Shadow-driven Document Representation

A summarization-based strategy to represent non-interoperable documents

Matheus Silva Mota
Institute of Computing – UNICAMP
matheus@lis.ic.unicamp.br

Claudia Bauzer Medeiros
Institute of Computing – UNICAMP
cmbm@ic.unicamp.br

ABSTRACT

Document production tools are present everywhere, resulting in an exponential growth of increasingly complex, distributed and heterogeneous documents. This hampers document exchange, as well as their annotation, indexing and retrieval. Existing approaches to these tasks either concentrate on specific formats or require representing document's content using interoperable standards or schema. This work presents our effort to handle this problem. Rather than trying to modify or convert the document itself, our strategy defines an intermediate and interoperable descriptor – *shadow* – that summarizes key aspects and elements of a given document, improving its annotation, indexation and retrieval process regardless of its format. Shadows can be used with different purposes, from semantic annotations and context-sensitive annotations, to content indexation and clustering.

Keywords

Document Summarization, Semantic Web, Annotations

1. INTRODUCTION AND MOTIVATION

This paper describes a Master work supervised by Claudia Bauzer Medeiros, being developed in the Institute of Computing, at the University of Campinas – UNICAMP. The work started in 2010 and is expected to finish by 2012

The proliferation of document formats is a result of both specific environments and multiplication of authoring tools. In most cases, such tools have not been conceived to produce files with explicit structure and interoperable formats, strongly coupling the content to the file structure and software representation [11, 7]. Furthermore, document production tools have offered increasing support for more than flat text, handling also artifacts like charts, tables, embedded multimedia etc. This further increases the problem of document heterogeneity.

In a scenario with high diversity of non-interoperable formats and a large volume of complex documents, challenges arise when it comes to systematize management and storage techniques, retrieval, interpretation and correlation algorithms and new methodologies to present, annotate and mine documents and their internal content. In addition, there are problems related to documents produced to be used in multiple domains – for instance, in the context of scientific research, participating research groups have different needs of document handling [7].

Document management and retrieval systems use three main strategies to deal with large volumes of complex and heterogeneous documents. The first strategy supports only some specific file format, where is necessary to convert the original document to the supported format. The second strategy requires documents that follow interoperable standards (e.g., XML) or schemes. The third strategy considers documents to be a general digital artifact, supporting only metadata and requiring user assistance. The first strategy presents problems when original file preservation is needed. In strategy two, the main difficulty is to handle format diversity, since interoperable formats and predefined schemes are a prerequisite. On the other hand, approach three deals very well with file format diversity, but provides limited support to indexation, retrieval and annotation.

This work presents *Shadow-driven Representation* (SdR), a novel strategy to represent documents independently of format, preserving the original file and handling large volumes of documents. The SdR strategy is based on the concept of descriptors and can improve the process of indexation, annotation, derivation and correlation discovery. Rather than requiring a specific format, we propose an intermediate structure – a *shadow* – that represents key aspects and elements of a document. *Shadows* are then instantiated with interoperable terms that are linked to ontologies. This enhances the process of indexation and retrieval, dissociating document comprehension from the format. A document may have different shadows, depending on context needs.

This paper is organized as follows. Section 2 introduces concepts and related work. Section 3 presents a detailed explanation of SdR. Section 4 presents an implementation of the presented strategy. Section 5 presents a case study where we use shadows to allow semantic annotations in the biodiversity context. Finally, Section 6 presents conclusions and ongoing work.

2. CONCEPTS AND RELATED WORK

2.1 Resource Descriptors

The representation strategy presented here is inspired on the concept of *resource descriptors*. *Descriptors* are structures that summarize aspects of some digital object in order to help its indexing, comparison and retrieval [2]. More

specifically, our representation strategy is inspired by the concept of descriptors borrowed from two research fields: image management and metadata standards.

