A Paper-based Technology for Personal Knowledge Management

Kinga Schumacher, Marcus Liwicki, and Andreas Dengel
German Research Center for AI (DFKI GmbH)
Knowledge Management Department, Kaiserslautern, Germany
{Firstname.Lastname}@dfki.de

Abstract: In this paper we extend the Semantic Desktop with a novel input modality: handwriting on paper. This extension allows for annotating printed documents with handwritten notes, which are then recognized, interpreted and integrated into the personal knowledge space. Thus it supports personal knowledge work on paper. In particular, the user can mark any text segment and annotate it with his or her own comments. The semantic of these comments and marks are then interpreted and, finally, the result is included in the user's Semantic Desktop. The methodology for finding the intended meaning of the annotation in context of the marked text is presented in this paper. To demonstrate the applicability of our approach, we have implemented a prototype for the NEPOMUK Semantic Desktop.

1 Introduction

Caused by the vast amount of information and insufficient support for its organization, people are faced with the problem of information overflow. The increasing need for supporting tools has motivated growing interest in Personal Information Management (PIM) research during the last years. Models and technologies from Semantic Web research have been adopted to desktop computers in order to overcome the limits of conventional file systems and current applications borders. The extension of desktop computers with a semantic layer [SGRB08, VSO08] supports the user to organize and retrieve data according to his or her own mental model. Several PIM applications for the work with personal computers have been developed recently [QHK03, TMPP05, CPG05, GHM$^+$07, RVH05]. However, these applications do not support the knowledge work with pen on paper yet.

Using pen and paper in our everyday work is motivated by several issues. First, the user is not bound to a specific device or to a specific medium. Second, paper is portable, making it possible to take notes anytime and anywhere. Furthermore, among other issues, writing on paper is more natural to most persons making the pen their preferred writing instrument for tasks such as brainstorming, collaborative work or reviewing documents [SH01, VSO08]. However, while using a pen on paper makes it easier and faster for the user to attach his or her thoughts to the document, the interpretation of the notes is often more complex compared to the situation of digital comments.

In order to overcome the problems mentioned above and to combine the advantages of

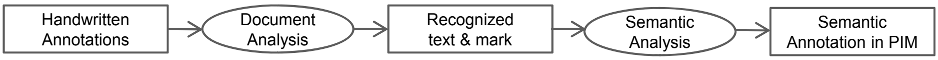

Figure 1: System overview

paper-based work with those of digital comments, we propose the Semantic eInk system (see Fig. 1 for an overview of the system). This system automatically processes handwritten annotations on printed documents. Then it interprets the semantic information of the recognized text in a Semantic Analysis step. This information will be expressed through formal semantics using the individual's vocabulary, and finally integrated into the personal knowledge base, the Semantic Desktop. The integration makes this knowledge searchable, reusable, and sharable and gives a context for its interpretation. Semantic eInk extends the Semantic Desktop with a new input modality, interactive paper. Thus it supports personal knowledge work on paper.

The ideas of the Semantic eInk system have been initially proposed in [LSD+08]. However, this paper goes clearly beyond the scope of [LSD+08]. While [LSD+08] focuses on the document analysis part of our approach, this paper describes the knowledge management methods in detail. Furthermore, we describe a prototype being able to handle any kind of text documents.

The rest of this paper is organized as follows. In the next section, an overview of Semantic Desktops and respecting tools supporting personal information management is given. Complementing, recent progress in digital paper research and methods applied to obtain digital information from the handwritten notes are summarized in Section 3. Next, Section 4 presents the methodology used for analyzing the recognized annotations and finding the meanings of them. A first prototype implementation of these ideas is described in Section 5. Finally, Section 6 draws some conclusions and gives an outlook to future work.

2 Semantic Desktops

Several systems supporting personal information management already exist, e. g., Haystack at MIT [QHK03], D-Bin by SEMEDIA [TMPP05], IRIS by SRI [CPG05], Gnowsis and NEPOMUK at DFKI [GHM+07] DeepaMehta at FZI [RVH05]. All of these means apply Semantic Web technologies and standards like RDF[1] and RDF Schema[2] on desktop computers in order to enable the identification of text and multimedia documents, e-mails, and contacts by URIs, across application borders. Therefore a semantic layer is added above the file system to link and describe the existing resources. The user is able to annotate, classify, and relate these resources, expressing his or her view in a Personal Information Model (PIMO) [SvED07].

