A Framework for Usage-based Document Reengineering

Madjid Sadallah[1], Benoît Encelle[2], Azze-Eddine Maredj[1], and Yannick Prié[3]

[1] CERIST, Algeria
[2] Université de Lyon, CNRS
[3] Université de Nantes

Abstract. This ongoing work investigates usage-based document reengineering as a means to support authors in modifying their documents. Document usages (i.e. usage feedbacks) cover readers' explicit annotations and their reading traces. We first describe a conceptual framework with various levels of assistance for document reengineering: indications on reading, problem detection, reconception suggestions and automatic reconception propositions, taking our example in e-learning document management. We then present a technical framework for usage-based document reengineering and its associated models for documents, annotations and traces representation.

Keywords: Digital reading, Reading usages, Annotations, Traces, Document reengineering, Document reconception

1 Introduction

A paramount concern of document authors, be these documents paper or digital, is to best convey knowledge by sustaining document reading, understanding and appropriation. However, designing documents that are received the way the author wishes has always been difficult, partly because of the intrinsic difficulty of structuring ideas and writing, partly because the readership and its reactions are not known at the time of writing. The digital world increases this difficulty by multiplying the possibilities related to mixed medias and interactivity, hence increasing the complexity of documents with the use of multimedia content, more and more interactivity, etc. While such documents promote innovative uses, their usages are neither totally known nor easily predictable. Digital documents are also easily editable/alterable and can be updated on a regular basis, be it for their conception (e.g. a scientific article) or their reconception (e.g. a course that evolves). Moreover, appropriate authoring/reading tools can allow the establishment of a persistent, two-ways communication concerning documents between authors and readers. As a result, it becomes possible for authors to consider reader usages and feedbacks as a *knowledge source* when reconceiving their documents in order to enhance their appropriation for instance.

This article presents our ongoing work to explore some issues related to usage-based document reengineering. Being the content creators, authors are in best position to update their documents by considering readers' document usage traces (records of their interactions on the reading tool, representing the history of their actions and readers' annotations). We claim that authors should be provided with assistance during this reconception task; hence, we propose a conceptual framework to give them various levels of assistance: indications on reading, problem detection, reconception suggestions and automatic reconception propositions. This serves us then to elaborate a more technical framework that included models for documents, annotations and traces representation.

The remainder of this paper starts by a short review of relevant background. Section 3 introduces some key concepts and briefly describes our general conceptual framework. Section 4 outlines the technical framework proposal. We finally conclude and highlight some future work.

2 Related work

Based upon *Adaptive systems* like AH (Adaptive Hypermedia) [2], many tools tailor contents to individual users [11]. Some of these tools have been implemented to monitor users' activities and to analyze their interactions and annotations. For instance, comments (i.e. explicit document annotations) left by readers on Web documents can be analyzed and used to help summarizing these documents [6]. Other tools use trace-based analysis. For instance, many *Technology-Enhanced Learning* systems use traces for tracking learners' activity in order to personalize the learning experience and environment [4, 10, 7, 5]. Another kind of trace-based tools concerns digital reading systems supported by content providers like *Google Books*[4] and *Amazon Kindle*[5]. These tools have the ability to collect and retain very detailed information about readers, their usages and their habits.

To reduce the amount of data resulted from interaction tracing, storage and processing, and in order to enable aggregating this data into more human and machine meaningful units, many authors use semantic modeling of activity traces. The theory of *Modeled Trace* (noted *M-Trace*) permits to establish such a modeling [10]. According to this theory, a trace is a set of timestamped observed elements (i.e. *obsels*) representing the interaction between the user and the system. A trace model defines constraints on contained obsels (i.e their structures, types and possible inter-relations). A modeled trace (M-Trace) is a trace together with its trace model. A *Trace-Based Management System* (*TBMS*) is used to store and manage M-Traces. Within a TBMS, two types of M-Traces can exist: *primary traces* (i.e. *initial traces*) and *transformed traces* (i.e. *higher level traces*). A *primary trace* is collected from external sources and stored as an M-Trace while M-Traces created after performing transformation operations on existing traces are called *transformed traces*.

