Proceedings of the 3rd WSEAS/IASME International Conference on ENGINEERING EDUCATION, Vouliagmeni, Greece, July 11-13, 2006 (pp227-232)

MemoNote, a semantic and personal memory tool based on annotations made on pedagogical documents

CYRILLE DESMOULINS
FAIÇAL AZOUAOU
CLIPS-IMAG
University Joseph Fourier
BP53, 38041 Grenoble cedex FRANCE

Abstract: This paper describes the specification of the MemoNote tool. This tool provides learners and teachers with a memory of their interaction with documents. It is based on annotations they add directly on educational documents, to be remembered later. Based on a description of the tool requirements, a precise specification of internal and external behaviour and component is described. This specification has been used to develop two different versions, one based on TabletPC for mobile use and the other based on a LCMS for use through the WEB, with the possibility to synchronise data between them. The current state of implementation for each of these two versions is described and future directions of research and development are depicted.

Keywords: personal memory, annotation, active reading, e-learning, personal information system.

1 Introduction

During learning activities, the use of documents by both teachers and learners is not only passive. It involves active reading [1], where documents readers not only read and understand the document content but also interact with this document during its reading to construct their own knowledge, by adding their own elements (questions, remarks, interrogations, complements, synthesis, …).

Active reading has usually two purposes. On a first hand, it helps the reader to keep his/her attention on the current task. On a second hand, it is a mean for him to build and manage a personal memory of his/her interaction with the document, to be reused in the future.

The personal elements added by the reader take two main forms. Direct annotations augment the document whereas external notes are taken separately, in another space that the document's one.

Whereas e-learning is progressively increasing and consequently the use of digital documents within learning activities, active reading is rarely supported within the computer. Usually it is simply because pedagogical documents are read-only (for example on the WEB) and if not, it is because some document parts have been previously designed to be filled in. In the best case, active reading is proposed in LCMS as notes to attach to the whole document (as in Learning Space [11], WebCT [14] and Phoenix [9]] or as a forum referring to parts of the document [2].

This article aims at describing MemoNote, a tool supporting active reading in order to provide the various actors of e-learning, mainly teachers and learners, with a memory of their interaction with documents to be reused later. This tool is based on annotations directly placed on educational documents during their use. The article is organised as follows. The first section present the general requirements for such a memory tool based on annotation and dedicated to education. The second section details the tool specification, in terms of external and internal behaviours and representations. The third section describes the current state of the MemoNote implementation on two platforms, Tablet PC for mobile use and LMCS-based to be used through the web.

2 MemoNote requirements

MemoNote aims to provide a memory to teachers and learners interacting with pedagogical documents. We define in this section what it means precisely through a series of requirements.

A long term memory

A basic and generally accepted classification of human memory in psychology is based on the duration of memory retention. It identifies three types of memory: sensory memory (from milliseconds to seconds), short-term memory (from seconds to minutes) and long-term memory (from days to years). MemoNote addresses long-term memory and is a sort of prospective memory [15]. As a long memory, it should provide three main functions: coding, storing and recall elements in the memory.

An *external* memory

MemoNote memory is intended to be the external complement to human personal memory, supporting and enhancing it, in the sense of Chalmers [7]. It does not aim at simulating people's memory like in [8]. Such an external memory contains both annotations and educational documents they are referring to.

A digital memory

MemoNote memory is a digital one. This requires that digital document can be used in every educational settings. For example, outdoors or lab activities require the use of mobile devices like PDA or Tablet PC. This digital memory contains both digital annotation and digital educational documents.

A use memory

MemoNote aims at memorising annotation made during educational activities. More precisely, it focuses on "active learning situations", like exercises, labs, experiments where learning is organised through tasks to be performed by the learner. Memorising and remembering are situated, relating to a precise learning situation.

A semantic memory

MemoNote is a semantic memory. It means that MemoNote annotations are not full text but refer to concepts which are semantically defined and structured, and agreed in the educational community. Then, remembering can refer to semantic concepts organised into ontologies. This semantic network of concept (and their instances) is part of the memory in addition to documents and annotations.

