XCDF: A Canonical and Structured Document Format

Jean-Luc Bloechle, Maurizio Rigamonti, Karim Hadjar,
Denis Lalanne, and Rolf Ingold

DIVA Group, DIUF
University of Fribourg, Pérolles 2 – Bd de Pérolles 90,
1700 Fribourg, Switzerland
firstname.lastname@unifr.ch

Abstract. Accessing the structured content of PDF document is a difficult task, requiring pre-processing and reverse engineering techniques. In this paper, we first present different methods to accomplish this task, which are based either on document image analysis, or on electronic content extraction. Then, XCDF, a canonical format with well-defined properties is proposed as a suitable solution for representing structured electronic documents and as an entry point for further researches and works. The system and methods used for reverse engineering PDF document into this canonical format are also presented. We finally present current applications of this work into various domains, spacing from data mining to multimedia navigation, and consistently benefiting from our canonical format in order to access PDF document content and structures.

1 Introduction

PDF (Portable Document Format [1]) is nowadays a standard format for exchanging documents through the Internet, thanks to its compactness and robust visualization functionalities. Despite these major capacities, PDF documents are difficult to index for information retrieval tasks because their content is most often disorganized due to optimization reasons, and, therefore, do not respects the reading order. Thus, existing indexing systems always need to preprocess the PDF documents in order to extract and structure the content [17, 4]. In fact, the format presents an important drawback: the documents are created by PDF producers, which 1) privilege layout preservation in spite of physical and logical structures; 2) add a multitude of inconsistencies in the document, e.g. extra blank spaces, over-segmented words, etc. [25]; 3) are unable to generate a PDF document in a unique manner: for instance, a producer might consider a table as a graphic whereas another one could consider the same table as an image.

These lacks imply that PDF efficiency inside the search-engines for information retrieval can be improved, but also that end users would not be able to copy-paste textual parts of a document from PDF viewers maintaining the reading order. Therefore 1) a reconstruction of homogeneous text entities (words, lines, and blocks) extracted from a PDF file and 2) a canonical format organizing the original content in respect of structures and annotations are required before any use of the original content.

H. Bunke and A.L. Spitz (Eds.): DAS 2006, LNCS 3872, pp. 141 – 152, 2006.

This paper is organized as follow: in section 2, we present a taxonomy of the techniques and methods used for extracting information from PDF documents. Section 3 presents the XCDF Canonical Document Format, our proposition for representing electronic document in a unique and structured manner. Section 4 is dedicated to XED, our tool for automatically reverting PDF documents into XCDF. Section 5 describes three different applications using XCDF: an extractor and organizer of TV Schedules, a tool for analyzing logical structures of newspapers and, finally, a multimedia meeting browser based on documents. The last section concludes this paper and announces future works.

2 Taxonomy of PDF Analysis Techniques

Nowadays, different works and researches have been accomplished for recovering hidden physical and logical document structures in PDF files and for deriving specific annotation (for instance the reading order or the table of content). Those methods are summarized in table 1.

Table 1. Taxonomy of techniques for PDF document analysis

Document image analysis	Electronic content analysis		
+ matures techniques + independent from document	+ accurate results + access to document hidden information		
	Extending methods	Restructuring methods	
	+ document preserved	+ information easily accessible	
		Conversion	Reverse engineering
			+ content and structures strictly related

The first methodology consists in analyzing the document image to recover the content and the original structures. This method profits of the entire knowledge acquired by researchers in the last decades, applied – in the majority of the cases - to an ideal document, without noise and printed at high resolution [12, 13]. Document image analysis is independent from PDF file inner structure, i.e. documents page can be represented either with PDF primitives (text, graphics and images), or with images. On the opposite, all information contained in electronic documents composed of PDF primitives is ignored.

The second methodology is based on the analysis of documents with electronic content [21]. Parts of these analysis techniques are derived from the classical document image analysis. In general, they make use of the information contained in the electronic version of documents, which is unfortunately rather difficult to access [25]. The main disadvantage of these electronic content-based methods is their complete inefficiency on documents composed only with raster images instead of primitives (text, graphics and images). In [11, 24], we proposed to mix the two methodologies in order to analyze each category of PDF documents.

The second methodology contains two different families of techniques for extracting structures and annotations from documents: extending and restructuring techniques.