[4] `http://books.google.com/`
[5] `https://kindle.amazon.com/`

3 Usage-based document reengineering

3.1 Document reengineering

Document engineering is concerned with principles, tools and processes that improve our ability to create, manage, and maintain documents in any forms an in all media. Modern digital documents are nowadays no longer single-version with static content, they are "*live*": multimedia, multi-user, dynamic and thus multi-version [1]. Consequently, these documents are never in a final state of absolute stability: *reengineering* can be applied on them. According to [3], "*reengineering, also known as both renovation and reclamation, is the examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form*". Regarding documents and from our point of view, the main goal of what we call "document reengineering" is to improve document structures and document content in order to facilitate their appropriations by readers. These documents are usually described along different structures which drive different possible representations. Being multimedia, these documents combine objects of different nature like text, sound, image and video; hence, they are usually modeled following four dimensions not totally independent [9]: 1/ logical (document organization into for instance chapters, shots, etc.), 2/ spatial (graphic layout), 3/ temporal (temporal ordering of the multimedia objects) and 4/ hypermedia or links (relations between documents and document fragments).

3.2 Using usages for reengineering documents

In the digital publishing and reading context, we define *usage-based document reconception* as a kind of reengineering that alters document content and structures in response to readers' explicit feedback (i.e. annotations), or implicit ones (i.e. reading traces). In this paper, we define a trace as *a temporal sequence of observed elements recorded from interactions between a reader and a document, through a reading tool*. Considering usage feedbacks for reengineering purposes assumes that the reading tool has firstly the ability to monitor readers, intercepting and eventually interpreting their interactions. The relevance of such feedbacks in regards to document reengineering will greatly depend with readers' involvement in content appropriation. Active reading can foster readers' involvement, as stated in [8]: document *active reading* provides a chance to best engage readers towards contents and enhance appropriation and understanding. Hence, the reading tool has secondly to provide active reading functionalities in order to improve the potential usefulness of such usage feedbacks.

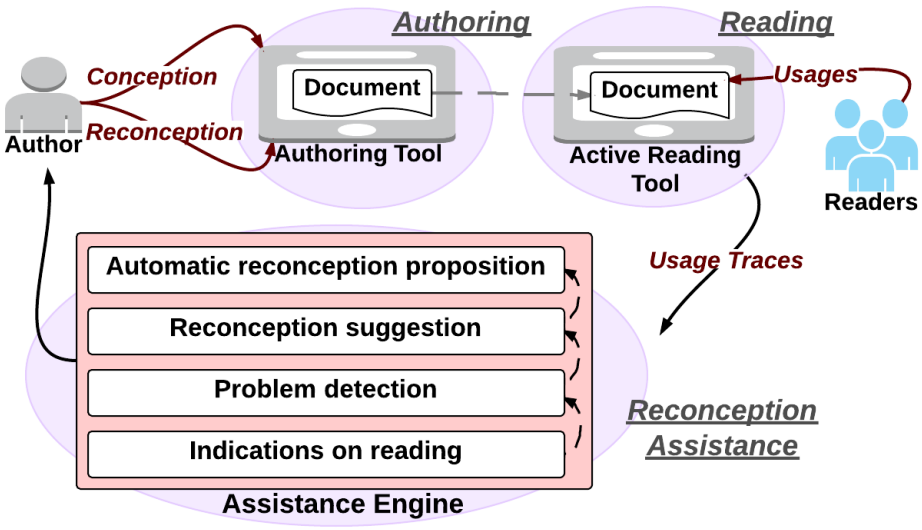

Fig. 1. Overview of the reconception model

3.3 Conceptual model for usage-based document reengineering

While our approach can be applied to any document, we instantiate it within the technical context of the Learning Content Management System **Claire**[6]. Claire aims to offer a simple, yet robust tool for authoring, improving and disseminating educational content. Our usage-based document reengineering proposal is based on the general model of figure 1. Instrumenting an active reading tool, data about usages (i.e. reader interactions: reading trace and annotations) is analyzed by an *assistance engine* which may then assess possible and appropriate document reconceptions. We identify four main levels of author assistance, each level exploiting data from the previous one. All of these levels are illustrated in the following using Claire use cases, where an author presents his course to a group of students and performs course assisted-reconception.