A cognitive memory

Following the distinction of Caussannel & al. [5], MemoNote is obviously a computational memory, because it provides automatic ways of memorising and retrieving annotation. But unlike Web semantic approaches, which are uniquely computational, MemoNote is also cognitive. This means that annotations are visible for human user and that human agents interpret this visible form.

A personal memory

Even if it is semantic, MemoNote does not deals with objective/universal annotations as in the Semantic WEB approach [4]. Only personal annotations and views on pedagogical documents are memorised. In psychology, this type of memory is usually called an "episodic memory" opposed to "semantic memory". MemoNote is both semantic and episodic, referring subjectively to objective and shared concepts.

An e-learning memory

Last but not least, MemoNote is an e-learning memory. It intends to support learner to learn or teacher to teach by extending their personal memory. It means that MemoNote is able to change its behaviour upon domains/topics and levels changes, and, mainly on the teacher's side, incorporates learning concepts (pedagogical objectives, strategies …). Concretely speaking, this requires that MemoNote semantic network of concepts and instances can be changed for each domain/level.

3 MemoNote specifications

From these requirements, we specify in this section MemoNote external and internal specifications.

3.1 External specifications

We present MemoNote external specifications first by identifying MemoNote main use cases, then by specifying the adaptation to context of these use cases and finally by presenting MemoNote Human -Computer Interaction principles.

MemoNote use cases

MemoNote uses cases are divided into two main categories: routine uses and content organisation.

Memory routine uses

- Memorising.

Memorising is performed by creating an annotation in manual or semi-automatic way.

- o *Manual annotating*

User selects an annotation form, an anchor on the document and then defines the annotation semantics (annotation objective and content, addressee...) referring to semantic concepts. The annotation episodic facet is automatically provided by the tool.

- o *Semi-automatic annotating*

User annotates using annotation patterns. In this case, the annotation semantics is deduced entirely or partially by the tool depending on the chosen pattern.

- Remembering
- o *On demand remembering*

The user asks explicitly MemoNote to retrieve past annotations. He/she provides annotation elements he/she remembers (for example, the date or the topic) and MemoNote returns the corresponding annotations. The user can then sort these annotations, focus on a detail (for example, to display it in the document context).

- o *Pro-active remembering.*

In this case, annotation itself can trigger an action under certain conditions. If these conditions are realised, MemoNote pro-actively reminds the user this annotation, for example by emphasizing it on the document or by sending an alert to the user.

Content organisation

Apart from memorising a new annotation, a MemoNote user can manage the memory content, (documents, ontologies, annotations) and the patterns it uses.

The following use cases describe the memory content management:

- *Adding documents*
- *Deleting documents*
- *Adding ontologies*
- *Deleting ontologies*
- *Receiving annotation given by another user.*
- *Sending annotation to another user.*
- *Synchronising memory.*

As annotation can be created using different MemoNote instances (on different computers) or versions (MemoNote TabletPC, MemoNote LMCS,…) by the same user, this creates different memories on each device/platform.

The user can then ask MemoNote to synchronise these memories content. He/she can synchronise only annotations, or ask also for ontologies or document synchronisation.

Pattern management concerns the following use cases:

o *Creating an annotation pattern*

The user defines an annotation pattern by specifying its visual form and semantics, and by defining the context is which it is valid.

o *Editing an annotation pattern*

o *Deleting an annotation pattern*

Context-awareness

As MemoNote is a use memory which means that the learner/teacher can memorise and remember in different educational situations. MemoNote should then adapt its behaviour according to each particular situation (domain, level…) and incorporate context-awareness into the precedent uses cases.

According to Chen & Kotz [6] *there are essentially two ways to use context: automatically adapt the behaviours according to discovered context (active context-awareness), present the context to the user on the fly, or store the context for the user to retrieve later (passive context-awareness).*

These two types of context-awareness are introduced in the following MemoNote use cases.

- Memorisation

o *Manual annotation*

MemoNote records the current context in the annotation object (passive awareness).

MemoNote changes dynamically the set of annotation ontologies depending on the current (active awareness).

o *Semi-automatic annotation*

MemoNote automatically selects and displays annotation patterns that are suitable to the current context (active awareness).