In general, extending methods analyze the content of the document in order to reconstitute the original structure and add annotations (e.g. PDF tags on the original raw data without reorganizing document primitives). Specific application and plug-ins often allow an interpretation of those annotations in order to access to structured content and to add new annotations. Extending methods have been applied with interesting results in different works [5, 14, and 18].

Restructuring techniques target to represent the electronic document in a format different from the original PDF such as in XML, which allows accessing easily the information for further uses. The most interesting case of restructuring is the reverse engineering where the document content is analysed in order to be reorganized in respect to the discovered structures. Different researches [3, 7, 8, 9, and 23] and products [29, 15] are based on reverse engineering techniques. Conversion belongs to a special case of restructuring techniques: logical structures are not recovered, PDF files being only transformed into another format, easier to handle. Currently, the amount of PDF converters is very consistent [6, 10, 19, 22, and 30].

In the following sections, we present the XCDF, a canonical format representing analysed PDF; XED, our system for reverse engineering of PDF, is then described; finally, different user-cases and applications are shown.

3 XCDF, an eXhaustive Canonical Document Format

Reverse engineering of PDF files implies the definition of a format able to represent the reorganized document in a structured and unique manner. This task is in general underestimated because most existing works target at recovering structures as a final scope or need only chunks of linearized document information. A canonical format representing electronic documents in a unique and structured manner would greatly help users and researchers to access easily the document content for further works. From our point of view, such a canonical document format must guarantee the full respect of the following principles:

1. All the primitives contained in the original document (texts, graphics and images) must be represented in the new document in an easy, concise and non-ambiguous way;
2. The textual content must be hierarchically structured: homogeneous text blocks containing lines themselves divided into character sequences (called tokens);
3. This format must be user-friendly.

The canonical format is represented in a structured way with a set of well-defined primitives, where the textual content of a page is segmented into blocks, which are themselves divided into lines and, finally, the latter into tokens (syntactical primitives: words, punctuation signs, numbers, special characters and white spaces). Graphical primitives are labeled as threads, frames or general paths. Finally, images are represented with information on their bounding boxes and a reference to their source files. The following partial DTD emphasizes the document hierarchy and describes the main primitives of the canonical format in more details:

```
<!ELEMENT document (fonts?, page+)>
<!ELEMENT fonts (font+)>
<!ELEMENT page (image*, graphic*, frame*, thread*, textblock*,)>
<!ELEMENT graphic (path*)>
<!ELEMENT path (line, cubic, quadratic)*>
<!ELEMENT textblock (textline*)>
<!ELEMENT textline (token*)>
```

This DTD only shows the elements, not their attributes. Each visible element contains attributes for its position, color, style and others specificities. For instance, the "token" element has a special attribute named "content" containing the syntactical text primitives encoded with the UTF-8 standard.

4 XED: The Canonical Format Builder

This section presents XED, a system that reverse engineers original PDF files into a canonical XML form: XCDF. XED proceeds in two main steps: firstly, it converts the PDF document in an internal Java tree, normalizing the primitives of the original document and taking into account all types of embedded resources such as raw images and fonts. Secondly, XED analyzes the internal Java tree document for recovering physical structures and representing them in the canonical format.

The reverse engineering method was developed and tested on a dataset containing documents with complex physical and logical structures such as newspapers. Documents with simpler layout have also been tested successfully so far, without supplemental and specific calibration of the system.

Since the extraction phase has already been exhaustively presented in [11, 24], the following of this section concentrates only on the documents physical structure analysis in order to produce the document canonical representation. The method proceeds in ten key steps:

1. Trim all text primitives in order to remove all superfluous white spaces (being part of text primitives as well as standalone). This is a crucial step because, sometimes, a word could have white spaces between its own letters.
2. Create a layer per angle α existing in the text primitives. The number of layers is then equal to the number of different angles existing in the document.

Following steps are applied for each layer:

3. Apply a linear transformation to rotate the current layer text primitives of an angle $-\alpha$ about the origin of the 2D space (so every text primitive is finally in a horizontal position, its angle being equal to zero).
4. Merge text primitives horizontally to obtain new strings, given a dynamic[1] distance threshold. This results in a basic text segmentation;
5. Tokenize the previous strings using a separators list and obtain isolated pure words, numbers, punctuation signs and special characters. These new-segmented entities are actually textual primitives in the canonical format; henceforth, we will call them "tokens".

[1] Thresholds are dynamically generated from the font size and occasionally from other relevant text features.