- ***Level 0:*** **Indications on reading.** The assistance engine can compute and present the author with indications on how the document has been read. *Example*: giving the author the percent of readers that have followed a given link may help him to understand the relevance of that link.
- ***Level 1:*** **Problem detection.** Based on the previous level, the assistance engine may detect problems in the reading process but not give any suggestion on how to fix them. *Example*: if a video component has never

[6] *Community Learning through Adaptive and Interactive multichannel Resources for Education.*
`http://www.projet-claire.fr/`

been watched longer than its first seconds, the engine reports to the author this fact as an unexpected behavior.

– ***Level 2:*** **Reconception suggestion.** At this level, not only the system detects problems but in addition, it may supply suggestion. However, the system is unable by itself to achieve these suggestions. *Example*: if many readers of a course document usually go back to a previous chapter, the engine may suggest ways of getting rid off them, for instance to include a recall of the main concepts already seen in a previous lesson unit.
– ***Level 3:*** **Automatic reconception proposition.** At this level, the engine may detect problems and resolve them automatically. Consequently, a reconception can be presented to the author for review and validation. *Example*: if many zooms are performed on a part of the document, the system can automatically readjust and increase its size or fonts for a text.

4 Technical framework

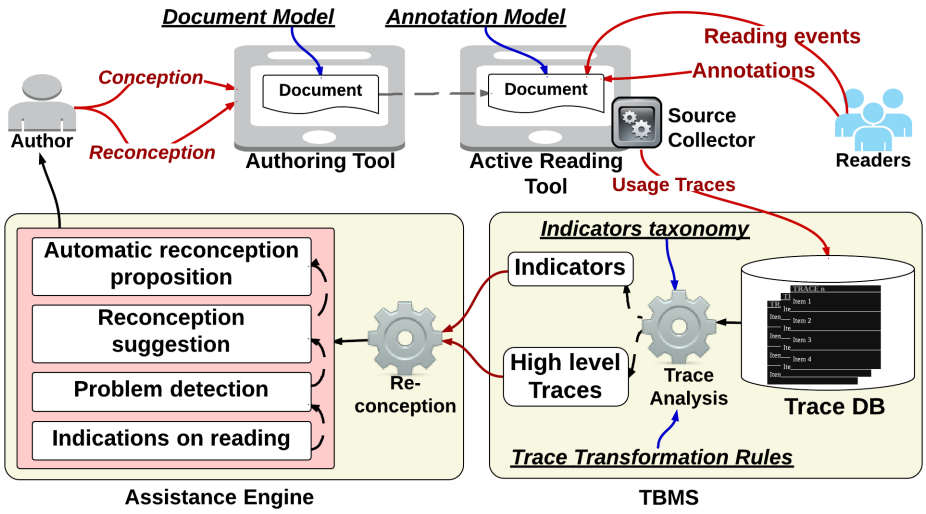

Fig. 2. Overview of the technical framework

Based on the general usage-based document reengineering conceptual model, an illustration of our framework proposal is presented on figure 2. Once a new document is conceived by an author, it is published. The active reading of the document produces a set of obsels (*primary traces*) collected by a *Source Collector* installed on the reading tool. This data is then sent into the *Trace-Based Management System TBMS* to be stored and processed. Once indicators and high level traces are computed, these can be used by the assistance engine to perform reconception and to ensure different levels of assistance. As shown on the

figure, a set of data models are introduced to describe many related features. In the following, we introduce some of these for documents, annotations and traces description.