- Remembering

o *On demand remembering*

MemoNote displays each annotation within its creation context (passive awareness).

MemoNote adapts the result and the display of annotation queries depending on the current context (active awareness).

o *Pro-Active remembering*

Pro-active remembering is fundamentally an active context-aware use case because actions are triggered upon current context change.

- Content organisation

o *Synchronising memory*

MemoNote triggers the annotations synchronising between the different databases when the context state permits it, for example when network resources are working (active awareness).

o *Creating an annotation pattern*

MemoNote automatically records the current context into the pattern at its creation (passive awareness).

MemoNote Human-Computer Interface

The rationale on which MemoNote HCI relies is twofold.

At first hand, to provide a cognitive memory, MemoNote HCI is based on the paper-based annotation metaphor. This means that the user manipulates annotation tools like in the paper environment. He can take out annotation tools from a tool case to display them in the MemoNote working place and put them back. When a tool is selected, each text selection creates immediately an annotation with the corresponding visible form. A rubber enables to delete annotations by its form on the document. Every action on the HCI does not require confirmation (as in paper settings) and is immediately recorded.

On a second hand, to provide a computational memory, MemoNote HCI benefits from computations made by the computer. User memorisation is then enhanced by automatically recording several annotation properties, providing easy and fast way of annotation. User remembrance is enhanced with multi-criteria filters and sorting based on ontologies, focus on annotation attributes (particularly visualising annotation in the document in which it was created).

3.2 Internal specifications

The MemoNote internal specification concerns first the persistent representation of the annotation, the pattern and the context. It concerns also the synchronisation process and the architecture.

Persistent representation of the memory elements

Annotation

MemoNote annotation model is semantic, personal and cognitive. These three aspects are represented in terms of an object with three main facets where each facet describes one of the three aspects (see the next Figure).

The semantic facet attributes take their values from a formal representation of knowledge, based on ontologies.

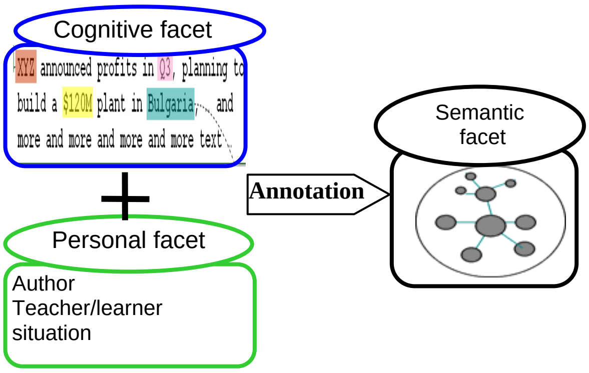
Cognitive facet
XYZ announced profits in Q3, planning to build a $120M plant in Bulgaria, and more and more and more and more text
+
Personal facet
Author
Teacher/learner
situation
Annotation
Semantic facet

Fig. 1. Three facets model

Then each facet is divided into sub-facets or attributes, to describe the annotator's annotation in terms of personal, cognitive and semantic attributes. In addition, as the annotation is personal, it is important in the case of annotation transmission from a sender to other addressees to keep the sender's view (self-confidence + importance). To describe this personal view, the annotation model includes then a transmission facet. (See the generic model in details in table 1).

Next, annotation model is adapted to the teacher and leaner's particularities. For example, in the semantic facet, the teacher's annotation objective is detailed into three sub-objectives (pedagogical, teaching domain and document) where the learner's annotation objective is detailed into only two sub-objectives (learning domain and document).

Table 1. Annotation model

Cognitive facet	Physical anchor Syntactic anchor Visual form
Personal facet	Author Educational situation -Date -Place -educational domain -educational activity
Knowledge facet	Annotation objective Content -Structured content -Free content Force -Importance -Confidence Recall situation -Addressee -Educational activity Effect
Transmission facet	Sender Possessor Rights Force -Importance -Confidence

Pattern representation

The pattern conceptual model includes the following elements:

1. *Name*. The name should express the semantics produced by the annotation related to this pattern.
2. *Context*. It represents the context state where the current pattern can be used.
3. *Problem* to be solved. It represents the annotation semantics aimed by the annotator.
4. *Solution*. A set of values for the annotation attributes, depending on the current context and the annotation form.
5. *Forces*. It is about a description of the arguments justifying the use of this particular pattern.
6. *Related patterns*. It is about the list of patterns that have relations with the current pattern.