The systems mentioned above provide different functionalities and integrate different kinds of applications, mostly calender, email, and notes. A fully-featured Semantic Desktop in-

[1]http://www.w3.org/RDF/
[2]http://www.w3.org/TR/rdf-schema/

volves many data sources, indexes the text and metadata of all documents and categorizes them. Together with the collected facts about the instances, like contact information and membership of a person, a critical amount of information is integrated and available.

Beside the personalized filing of all these information, the Semantic Desktop offers effective access to this personal memory since the information is not only personalized but also machine readable and interpretable. This allows the development of machine processes and learning methods assisting the user in his or her every-day activities and the extension and maintenance of the PIMO.

Summarizing all these properties, the Semantic Desktop supports individual knowledge workers in their information management. For our prototype, we use the Social Semantic Desktop NEPOMUK[3] which extends the described personal aspects of the information management with social aspects. The desktop is expanded to a collaboration environment in order to facilitate shared knowledge work.

3 Handwritten Input

The *paperless office*, i. e., the philosophy of working with a minimal amount of paper and converting everything into digital documents, was predicted more than thirty years ago. However, there is still an increasing amount of paper used by humans in their everyday work. Using paper is motivated by several issues. First, the person is not bound to a specific device or to a specific medium. Second, paper is portable, allowing for making notes anytime and anywhere. Furthermore, it allows for fast prototyping, since paper puts almost no restrictions on the content that can be written on it (text, remarks, highlighting, sketches).

In workflows like reviewing, where the annotations have a meaning for the marked text, the problem of mapping the paper to the digital counterpart arises. A variety of approaches have been investigated to enable this kind of paper-driven digital services. They use cameras, Wacom Graphics Tablets[4], ultrasonic positioning, RFID antennas, bar code readers, or Anoto's Digital Pen and Paper technology[5].

The Anoto technology is particularly interesting because it is based on regular paper and the recording of the interactions is precise and reliable. Therefore we use the Anoto technology in this paper. The documents are Anoto-enabled at print time by overlaying a special Anoto dot pattern with them. The Anoto pattern represents an absolute positioning system based on (x,y) coordinates that can be read by special digital pens such as the Magicomm G303[6]. After decoding the pattern, the pen transmits the (x,y) positions to a computer through a wireless Bluetooth connection.

[3]NEPOMUK (Networked Environment for Personalized, Ontology-based Management of Unified Knowledge) is funded by the European Union IST (Grant FP6-027705). For more details see the project homepage http://nepomuk.semanticdesktop.org/xwiki/bin/view/Main1/

[4]http://www.wacom.com

[5]http://www.anoto.com

[6]http://www.magicomm.co.uk

Even though the digital pen retrieves the position of the written annotation from paper, this information is usually not sufficient to access the digital content to which an annotation refers. In order to associate the recognized pen-based annotation with a specific word, sentence, or section within the digital document, a precise mapping between the digital content and its printed version has to be performed. In this paper we apply an OCR on the pdf-version of any printed document and store the direct layout mapping in a specific XML-file.

The processing system for handwritten annotation builds on recent advantages in handwriting recognition and gesture recognition. The problem of handwriting recognition has been considered for more than 40 years [PS00]. Nowadays, the first successful commercial applications are available on the market, which could be applied for this stage. For analyzing the gestures, we use the iGesture framework [SKN07]. This framework recognizes any pen-based gestures and translates them into the corresponding digital operations.

Since the focus of this paper is the semantic interpretation of the annotations, only a few details of the document analysis part are presented in this section. For further information on this part, refer to [LSD+08].

4 Methodology

In the approach proposed in this paper, the user can make two types of annotations, marks (of text), and comments. To mark a text, the user can make parentheses around the desired text passage (see the right-angle strokes around the word NEPOMUK in Fig. 2 for an example). Subsequently, he or she can write some text as an annotation to that marked text (e. g., "Title" and "Author" in Fig. 2). To make a comment, the user can write a text at any place in the document without marking a text (the comment "Seen at ..." above the title in Fig. 2).

After the handwriting and gesture recognition methods have been applied as described in Section 3, the type of annotation (mark/comment) can be derived easily. The system simply needs to check whether a text has been marked with gestures before or not. Afterwards, the semantics of the annotations have to be determined. The methodology for that will be described in the rest of this section.

The integration of handwritten semantic annotations into the PIMO requires the identification of the instance `<document>`[7] which is associated with the annotated document. For this, the absolute path of the file is used. If no such instance exists yet, it will be created.