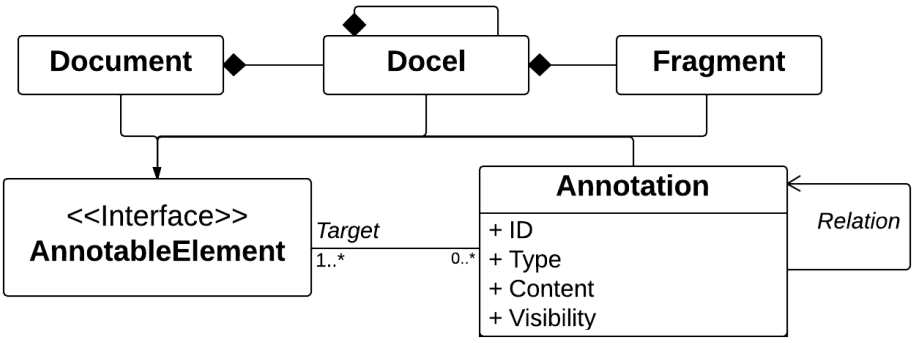

Fig. 3. Document and annotation generic model

4.1 Documents

Document reconception may target and affect both document content and its structures (spatial adjustment, temporal synchronization, etc.). This motivates document modeling in order to well describe the document features and structures that may be directly or indirectly involved in an interaction or a reconception. Figure 3 introduces a generic logical model to describe digital documents. A *document* is considered as the nesting and composition, at different levels of granularity, of *docels* (document elements) which are the building blocks to represent formal elements and composition units. Each element is associated with a list of attributes that describe its composition, placement, synchronization and behavior. A *fragment* is a logical part of a document element. It can be defined using spatio-temporal coordinates.

We have instantiated this generic model for representing *Claire* data structures. Three levels were defined. The lower one is called *assets*, typically describing a title, paragraph, graphics, etc. A *granule* is composed of a set of assets and generally represents a course chapter. A pedagogical *module* — a *document*— is a coherent assembly of granules that typically forms a course.

4.2 Annotations

We define a document annotation as any information provided by a user that is associated with a whole document or a part of it. Since explicit annotation structuring and typifying allows automatic processing and analysis, we propose a generic model contained within figure 3. An annotation target can be one

or more document elements and/or fragments and/or annotations, all of these referred as "annotable element". Each annotation has one and only one type. The available annotation types depend on the nature of the "annotable element". Among the types we have defined within the Claire project are: *Question*, *Form error* (formal mistake like spelling, grammar, etc.), *Content error* (mistake in the content, for instance in a source code), *Comment*, *I understood* (to point out that an item has been useful for understanding), *I did not understand* (to report a lack of understanding), *Lecture notes* (personal notes) and *Other* (a custom annotation). We also have *Highlighting* to emphasize some document parts (such annotations do not have content) and *Linked annotation* to associate an annotation to another one (to annotate an annotation). Another aspect of annotations is their visibility to control their availability to different types of users. In Claire project, an annotation can be *private* (only available to the author of the annotation), *to author/reviewer* (only available to the annotator and to the author and reviewers), *group* (only available to a specific group of users) or *public*.

4.3 Initial traces

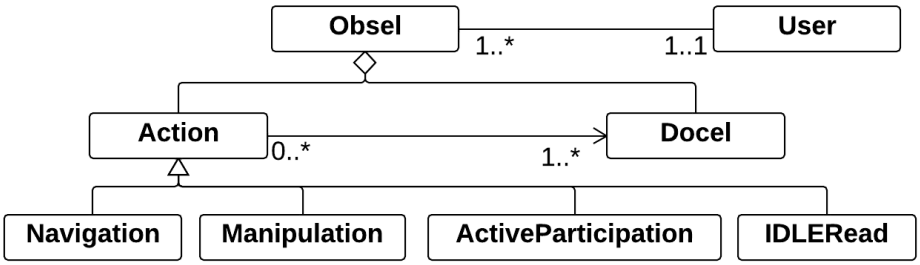

Fig. 4. Trace model

We consider traces as *modeled Traces* [10], their model is presented on figure 4. Each observed element (*obsel*) is associated to a user and connects a specific action with a document element *docel*. We have identified some generic actions (*obsel types*) that are commonly used by digital readers and divided them into four main classes:

- **Navigation**. This class covers common navigation actions like *following links*, *visiting specific URLS*, *scrolling* (spatially and/or shifting in time) and moving *back* and *forward* in navigation history.
- **Manipulation**. This class refers to readers manipulation actions on the document content (e.g. *select*, *find*, *print*, *zoom*, *copy* and *bookmark*) and context (e.g. activating system interface to *open/close/download* the document). Particular media related actions cover the very common ones (*play/pause/stop*, *seek*, etc.).