Context representation

To implement the contextual functions, MemoNote represents all the relevant context data using the following context ontology:

- *Context*
- *Computing*
    - *Computing Software*
        - *Annotation tool*
        - *Host system*
    - *Computing Hardware*
- *Teacher*
- *Learning Activity*
    - *Learning Domain*
    - *Learning Degree*
- *Teacher Activity*
- *Place*
- *Time*.

The particularity of this context model is that it enables to represent the educational concepts which are relevant the teacher/learner's annotation.

Annotation synchronisation

To synchronise the content of the different annotation databases belonging to the same annotator on different devices, MemoNote keeps marks of annotations made since the last synchronisation with some device or other. This synchronising process uses specific protocol to identify the differences between the annotator's databases in order to update them.

Architecture

The architecture of the context-aware application is divided into six autonomous components.

Context captors. They are the logical components, which are abstraction of the physical sensors. A given context captor is in charge of acquiring data on particular MemoNote's context elements.

Capture manager. It is in charge of collecting context data, aggregating them into a unique format and providing the latter to the other components.

Context manager. It is in charge of context supervision: collecting context data and processing them into context, storing con text states and p*roviding both current context and state log.*

Pattern manager. It is in charge of patterns management: creation, update, and patterns selection for a given context …

Memory manager. It is in charge of managing the memory components: documents, annotation ontologies, and annotations.

Interface manager. It is in charge of building MemoNote's interface element (palettes, menus…).

4 Implementation

MemoNote is implemented in two versions. The first one is implemented on TabletPC platform and is based on Mobipocket-Reader software [13]. The second one is a web-based version, in order to be integrated in a LMCS (currently the University of Savoie LMCS, based on Zope/Python). This version has a client side, implemented mainly in JavaScript and the server side is the annotation server implanted in Python.

The states of development of these two versions are different. We detail in this section for each internal or external specification what is implemented and how it is implemented

Users. At present, both versions implement only the learner's annotation model, but they include some components particular to the teacher (context + pattern in the TabletPC version and the domain ontologies in both the two versions).

Patterns. For the moment the pattern model, leading to assisted and easier annotation process, is only implemented on TabletPC.

Context-awareness. Even if the two versions implement and exploit the context, its data is mainly (except the date) provided by the annotator, who has to fill a context form. MemoNote-LMCS uses the host LMCS as source for the context data (mainly user data for the moment).

Synchronisation. We are currently working to implement the synchronisation facility into MemoNote-TabletPC. It is naturally included in MemoNote-LMCS, which relies on a central annotation server.

HCI. MemoNote Human Computer Interface is different between the two versions. Relying on another piece of software and its API (provided as DLLs), some MemoNote specifications cannot be achieved in the tabletPC version which provides a fixed number of highlighting tools (five colours) and does not provide annotation tools management (tool case). The HCI specification is fully implemented in the client side of the LMCS-version. Both versions respect the rest of the annotation metaphor (implicit record, no confirmation, …).

Data format. The two versions use different document formats. MemoNote-TabletPC must be converted (with an appropriate Mobipocket software (MobiPocket-Publisher) into a .prc file respecting the OEB standard (Open e-book format [10]). MemoNote-LMCS documents are html ones as it is a web-based application. Ontologies are represented in the standard OWL in the MemoNote-LMCS whereas they are simply represented following an XML DTD in MemoNote-TabletPC. In both versions annotations are represented in XML, the objective being to switch to the RDF standard, which is currently used to represent the patterns in the TabletPC version.

Finally, to describe the domain of each attribute, in particular the semantic ones, we developed several ontologies aimed to either the teacher activity (like as teacher pedagogical objective) or the learner activity (like learner structured content), in addition to generic ontologies (like document objective). These ontologies can be used with the two MemoNote versions, after being represented in the corresponding format. The learner ontology has been experimented with MemoNote-TabletPC [12] and the teacher one was developed with the help of chemistry teachers using annotations they have made on paper [3].