The inclusion of a comment (an annotation without a reference to a marked text) is done by generating a comment instance with the received parameters and linking it to the document instance. A link to the received raw pen data is also added to the PIMO.

Marks are suited to extend the PIMO with any facts (RDF-Triples) through paper-based interaction with documents. The user can mark any text segment and annotate it by writing expressions which are also applied in his or her personal knowledge space. Note that

[7] Note that the parentheses < and > denote PIMO resources in this paper.

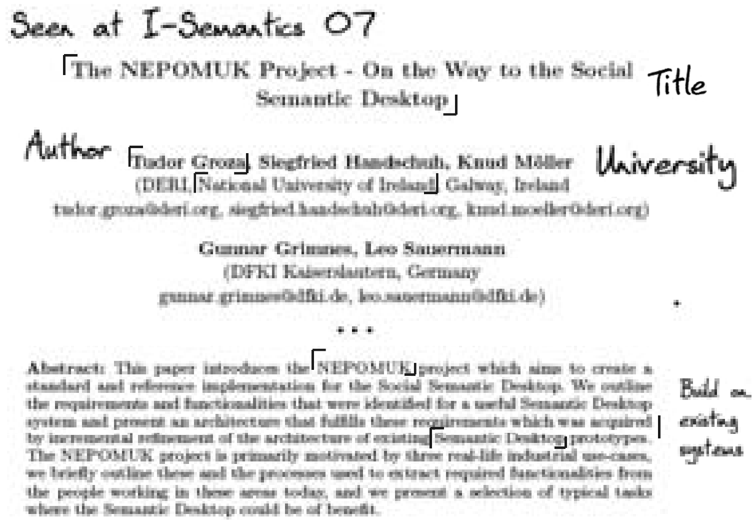

Figure 2: Annotations on the document which are interpreted by Semantic eInk

in general the user knows the relations which are available, because it is a personalized knowledge space. Now the principal task is to find the intended meaning of the annotation in the context of the marked text. Therefore, a syntactic matching of the recognized annotation against the available labels of PIMO-resources is required. If a matching PIMO-resource is found, the type of this resource (denoted as `<annotation>` in rest of this section) determines the further steps:

Property If the property `<annotation>` is applicable for documents we assume that it defines the relation between the document and the marked text. Otherwise, we consider the inverse relation if available. In case of success, `<annotation>` is replaced by its inverse. Note that a property is applicable for documents when the most special class of the document instance is the same or a subclass of the domain of the property. The two possible kinds of properties require different conditioning:

- `<annotation>` is a datatype property: Then the marked text segment (denoted as `marked text`) has to be its value. If the triple `(<document>, <annotation>, marked text)` does not exist yet, it is added to the PIMO. For example, when "title" is defined as a property with literal value, we assign `marked text` as the document's title.
- `<annotation>` is an object property: Then ``` ``marked text'' ``` is expected to be a resource in the PIMO. Therefore, we match it against all the labels of PIMO resources. If a resource `<marked text>` is matched, its type is checked to fit to the range of `<annotation>` in order to keep the PIMO consistent. This process is based on subclass relations where the most special class of the resource must be the same or a subclass of the class defining

the range of the property. If no such resource exists, we create a new instance (<marked text>) of the class defined by the range of <annotation>. The triple (<document>,<annotation>,<marked text>) is then added to the PIMO. As an example, we refer to the annotation “Author” in Fig. 2. It matches the property <author> with the range <person>. If the marked text “Tudor Groza” does not match any resource, a person with this label is created, otherwise we check if “Tudor Groza” is a person and extend the knowledge space with the fact, that the person is an author of the document.

Class We assume that the marked text is an instance of the class <annotation>. Therefore, we try to match the resource. If a resource <marked text> exists, we check if its type fits the class <annotation>. That is given when the most special class of the instance is the same or a subclass of <annotation> or <annotation> is a subclass of the most special class of the resource. In the second case we extend the class information of the resource with <annotation>. On the one hand, in case of success, we extend the PIMO with the triple (<document>, <related>, <marked text>), where <related> denotes a general link between two resources.[8] On the other hand, i. e., if no resource with a label marked text is found, we create it as an instance of the class <annotation> and add the triple to link it to <document>. An example for this case is shown in Fig. 2. There, the “National University of Ireland” has the annotation “University”. This is a class in the PIMO and a link to the resource “National University of Ireland” is added to the PIMO.