2.1.1 Metadata and Metadata Standards

Metadata can be seen as a high level description of data. Metadata, or meta-information, is a structured information and regulatory tool to explain, locate, identificate, describe and provide semantic increment to resources, helping users or management tools [12]. These "data about data" or "information about information" can be associated with some resource or parts thereof [6, 3]. Different domains and needs require different metadata vocabularies. Metadata standards propose and define a set of elements that improves data sharing and integration among different users and applications [6]. In this work, as presented in Section 4, we use a set of metadata standards related to documents and other kinds of digital objects to generate an intermediate and interoperable document descriptor – the *Shadow*.

2.1.2 Image Descriptor

An Image Descriptor is a data structure that summarizes the content of an image. According to [2], an image descriptor can be defined as a pair composed of a feature vector and a distance function. The feature vector represents a set of properties (e.g, shape, color, texture) extracted from the images. The distance, or similarity, function is used to compare feature vector through a specific metric [2, 9]. To extract visual properties, image processing algorithms usually focus on specific characteristics of an image and mainly follow two steps: (i) points of interest are identified and pass through a feature extraction process; and (ii) values are computed based on each point of interest, according to the required type of information [9]. Image descriptors have two advantages: (i) the features extracted can be stored for subsequent processing; and (ii) different image descriptors (e.g., based on color, shape etc.) can be combined, implying on scalability.

Image descriptors are particularly helpful in understanding the SdR. Rather than looking for matches of metadata or annotations, or opening a document to extract specific characteristics – which is the usual approach in document management systems –, the SdR strategy pre-processes and extracts points of interest (key elements) of a document. Then, based on the extracted features, we generate an intermediate document descriptor that describes the extracted elements (the shadow). Subsequently, we are able to annotate parts of a document or perform a shadow-based search (via some distance function, as in image descriptors).

2.2 Semantic Annotation and Resource Links

The concept of Semantic Annotation is derived from the textual annotation concept [10]. In computing, such annotations can have different objectives and be structured in many forms, e.g., links, free remarks, tags, floating layers etc. [4]. Annotations are used, among others, to describe a resource, its relations and what it represents. Informal annotations are usually inserted on documents for human consumption. This hampers computer processing and annotation exchange. Semantic annotations appeared with the purpose of third-party interpretation, providing explicit and machine interpretable semantics, as supported by Semantic Web standards [8, 4]. Annotations acquire more semantics when they follow structural schemes and relate concepts and relationships between concepts and/or resources. This strategy allows machine consumption, therefore the development of new types of applications [8], such as text categorization, content and multimodal information retrieval etc.

There are several reference standards/languages to annotate and link XML documents and their fragments, such as XPath, XLink and XPointer. XPath models an XML document as a tree of nodes and provides a URL path notation for element addressing, while XLink allows elements to be inserted into the XML documents to create and describe links between resources. Finally, XPointer defines a language to be used to locate a fragment via a URI, allowing a URI reference to locate some resource. Reference standards are important to this work when is necessary to address link a shadow or a specific part thereof.

3. SHADOW-DRIVEN REPRESENTATION

The SdR – *Shadow-driven Representation* – strategy aims to build an interoperable document descriptor that summarizes key aspects and elements of a document in a XML, allowing its future indexing, comparison, annotation. A shadow can be seen as a generic structure that specifies a document's key elements. These elements (e.g., metadata, pages, paragraphs, embedded multimedia artifacts, sections) are previously defined by users (e.g., research groups may have different interests). Once a set of elements of interest is defined, shadows are instantiated based in this set.

Figure 2 represents an abstraction of the main SdR concepts, where a shadow's components points to elements of the corresponding document (extreme left). Figure 2 also shows the internal structure of the shadow (right part of the figure), where the corresponding document is persisted as a tree that summarizes the document's elements – in this example, document contains pages, which contains paragraphs etc. Those elements that are reflected in a shadow are the elements initially defined by users.

The production of a document shadow is divided in two steps: (a) Definition of the elements of interest that should be present in the shadow; and (b) instantiation of the shadow for each document, based on these elements. Stage (b) is organized in two steps: (i) document analysis for recognition of elements of interest; (ii) production of the document's shadow. These steps are described next.