Fig. 1. Blocks with overlapping bounding boxes are correctly merged

Fig. 2. Result of retroactive merging

6. Merge horizontally the tokens into lines with a dynamic distance threshold, adding required nonexistent white spaces between consecutive token of a line;
7. Merge the lines vertically into blocks given a dynamic distance threshold. This part of the algorithm is similar to connected component detection in image processing (in the paradigm, we can consider lines being pixels and neighbors being adjacent lines). This technique allows to detect overlapped text blocks and to avoid unwanted merging (see fig. 1).
8. Apply retroactive merging. Parse all blocks content in order to recompose over-segmented lines. This over-segmentation is inducted from the dynamic low thresholds generated in step 6: in case of justified lines (fig. 2), the distance between strings increases and, consequently, lines are not always correctly merged. This step corrects all over-segmentation errors in blocks.
9. Apply inverse linear transformation to rotate every text primitive of the current layer back into its original angle.
10. Parse all canonical tokens and label them with a syntactical attribute "word", "number", "white space", "punctuation" or "symbol".

For concision purpose, the ten steps presented above have been simplified and adapted for western newspapers (Latin languages). Arabic and oriental languages are not currently taken into account. Our algorithm is effective with any text angle, since the merging occurs only in a specified layer at a time (text primitives with identical angles). The reading order is perfectly respected; indeed, we sort tokens, lines and blocks before each merging. A relevant feature of our system is its ability to generate the canonical format over any PDF file without specific customization. Indeed, thresholds are not static, they are generated by ratios of dynamic values (font size, interline, etc). Moreover, used ratios (thus thresholds) tend to be minimal, over-segmentation being preferable than under-segmentation.

Fig. 3. Canonical text generation steps

Fig. 3 shows the application of our method on an extract of the International Herald Tribune electronic version. The four images illustrate respectively: the raw segmentation, the canonical tokens, text lines and text blocks. An overview of the XCDF file generated for the current example is presented below (only a subset of element attributes is shown).

```
<textblock x="81" y="374" w="145" h="137">
  <textline x="81" y="374.85" w="145" h="32">
    <token size="32" content="Bush"/>
    <token size="32" content=" "/>
    <token size="32" content="plans"/>
  </textline>
  <textline x="81" y="409" w="140" h="32">
    <token size="32" content="to"/>
    <token size="32" content=" "/>
    <token size="32" content="support"/>
  </textline>
  <textline x="81" y="444" w="116" h="32">
    <token size="32" content="a"/>
    <token size="32" content=" "/>
    <token size="32" content="'"/>
    <token size="32" content="strong"/>
  </textline>
  <textline x="81" y="479" w="105" h="32">
    <token size="32" content="Europe"/>
    <token size="32" content="’"/>
  </textline>
</textblock>
```

An evaluation of the XCDF file generation has been performed on a set of representative Latin newspapers front pages, i.e. *La Liberté*, *Le Monde* and the *International Herald Tribune*. For each newspaper, 10 front pages have been extracted and represented in the canonical format. Table 2 shows the percentage of correct tokens, text lines and text blocks detected, in respect to human judgement.

Table 2. Evaluation of canonical format generation

	La Liberté	*Le Monde*	*International Herald tribune*
% of correct tokens	99.90	99.94	99.94
% of correct text lines	99.24	99.57	99.47
% of correct text blocks	97.00	98.26	98.96

5 Applications Using XCDF and XED

This section presents three applications based on XCDF, the canonical format generated with XED: first, an implementation extracting daily TV schedules is presented; second, a tool for the logical restructuring of newspapers is overviewed; finally, the integration in a multimedia meeting browser is introduced.

TV Schedules

Given our canonical document format, the great deal is now to reconstruct the underlying logical structure of different classes of documents. We first tried to generate the logical information of TV schedules. To reach this purpose, we elaborated a simple DTD corresponding to our requirements (for concision purposes, attributes are omitted):

```
<!ELEMENT tvprogram (tvdate+)>
<!ELEMENT tvdate (tvchannel+)>
<!ELEMENT tvchannel (tvshow+)>
```

We then downloaded various standard PDF files from the Swiss TV website (fig 4). We generated one week of TV schedules for 6 different TV channels, so we got 42 PDF files. After that, we generated the 42 associated XCDF files. Since there, we analysed the canonical files as follows:

1. Removal of superfluous information, in our case: images, graphics, and texts using small fonts;
2. Seeking of the most pertinent information: the TV show times. The simplest way to achieve this was to apply a regular expression over all the remaining text blocks: "\d{2}:\d{2}" in Perl style.
3. Retrieving the text lines corresponding to the TV show times. This was done by analysing relative positions.
4. Label text lines with title and description attributes. This was done simply by querying the font face (bold for title and normal for description).