It is also possible to mark a text without writing an annotation. This might be used for a simple definition of related resources and keywords. If the marked text matches a resource <marked text>, the PIMO is extended with the triple (<document>, <related>, <marked text>). Otherwise the marked text can be added as a keyword or tag to the document, if such a property exists in the PIMO. Note that this could help to improve the results of retrieval algorithms which search for texts in the PIMO.

5 Prototype for the NEPOMUK Semantic Desktop

The prototype implements the introduced system for the NEPOMUK Semantic Desktop. NEPOMUK supports the application development with a package called Nepomuk Client which contains all necessary libraries for the connection and communication with the Nepomuk Server based on XML-RPC. It also provides an easy integration of ontologies required by external applications in the RDF-Store of the Semantic Desktop. In general, our prototype is an independent web-service which uses the Nepomuk Client package.

Fig. 3 depicts the architecture of the prototype. The first component is the virtual printer which overlays the paper with the Anoto dot pattern for printing and performs an OCR

[8]Note that the adequate relation is defined by the underlying ontology of the used Semantic Desktop. The property <related> is included in the eInk ontology see Section 5 and is used by the application. To adapt it to a relation in the namespace of the Semantic Desktop, the two relations have to be associated to one another.

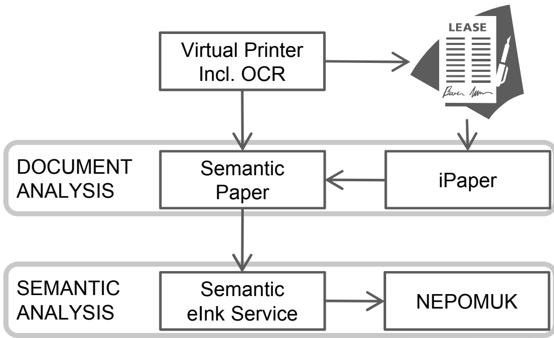

Figure 3: Architecture of Semantic eInk

for generating the XML-representation of the document (see Section 3). The printout is ready to be annotated with the digital pen which sends the recorded stroke information to the iPaper service. This service maps the Anoto-coordinates to the electronic version of the documents and sends the document coordinates to the Semantic Paper component, where handwriting and gesture recognition is performed. For the recognition of text, the Microsoft handwriting recognition engine[9] is used. The latter Sematnic Paper component also searches for the associated marked text in the XML-representation of the document.

The Semantic Paper component finally calls the eInk Web Service posting the path of the annotated document, the recognized text, the interpreted gesture, i. e., mark or comment, the position of the made annotation and the path of generated PDF. The eInk service carries out the semantic analysis using the the Nepomuk Client to query and extend the RDF-Store as described in Sect. 4.

For integrating the handwritten semantic annotation into the NEPOMUK desktop, a Semantic eInk ontology[10] has been developed. It defines a controlled vocabulary for the description of both types of annotations which we differentiate, i. e., mark and comment, and also for the integration of them in an existing Personal Information Model. Note that this prototype ontology focuses on pen-based annotations. For productive use, a similar ontology will be defined which builds on an existing annotation ontology like the Nepomuk annotation ontology[11].

Beside the integration of the semantic annotations into the NEPOMUK desktop, the handwritten notes and their transcriptions are also stored in a PDF document which is referred to by the PIMO. The information is stored in both versions, as strokes and as machine readable PDF-annotations.

Fig. 4 shows screenshot of PIMO-Editor after integrating parts of the annotations depicted in Fig. 2. Note that now the marks 'Title', 'Author', 'University' and the comment 'Seen at...' are now available as digital information. They have been interpreted and integrated in the personal knowledge space.

[9] http://www.microsoft.com/windowsxp/tabletpc/default.mspx
[10] http://eink.opendfki.de/wiki/eInkOntology
[11] http://www.semanticdesktop.org/ontologies/nao/

Figure 4: PIMO-Editor after integrating some annotations depicted in Fig. 2

6 Conclusions and Future Work

In this paper we presented a system for Personal Knowledge Management which first recognizes and interprets handwritten annotations on printed documents, and finally includes the information in the Semantic Desktop. This system incorporates methods from pattern recognition and knowledge management for the several processing stages.

A prototype of the Semantic eInk system has been implemented for the NEPOMUK Semantic Desktop. With this prototype any document can be printed on the Anoto dot pattern and the user can mark the text with predefined gestures. These marks and his or her comments are sent to a web-service, which extracts semantic information from the PIMO to interpret the comments correctly. Finally, the new information is included in NEPOMUK. Note that, even if the prototype has been implemented for the NEPOMUK system, the approach can be applied to any other Semantic Desktop system as well.