5 Conclusion and future work

In this paper, we presented MemoNote, a personal memory tool, based on annotations and dedicated to both teacher and learner. This memory must be at once external, digital, semantic, cognitive and personal. It must also be a use memory dedicated to educational activities.

MemoNote provides memory functionalities of memorising (by annotating) and remembering on one hand, and content management functionalities (patterns management, synchronising…) in the other hand.

MemoNote internal models (annotation, pattern and context) reflect the requirements (semantic, cognitive, e-learning…) the resulted memory should respect.

We developed two versions of MemoNote. The first one is based on TabletPC for mobile use, the second one is integrated into an existent university LCMS. The data formats for the different internal models use XML-based languages in order to facilitate annotation processing and exchange.

Future work will focus first on extending the integration between the MemoNote-LCMS and the LCMS system so that the whole context data will be recovered automatically and directly from that LCMS. The patterns, which are at present implemented only in the MemoNote-TabletPC, should be generalised to the LMCS version.

Another issue is to improve the annotator‘s interaction in the TabletPC version in order to make it as effortless as the LCMS version which reproduce the paper annotation process. This will require to either rely on other DLLs for the integration with Mobipocket Reader, or to choose another reader.

As soon as synchronisation will be ready, we plan to start experimenting the concurrent use of the two versions with users accessing to the LCMS platform and needing to use mobile device.

The major challenge are long term experiments (at least a semester or a year), to evaluate the usefulness of the memory. This will require the implementation of on demand remembering which is completely specified (from the data to the interface) but not yes implemented.

Finally all the data formats has to be migrated to the standards formats of RDF and OWL (it is only partially the case currently).

References:

[1]Adler, M.J. and C.v. Doren, *How to read a book*. New York: Simon and Schuster. 426. 1972.

[2]Azouaou, F., W. Chen and C. Desmoulins. Semantic annotation tools for learning material: Specification and categorization. *CAiSE'05 (The 17th Conference on Advanced Information Systems Engineering)*. Porto, Portugal: FEUP edicoes. pp. 645-660. 2005.

[3]Azouaou, F., Desmoulins, C, Teachers' document annotating: Models for a digital memory tool. *Int. J. Cont. Engineering Education and Lifelong Learning (IJCEELL)*. Vol. 16(1/2) p. 18-34. 2006.

[4]Berners-Lee, T., J. Hendler and O. Lassila, The semantic web. *Scientific American*. Vol. 5: p. 34-43. 2001.

[5]Caussanel, J., J.-P. Cahier, M. Zacklad and J. Charlet. Cognitive interactions in the semantic web. *Semantic Web Workshop at WWW-2002*. Hawai. pp. 1-10. 2002.

[6]Chen, G. and D. Kotz, A survey of context-aware mobile computing research. Technical Report: TR2000-381 Dartmouth College. 2000 Hanover, NH, USA. p. 16.

[7]Clark, A. and D. Chalmers, The extended mind. *Analysis*. Vol. 58(1) p. 7-19. 1998.

[8]Damas, L., A. Mille and R. Versace. Authoring interface and learner models for long-term skill acquisition. *AIED*. Sydney, Australia. 2003.

[9]Fernandes, E., H. Madhour, S. Miniaoui and M. Wentland. Phoenix: An xml-based tool for e-learning environment. *EDMEDIA*. 2005.

[10]Forum, I.D.P., Open ebook publication structure specification version 1.2. 2006.

[11]IBM, *Using ibm lotus learningspace – virtual classroom*. Ibm redbooks. 280. 2002.

[12]Mille, D., Modèles et outils logiciels pour l'annotation sémantique de documents pédagogiques. Thesis. 2005, Joseph Fourier University, Grenoble.

[13]MobiPocket, Mobipocket reader software. 2004 Paris. www.Mobipocket.com.

[14]WebCT, Http://www.Webct.Com/. 2006.

[15]Winograd, E., Some observations on prospective remembering. *In M. M. Gruneberg, P. E. Morris & R. N. Sykes (Eds.), Practical Aspects of Memory: Current Research and Issues*. Vol. 2: p. 348-353. 1988.