We finally got the XML logical structure corresponding to the previous DTD:

```
<tvprogram descripiton="TSR television programs">
 <tvdate year="2005" month="09" day="21">
  <tvchannel channel="TSR2">
   <tvshow p="am" h="06" m="40" t="Zavévu" d="" />
   <tvshow p="am" h="06" m="42" t="TiTeuF" d="Tchernobyl" />
   <tvshow p="am" h="06" m="50" t="Shin Chan" d="Maman a …" />
   <tvshow p="pm" h="20" m="00" t="Banco Jass" d="" />
   <tvshow p="pm" h="20" m="05" t="Le doc nature" d="Jura …" />
  </tvchannel>
</tvprogram>
```

The logical file generated from the 42 canonical files was perfect. Of course, the physical layout was not very complex and so was it for the logical layout. This simplicity allowed us to implement a hard coded algorithm. These results prove the relevance of our canonical format and open the field for more complex document logical restructuring.

Fig. 4. Extract of TV schedule (September 21, 2005)

Dolores

From the canonical document format presented above, Dolores (**Do**cument **Lo**gical **Res**tructuring, a new tool under development) aims at recovering the underlying logical structures of documents (newspaper, scientific papers...). This knowledge could drastically improve search, retrieval and document alignment due to a more precise indexing (profiting of logical information). Currently, Dolores focuses on the newspaper class because it offers a lot of interesting and relevant features: a rich layout with a lot of typographical and topological information as well as deep logical hierarchies.

So, the first step was to define a logical format able to represent, in an adequate way, the logical structure of the newspapers. No pure physical data like position, width, typographical information or even the page on which elements are located are given; instead, a link to the physical representation described by the canonical format is established by means of unique identifiers. The newspaper class logical format is subject-centered, giving a way to group articles belonging to a same theme. It is also designed to ignore topological complications, e.g. an article distributed on different pages. Finally, this format enables to differentiate various types of articles (e.g. news or interviews). Figure 11 shows an example of this logical format applied to an article.

The method we applied to recover the logical structure from newspapers was inspired by the approach introduced by Souafi-Bensafi et al., detailed in [27]. The authors focused on periodic magazines. They labeled the text blocks by means of a naïve bayesian network, justifying this probabilistic approach by the need to recover errors inducted from OCR. As we do not rely on OCR for the physical structure extraction phase but on PDF electronic documents, we opted for artificial neural networks (ANN). So, we kept their bottom-up scheme: first label the basic text blocks (canonial text blocks in our case) with an ANN using mainly topological and typographical information and, later on, reconstruct the logical structure using previously labeled text blocks. More precisely, this reconstruction is based on the one hand on the results of the labeling generated by the ANN, and on the other hand on geometric information. Small deterministic automatons have been implemented in order to describe general article disposition for a given class of newspaper. For example, the typical article shown on fig. 5 begins by a title, followed by the content of the article (which can be composed of text blocks or images). This particular structure is a simple one, but more complex structure can also be described with this mechanism. Rules are then used to reconstruct a logical structure, based on the information detailed above.