The document analysis part of our system still suffers from weak handwriting recognition results. Therefore, we plan to use more specialized recognizers. Furthermore, we may reduce the lexicon, i. e., the list of possible recognition alternatives, to a list of words extracted from the PIMO. From such a reduction, a better recognition performance can be expected, since there is less ambiguity [Bun03, PS00, Vin02]. Note that a lexicon reduction also leads to faster response times which is useful for a real-world application. Beside the handwriting recognition, the restriction to predefined gestures might be too strict. In the future we will apply improved gesture recognition methods which may be adapted to new user-defined gestures.

In order to evaluate the usability of Semantic eInk we will perform user studies where participants are asked to annotate papers and to perform tasks in the NEPOMUK system utilizing their annotations. Afterwards, they will have to fill out a questionnaire. From these studies we can infer the usefulness of paper-based annotations for the Semantic Desktop. Possible future research directions will arise from answers in the questionnaire.

References

[Bun03] Horst Bunke. Recognition of Cursive Roman Handwriting – Past Present and Future. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, volume 1, pages 448–459, 2003.

[CPG05] A. Cheyer, J. Park, and R. Giuli. IRIS: Integrate. Relate. Infer. Share. *1st Workshop on The Semantic Desktop. 4th International Semantic Web Conference*, page 15, Nov 2005.

[GHM+07] Tudor Groza, Siegfried Handschuh, Knud Moeller, Gunnar Grimnes, Leo Sauermann, Enrico Minack, Cedric Mesnage, Mehdi Jazayeri, Gerald Reif, and Rosa Gudjonsdottir. The NEPOMUK Project - On the way to the Social Semantic Desktop. In Tassilo Pellegrini and Sebastian Schaffert, editors, *Proceedings of I-Semantics' 07*, pages 201–211. JUCS, 2007.

[LSD+08] Marcus Liwicki, Kinga Schumacher, Andreas Dengel, Nadir Weibel, Beat Signer, and Moira C. Norrie. Pen and Paper-based Interaction with the Semantic Desktop. In *DAS*, 2008.

[PS00] R. Plamondon and S. N. Srihari. On-Line and Off-Line Handwriting Recognition: a Comprehensive Survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.

[QHK03] Dennis Quan, David Huynh, and R. Karger. Haystack: A platform for authoring end user semantic web applications. In *Proceedings of the 2nd International Semantic Web Conference*, pages 738–753, 2003.

[RVH05] Jörg Richter, Max Völkel, and Heiko Haller. DeepaMehta - A Semantic Desktop. In Stefan Decker, Jack Park, Dennis Quan, and Leo Sauermann, editors, *Proceedings of the 1st Workshop on The Semantic Desktop. 4th International Semantic Web Conference (Galway, Ireland)*, volume 175. CEUR-WS, NOV 2005.

[SGRB08] Leo Sauermann, Gunnar Grimnes, and Thomas Roth-Berghofer. The Semantic Desktop as a foundation for PIM research. In *Proceedings of the Personal Information Management Workshop at the CHI 2008*, 2008.

[SH01] Abigail J. Sellen and Richard H. R. Harper. *The Myth of the Paperless Office*. MIT Press, 2001.

[SKN07] Beat Signer, Ueli Kurmann, and Moira C. Norrie. iGesture: A General Gesture Recognition Framework. In *Proc. of ICDAR 2007, 9th Int. Conference on Document Analysis and Recognition*, pages 954–958, 2007.

[SvED07] Leo Sauermann, Ludger van Elst, and Andreas Dengel. PIMO – A Framework for Representing Personal Information Models. In *Proceedings of I-SEMANTICS 2007*, pages 270–277, 2007.

[TMPP05] G. Tummarello, C. Morbidoni, P. Puliti, and F. Piazza. The DBin Semantic Web platform: an overview. In *Workshop on The Semantic Computing Initiative at WWW2005*, 2005.

[Vin02] A. Vinciarelli. A survey on Off-Line Cursive Script Recognition. *Pattern Recognition*, 35(7):1433–1446, 2002.

[VSO08] Max Völkel, Sebastian Schaffert, and Eyal Oren. Personal Knowledge Management with Semantic Technologies. In Jörg Rech, Björn Decker, and Eric Ras, editors, *Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications*, chapter 9. Information Science Reference, JUN 2008.