3.1 Definition of Elements of Interest

Different domains may have different needs of document handling [7]. In the SdR strategy, this implies in different subsets of elements of interest. For instance, consider a collection of documents. A document search system related to some specific domain may be interested on search by abstract, title and authors. Another domain may be interested in finding images or result tables or bibliographic references. Both domains need to define their subset elements of interest. Once these elements are defined, the document analysis and shadow instantiation process for each document from the collection will be driven by this definition. The possibility of defining different subsets of elements and element levels makes the shadow representation scalable.

3.2 Shadow Instantiation

A *Shadow* can be defined as an open and interoperable descriptor that points to domain-relevant elements of documents. The SdR main goal is deal with different document formats equally, through the shadow. To do that, there are two steps in the shadow instantiation process. In the first, the document is automatically analyzed in order to identify

and classify elements. Later, this identification and classification will be useful to produce shadow according to the predefined elements of interest.

The first step on the instantiation process focus on **document analysis and element recognition** – mainly document metadata and elements, such as embedded multimedia, structure information etc. Once all extractable elements are identified, those elements pass trough a categorization process. The analysis and element recognition step is important to drive shadow production.

To produce shadows, this work treats documents as special cases of complex objects [1], i.e., they are self-contained units, defining recursive hierarchical containment structures – e.g., a document contains pages, which contain paragraphs etc. Consider now a collection of documents and a set of elements of interest. To produce shadows for this collection, each document should pass through a analysis and element recognition step. After that, the production of shadows will be driven by the set of elements of interest previously defined. Basically, the algorithm will open each document and extract and categorize its elements. After that, all elements whose types are present on the interest set will be instantiated into a corresponding shadow.

4. IMPLEMENTATION

We present a novel strategy to represent documents independently of file formats. This implies in many challenges, such as *"how to perform document analysis and element extraction independently of formats?"*, *"since shadows should be interoperable, how should internal elements be instantiated?"*. This section briefly presents our implementation strategy for document processing and shadow instantiation.

4.1 Definition of Elements of Interest

As previously presented, to drive the shadow production, users must define a set of elements of interest which should be reflected in the shadow. In our implementation, users can specify elements that will compose a shadow through a predefined XML schema. Basically, users can produce a file that acts like a template. This template contains generic document element types with corresponding ontology or metadata standards terms. Those terms will be instantiated for each element type recognized in the step of document analysis and element recognition.

4.2 Shadow Instantiation

Our approach to instantiate shadows is divided in two steps. In the first, we use our implemented framework to analyse the document and recognize elements. Later, a second application is connected to the framework to produce shadows based on the XML specification defined by users.

4.2.1 Document Analysis and Elements Recognition.

One of the main challenges of this work is to deal with the large volume of documents and file formats. This format heterogeneity hampers the document analysis and consequently the shadow production process. To handle this problem, we implemented a *shadow-builder* over DDEx.

DDEx - Document Data Extractor[1] - is a Java framework, implemented by us, that allows other applications to transparently open and extract the content of documents, regardless of file types. DDEx aims to decouple the content extraction process from content processing [11]. To do that, DDEx uses a set of APIs and a specific software design pattern (Builder, as in [5]) to allow applications to use document content, encapsulating and performing the extraction independently of format. Furthermore, DDEx is scalable to multiple file formats, handling each format specifically, but providing information to applications transparently. Each specialized document analyst works as a back-end module, which recognizes elements from the document's content and implements a standard output API able to produce a sequential stream of descriptive calls, reflecting the document internal structure and content.

[1]http://code.google.com/p/ddex

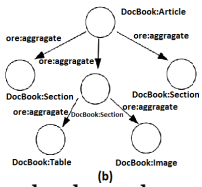
4.2.2 Shadow Production.

DDEx forwards the document content to other systems via a stream of method calls, such as *foundSection* or *foundMultimediaObject* – where all object information and a byte stream itself is transferred. On the shadow production process, these calls are mapped on the instantiated shadow with a corresponding metadata or ontology term. For instance, when the method *foundSection* is invoked, it is instantiated in the shadow as the DocBook [13] element *section*.