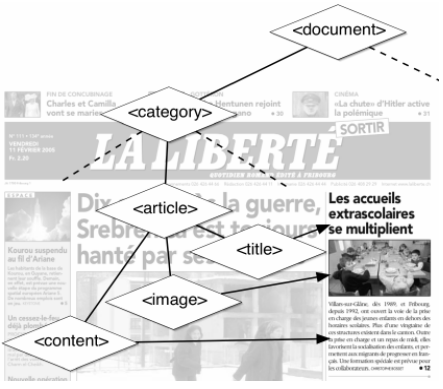

Fig. 5. Example of a logical structure

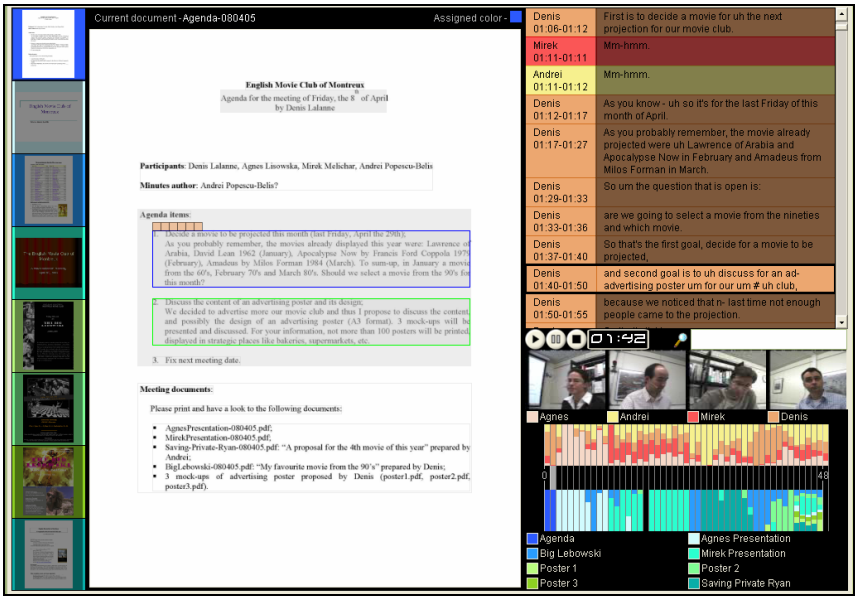

Fig. 6. A screenshot of JFriDoc, a document-centric multimedia meeting browser

Our first experiments gave encouraging results and allowed us to underline some difficulties to overcome. It is relevant to note that these results were first experimentations, but they proved the viability of our approach. More extensive and complete research on logical structure reconstruction will be conducted in the near future.

JFriDoc Multimedia Browser

Building document-centric multimedia meeting browser requires numerous preliminary multimodal analyses. A meeting is an event containing discussed documents, videos of participants and presentations, audio and further. The main tasks required

before being able to query or browse meetings, using documents, are multimedia mining and indexing, document structures extraction and finally multimodal document alignment.

XED has been used in different prototypes of multimedia browsers and in particular with an implementation of FriDoc [26], based on JFerret [14, 28].

JFriDoc needs XED for extracting the physical structures from documents, which are then manually annotated with logical information using the Inquisitor tool [26]. Finally, the resulting canonical physical structure is matched with the structured meeting dialogues transcription in order to thematically align them and, thereafter, enrich documents with time indexes [20].

In the next future works, XED will be fully integrated in the multimedia browsers FaericWorld [26], exploring new types of annotation allowing indexing and alignment of documents.

6 Conclusion

This paper presents XCDF, a canonical format for representing electronic document in a unique and structured way; it is also an entry point for further researches on electronic analysis. A taxonomy of existing systems for PDF extraction and analysis has been presented, organizing systems according to the analysis methods used: either based on image or on electronic content analysis. The latter either extends the original document with annotation or restructures the content in a different format. We then presented the algorithm we developped, which restructures the document in XCDF, a canonical format based on XML and well-defined properties guaranteeing an easy access to document structured content. XED is a system allowing reverse engineering of PDF files into their canonical representation. This system needs a first processing step for extracting the electronic content and a second one for analyzing the electronic content, in particular for extracting the physical layout. The last section of this paper presents three applications exploiting XCDF and XED. The first one extracts TV schedules from PDF documents and organizes them into an XML format. The second application is a system for extracting newspapers' logical structures. Finally, the last application is a multimedia meeting browser using the restructured PDF as an interface to access other medias.

Future efforts will focus on the development of the logical analyzer Dolores, specialized with newspapers, and on the creation of document-centric annotations for multimedia browsing, such as classification of document by type (for instance, newspapers, research articles, papers, etc.).

References

1. Adobe PDF reference, http://partners.adobe.com/asn/tech/pdf/specifications.jsp
2. Adobe's Online Converter, http://www.adobe.com/products/acrobat/access_onlinetools. html
3. Anjewierden, A.: AIDAS: Incremental logical structure discovery in PDF document. Sixth International Conference on Document Analysis and Recognition (ICDAR'01), Seattle, USA (2001) 374-377