– **Active participation**. User explicit participation is mainly expressed in terms of annotation actions. These actions include: *adding/altering/deleting* one's annotations, *annotating/opening/closing* an annotation, *highlighting*, etc.
– **Idle Read**. This class describes a reading that mostly appears passive (without significant interaction). It can also characterize the user inactivity (or absence).

4.4 High-level traces and indicators

The assistance engine is responsible of traces analysis and interpretation. Two kinds of results can be produced: a/ ***high level traces*** *generation:* a *transformation process* performs transformations on the primary trace to interpret and abstract it. Such transformations include filtering, rewriting and aggregating obsels; b/ various *reading* ***indicators*** *computation:* these are variables computed to characterize readers' interaction against a specific monitored feature or event (e.g.: unread sections, visited/unvisited links, interaction level, spent time on specific parts). To this end, a meaningful taxonomy of these indicators has first to be established. Using these two kinds of analysis results, authors can be assisted during the reengineering tasks following the four levels already presented in the conceptual framework. The author can choose to consider an arbitrary set of feedbacks originated from a single reader, a given group of readers or the entire readership. The end result is a new version of the document which can in turn be subject to further revisions.

5 Conclusion and future work

The ongoing work presented in this paper focuses on some issues related to usage-based document reengineering. Ideas presented are twofold: how to reconceive documents by exploiting readers' feedbacks and how to assist authors to achieve such reconceptions. As a result, a conceptual framework for document reengineering is presented that uses readers' usage feedbacks (reading traces and annotations) and offers authors various levels of assistance. A technical framework and associated data models are then developed according to this conceptual framework.

Future work will focus on the conception of suitable means and tools to assess reconceptions, using the primary traces and going through the suitable trace transformations and indicators computation. Proper reconception being tightly related to the technical context, we rely on Claire project to conduct interviews with some course authors in order to identify the actual reconception needs and therefore to precise/specialize the different associated models. This will serve us then to elaborate a meaningful set of transformations and indicators for enhancing documents that are provided to learners. The ongoing implementation of the technical framework and its future integration within Claire will be our first proof of concept. Thereafter, we can consider expanding our proposals to other application areas.

References

1. H. Balinsky and S. J. Simske. Secure document engineering. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 269–272. ACM, 2011.
2. P. Brusilovsky. Adaptive hypermedia. *User modeling and user-adapted interaction*, 11(1-2):87–110, 2001.
3. E. J. Chikofsky, J. H. Cross, et al. Reverse engineering and design recovery: A taxonomy. *Software, IEEE*, 7(1):13–17, 1990.
4. C. Choquet and A. Corbière. Reengineering framework for systems in education. *Journal Of Educational Technology and Society*, 9(4):228, 2006.
5. S. D'mello and A. Graesser. Autotutor and affective autotutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems*, 2(4):23:1–23:39, 2012.
6. M. Hu, A. Sun, and E.-P. Lim. Comments-oriented document summarization: understanding documents with readers' feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 291–298. ACM, 2008.
7. J.-C. Marty, T. Carron, and P. Pernelle. Observe and react: interactive indicators for monitoring pedagogical sessions. *International Journal of Learning Technology*, 7(3):277–296, 2012.
8. M. McLaughlin. Reading comprehension: What every teacher needs to know. *The Reading Teacher*, 65(7):432–440, 2012.
9. C. Roisin. Authoring structured multimedia documents. In *Proceedings of the 25th Conference on Current Trends in Theory and Practice of Informatics: Theory and Practice of Informatics*, SOFSEM '98, pages 222–239, London, UK, UK, 1998. Springer-Verlag.
10. L. S. Settouti, Y. Prié, J.-C. Marty, and A. Mille. A trace-based system for technology-enhanced learning systems personalisation. In *Proceedings of the 2009 Ninth IEEE International Conference on Advanced Learning Technologies*, ICALT'09, pages 93–97. IEEE Computer Society, 2009.
11. D. Smits and P. De Bra. Gale: a highly extensible adaptive hypermedia engine. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, HT'11, pages 63–72, New York, NY, USA, 2011. ACM.