Figure 1 (a) shows the set of metadata standards and ontologies adopted in our initial implementation. Its first line, for instance, indicates that the *Docbook* standard is adopted. Moreover, a shadow can be instantiated with other metadata standards and ontology terms. To do that, users must associate elements with other standards or ontology terms on the step of definition of elements of interest. Figure 1 (b) shows some elements of a shadow and the relation between them. Is important to note that the composition and relations between the instantiated elements of a shadow can be also represented with metadata standards and ontology terms – in our implementation, we adopted OAI-ORE, represented in the figure by the prefix `ore:`.

Prefix	Namespace URI	Description
docbook	http://docbook.org/ns/docbook	Docbook namespace
dc	http://purl.org/dc/elements/1.1/	Dublin Core elements
dcterms	http://purl.org/dc/terms/	Dublin Core terms
xmp	http://ns.adobe.com/xap/1.0/	XMP Properties
ore	http://www.openarchives.org/ore/terms/	ORE vocabulary terms
xsl-fo	http://www.w3.org/1999/XSL/Format	XSL-FO formatting terms
rdf	http://www.w3.org/1999/02/22-rdf...	RDF vocabulary terms

(a)



(b)

Figure 1: (a)Adopted metadata standards and ontologys; (b)Abstraction of a shadow structure

5. CASE STUDY: SHADOW ANNOTATION

Since a shadow is an interoperable document descriptor, it can be easily used with different purposes. Once shadows are produced, applications are able to interpret and manipulate it with different objectives. For instance, we are using shadows to allow the creation of links between elements of a shadow and ontologies, producing semantic annotations. By annotation we mean user free remarks or structured statements attached to shadows or a specific element of a shadow. To annotate shadows, we adopted a RDF based schema for describing annotations and an already established XML reference standard – XPointer – to address elements.

Our approach to annotate documents through shadows is called SdA – Shadow-driven Annotation. SdA is very useful, since the annotations can be done with multiple document formats without actually touching the original – local or distributed – file. Furthermore, this strategy isolates the file format and improves the document retrieval process by inserting internal elements of documents into the Semantic Web scenario. Figure 2 illustrates an RDF annotation

(bottom of the figure), showing the relation between an annotation, a specific element of a shadow (highlighted in red) and the original element of a document (extreme left).

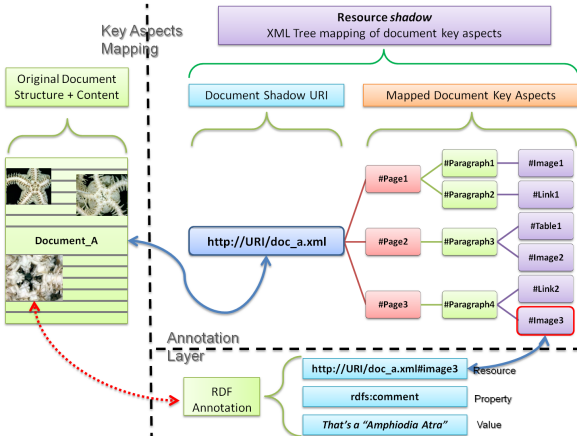


Figure 2: Abstraction of the SdA strategy

Elements of interest in a document can be those usually associated with a paper – such as section, title, paragraph. However, biodiversity researchers are also concerned with other elements – for instance, related animals, photos, result tables, papers cited and database records where the observations of the living beings were recorded. Figure 3 shows an example of a shadow created from documents produced in biodiversity studies, in which researchers need to correlate work (mainly papers) of several scientific domains (e.g., climatology, phenology, biology, pedology) with observations of living beings (plants, animals) and their interactions.