4. Anjewierden, A., Kabel, S.: Automatic indexing of documents with ontologies. 13th Belgian/Dutch Conference on Artificial Intelligence (BNAIC 2001), Amsterdam, Holland (2001) 23-30
5. Bagley, S. R., Brailsford, D. F., Hardy, M. R. B.: Creating reusable well-structured PDF as a sequence of component object graphic (COG) elements. ACM Symposium on Document Engineering (DocEng'03), Grenoble, France (2003) 58-67
6. BCL, http://www.bcltechnologies.com/document/index.asp
7. Chao, H., Fan, J.: Layout and Content Extraction for PDF Documents. IAPR International Workshop on Document Analysis Systems (DAS'04), Florence Italy (2004) 213-224
8. Chao, H., Xiaofan, L.: Capturing the Layout of electronic Documents for Reuse in Variable Data. Eighth International Conference on Document Analysis and Recognition (ICDAR'05), Seoul, Korea (2005) 940-944
9. Futrelle, R. P., Shap, M., Cieslik, C., Grimes, A. E.: Extraction, layout analysis and classification of diagrams in PDF documents. Seventh International Conference on Document Analysis and Recognition (ICDAR'03), Edinburgh, Scotland (2003) 1007-1012
10. Glance, http://www.pdf-tools.com/en/home.asp
11. Hadjar, K., Rigamonti, M., Lalanne, D., Ingold, R.: Xed: a new tool for eXtracting hidden structures from Electronic Documents. Document Image Analysis for Libraries (DIAL'04), Palo Alto, USA (2004) 212-221
12. Hadjar, K., Hitz, O., Robadey, L., Ingold, R.: Configuration REcognition Model for Complex Reverse Engineering Methods: 2(CREM). 5th International Workshop on Document Analysis Systems (DAS'02), Princeton, New Jersey (2002) 469-479
13. Hadjar, K., Ingold, R.: Arabic Newspaper Page Segmentation. Seventh International Conference on Document Analysis and Recognition (ICDAR'03), Edinburgh, Scotland (2003) 895-899
14. Hardy, M. R. B. , Brailsford, D., Thomas, P.L.: Creating Structured PDF Files Using XML Templates. ACM Symposium on Document Engineering (DocEng'04), Milwaukee, USA (2004) 99-108
15. JFerret, http://mmm.idiap.ch
16. JPEDAL, http://www.jpedal.org
17. Lawrence, S., Bollacker, K., Lee Giles, C.: Indexing and Retrieval of Scientific Literature, Eighth International Conference on Information and Knowledge Management (CIKM'99), Kansas City, USA (1999) 139-146
18. Lovegrove, W. S., Brailsford, D. F.: Document analysis of PDF files: methods, results and implications. Electronic publishing (1995) 207-220
19. MatterCast, http://www.mattercast.com/default.aspx
20. Mekhaldi, D., Lalanne, D., Ingold, R.: From Searching to Browsing through Multimodal Documents Linking. Eighth International Conference on Document Analysis and Recognition (ICDAR'05), Seoul, Korea (2005) 924-928
21. Paknad, M. D., Ayers, R. M.: Method and apparatus for identifying words described in a portable electronic document. U.S. Patent 5,832,530 (1998)
22. PDFTextStream, http://snowtide.com/home/PDFTextStream
23. Rahman, F., Alam, H.: Conversion of PDF documents into HTML: a case study of document image analysis, Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers 2003, USA (2003) 87-91
24. Rigamonti, M., Hadjar, K., Lalanne, D., Ingold, R.: Xed: un outil pour l'extraction et l'analyse de documents PDF. Huitième Colloque International Francophone sur l'Ecrit et le Document (CIFED 2004), La Rochelle, France (2004) 85-90

25. Rigamonti, M., Bloechle, J.-L., Hadjar, K., Lalanne, D., Ingold, R.: Towards a Canonical and Structured Representation of PDF Documents through Reverse Engineering. Eighth International Conference on Document Analysis and Recognition (ICDAR'05), Seoul, Korea (2005) 1050-1054
26. Rigamonti, M., Lalanne, D., Evéquoz, F., Ingold, R.: Browsing multimedia archives through implicit and explicit cross-modal links. 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI'05), Edinburgh, Scotland (2005) *to be published*
27. Souafi-Bensafi, S., Parizeau, M., Lebourgeois, F., Emptoz, H.: Logical labeling usings Bayesian Networks. Sixth International Conference on Document Analysis and Recognition (ICDAR'01), Seattle, USA (2001) 832-836
28. Wellner, P., Flynn, M., Guillemot, S.: Browsing Recorded Meeting With Ferret. Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI'04), Martigny, Switzerland (2005) 12-21
29. Xed online, http://diuf.unifr.ch/diva/xed
30. xpdf, http://www.foolabs.com/xpdf/home.html