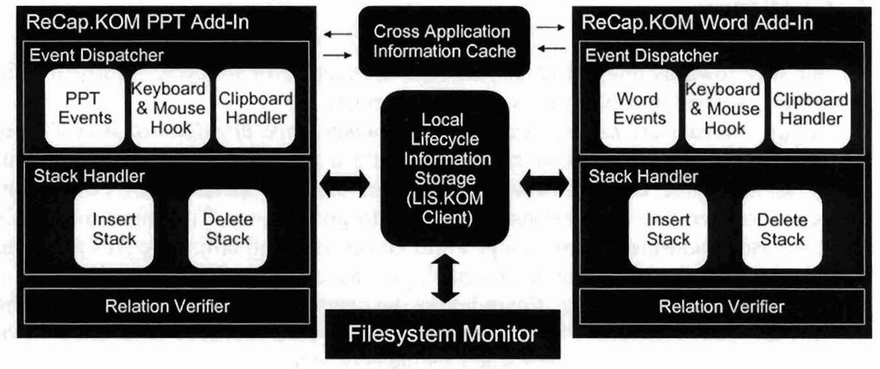

Figure 1: Simplified Component Diagram

The information needed to depict a relation includes information about source and target document (IDs, Path/URL, Title, Owner etc.), the relation type as well as which elements within the source and target document it points to. For the elements the internal ID, the type and a (high-similarity-near-distance) fingerprint is stored. To get this information we basically have three possibilities: If it is a PowerPoint internal relation, we get the information directly from the add-in. If it is an external object, we have to rely on information we get from the Clipboard. And lastly if an object or text is copy-pasted between applications which both have a ReCap.KOM add-in (like e.g. between Word and PowerPoint), the relevant information can be gotten by the add-ins but has to be transferred between the two application contexts. This is done using the Cross Application Information Cache (CAIC). Every time an object is copied, the source information of this object is stored along with its unique hash in the CAIC. When something gets pasted in a monitored application the source information for the pasted object is gotten from the CAIC by its hash. Thus a fully-fledged relation can be constructed and stored in the local lifecycle information storage (which at the same time represents the LIS.KOM client described in [3]).

To keep the actual information state in the local storage synchronous with the actual document state, delete, undo and redo actions of users have to be monitored carefully. Since the given API does not provide access to the undo and redo stacks we had to implement our own solution. The Stack Handler handles all actions relevant for the actual lifecycle information state.

In addition to the add-in itself we have implemented a file system monitor to cope with changes to relevant files in the file system outside of PowerPoint. For instance, when a PowerPoint file is renamed or moved the information kept in the local storage is updated accordingly.

Lastly we need a mechanism to verify captured relations, e.g. in case of an open scenario where a presentation is edited by a user without ReCap.KOM add-in. The relation verifier in PowerPoint iterates all relations in the local storage and verifies them in two steps when the document is opened. First the existence of involved elements is tested and secondly fingerprints of the elements are compared to judge their similarity. When the similarity is below a specific threshold, the relation is dismissed.

We have implemented the aforementioned system and conducted a first evaluation which has shown that re-use relations can be captured in PowerPoint with high reliability (see [3] for details). We currently try to further improve this.

5 Conclusion

In this paper we have proposed the capture of information in proprietary office applications as means to automatically generate information that is utilizable for search, retrieval or authoring. We identified the main challenges and described how they have been overcome. First evaluations have validated our assumption that re-use of PowerPoint presentations is actually happening and that the capture is feasible.

Next Steps include the implementation of add-ins for other applications and further improvements of the existing add-ins. A user driven evaluation will show how useful lifecycle information actually is.

References

1. Hodel, T.; Hacmac, R. & Dittrich, K.R. (2005), Using Text Editing Creation Time Meta Data for Document Management, in 'Proceedings of Conference on Advanced Information Systems Engineering'.
2. Lehmann, L.; Hildebrandt, T.; Rensing, C. & Steinmetz, R. (2007), Capturing, Management and Utilization of Lifecycle Information for Learning Resources, in 'Proceedings of EC-TEL 2007
3. Lehmann, L.; Rensing, C. & Steinmetz, R. (2008), Capture of Lifecycle Information to Support Personal Information Management, accepted at European Conference on Technology Enhanced Learning ECTEL 2008'.
4. Verbert, K.; Jovanovic, J.; Duval, E.; Gasevic, D. & Meire, M. (2006), 'Ontology-Based Learning Content Repurposing: The ALOCoM Framework', International Journal on E-Learning 5 (1), 67-74.
5. Wolpers, M.; Najjar, J.; Verbert, K. & Duval, E. (2007), 'Tracking actual usage: the attention metadata approach', International Journal Educational Technology and Society 11, 106-121.