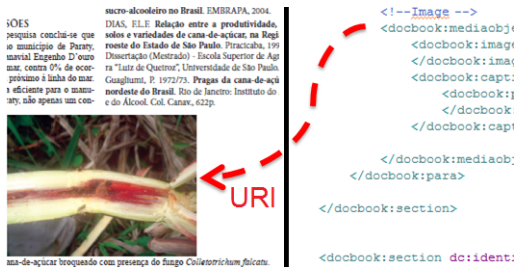


Figure 3: Piece of a document and its corresponding shadow elements

Figure 3 shows an excerpt of a paper on sugar cane agricultural culture and an instantiated corresponding shadow (extreme right of the figure) that represents a photo. At the left, we can see a photo that shows a sugar cane pathogen caused by the *Colletotrichum falcatum* – a fungus, widespread in subtropical regions, that have many synonymous and popular names, such as *Glomerella tucumanensis*, *Physalospora tucumanensis* and "red rot of sugarcane".

Once the shadow contains an element that represents the photo related to *Colletotrichum falcatum*, we are able to address this element (using XPointer) and produce a semantic link with an ontology that describes this fungus. Later, queries like "*plant diseases caused by fungus in subtropical regions*" or "*document and images related to the Physalospora tucumanensis*" or "*images of red rot sugarcane*" will return an indication to this specific image.

6. CONCLUSIONS AND ONGOING WORK

This work proposes a different approach to handle the large volume of documents and format diversity. SdR – Shadow-driven Representation – adapts the notion of "descriptor" to generate context dependent document descriptors – shadows. The main advantages of this approach are: (i) shadows isolate domain-relevant elements of a document from its format and/or location; (ii) shadows may have different granularity levels, based on the domain needs; and (iii) shadows follow interoperability standards, enabling its exchange and machine consumption. To validate the strategy, we implemented a prototype which is able to create shadows, independently of file formats, for documents in the biodiversity domain. Shadows are then used to allow semantic annotations of a document itself or fragments thereof. As future work, the SdR strategy can be applied to other areas, like textual content summarization, document clustering and non-interoperable documents versioning and derivation discovery.

7. ACKNOWLEDGMENTS

Research partially financed by CNPq, FAPESP, CAPES and INCT in Web Science.

8. REFERENCES

[1] M. V. Cundiff. An introduction to the Metadata Encoding and Transmission Standard (METS). *Library Hi Tech*, 22(1):52–64, 2004.

[2] R. da Silva Torres and A. Falcão. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, 2(13):161–185, 2006.

[3] E. Duval, W. Hodgins, S. Sutton, and S. L. Weibel. Metadata Principles and Practicalities. *D-Lib Magazine*, 8(4):1–10, Apr. 2002.

[4] J. Euzenat. Eight questions about semantic web annotations. *IEEE Intelligent S.*, 17(2):55–62, 2002.

[5] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides. *Design Patterns*. Addison-Wesley Longman Publishing Co., Inc., Boston, USA, 1995.

[6] J. Greenberg. Understanding Metadata and Metadata Schemes. *Cataloging & Classification Quarterly*, 40(3):17–36, Sept. 2005.

[7] T. Hey, S. Tansley, and K. Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, Redmond, Washington, 2009.

[8] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1):49 – 79, 2004.

[9] H. Lejsek, F. H. Ásmundsson, B. T. Jónsson, and L. Amsaleg. Scalability of local image descriptors: a comparative study. In *MULTIMEDIA '06*, pages 589–598, NY, USA, 2006. ACM.

[10] E. Oren, K. H. Moller, S. Scerri, S. Handschuh, and M. Sintek. What are semantic annotations??

[11] A. Santanchè, M. Mota, D. P. Costa, N. Oliveira, and C. O. Dalforno. Componere – web authoring based on components. In *Proc. of XV Brazilian Symp. on Multimedia and the Web*, 2009.

[12] J. van Ossenbruggen, F. N., and L. Hardman. That Obscure Object of Desire: Multimedia Metadata on the Web, Part 1. *IEEE MultiMedia*, 11(4):38–48, 2004.

[13] N. Walsh and L. Muellner. *DocBook: The Definitive Guide*. O'Reilly, 1999.