Web Document Description Based on Ontologies

Martin Milička and Radek Burget
Faculty of Information Technology
Brno University of Technology
Božetěchova 2,
612 66 Brno, Czech Republic
email: {imilicka}{burgetr}@fit.vutbr.cz

***Abstract*—The effective document modeling based on the visual features is not always trivial in the case where we want to apply the advance document processing with the visual features reflection. In this paper, we are suggesting a document description based on RDF and ontologies. Such approach allows modeling on several levels of abstraction that is reflecting the document visual features perceived by human. Such approach also advantageously uses the existing tools for the RDF querying and reasoning. The paper presents four levels of the document description where all of them are based on the ontologies but each level represents a different knowledge. The main part of the paper is oriented on the document description based on the Semantic ontology. The ontology is based on SWRL rules where rules assign the elements to a given classes. Finally, we are presenting examples of the class definitions based on the SWRL rules.**

I. Introduction

Current World Wide Web consists of a huge number of documents that are primarily intended to be browsed by human users. When we want to process the WWW documents by a computer (for example for the purpose of the document indexing, classification or information extraction), we have to create an appropriate model of the document. Depending on the particular application, the model may describe different aspects of the document on different levels of abstraction.

In case the HTML and CSS documents, the most used is the standardized Document Object Model (DOM) [13]. It is a low-level hierarchical model that describes the HTML code of the document in a standard machine-processable way that is suitable for the document rendering and programmatic manipulation. On the other hand, for other applications such as the document indexing, some model with higher level of abstraction seems to be more appropriate because the implementation details expressed by the DOM are not significant for the document processing or they may even negatively influence the results [1].

In general, the documents contain two types information: the presented *document content* itself (the text, images and other content objects) and the *visual information* that includes the visual layout of the document and the visual presentation of the contents such as the font styles, sizes, colors and other graphical means. Recent research shows that the visual information may be useful for many applications such as document indexing [14] or content classification [3]. However, there exists no general document model that would allow to explicitly represent the visual information together with the other mentioned types of information about the document. Each of the above mentioned approaches uses a proprietary way of representing the visual information that is limited to the particular application. This makes it impossible to reuse the proposed document analysis and preprocessing methods for other applications.

In this paper, we propose a generic document model that allows us to represent a document simultaneously on different levels of abstraction. It is independent on the actual format of the documents and unlike the most of the existing application-specific models that have mostly a hierarchical structure, it is based on the standardized RDF formalism that allows the representation of different relationships among the individual document parts and their features using a generic graph structure. We also define a set of ontologies for representing different aspects of the documents.

The purpose of the model is mainly to allow sharing the document information obtained by the document preprocessing methods between the document preprocessing algorithms on one side (such as the page segmentation and visual analysis algorithms) and the particular applications such as the document indexing or classification on the other side in a standard way. The chosen RDF technology makes the model very flexible and extensible and moreover, it also allows us to use existing tools for the modeled data querying and reasoning.

The paper is organized as follows. Section II provides the related research. In the section III there are presented the levels of document description. Section IV is mainly concerned to the document description based on Semantic rules. The examples of ontology class description defined using SWRL are presented in section V. Section VI presents the environment of experiments. Finally, section VII draws the conclusion.

II. Related research

The role of visual information in web documents and its usage in information retrieval and knowledge discovery are still in active research. In [9], Liu et al. published data record extraction based on visual features. Also Penna et al. presented an information extraction approach based on visual features in [12]. These types of information extraction have a connection to the textual data for better results. In general, we can meet the document models based on tree structures [4], [6], spatial relation [12] (database approach) or with a partial use of ontologies [8].

Document model based on tree structure is typical for the information retrieval approach because it is similar to the Document Object Model (DOM). However, the hierarchical structure has a limitation in the number of inter-element

ISBN: 978-1-4673-5255-0

relationships because it defines just one direct inter-link for each element.

The ontological and RDF-based models are often used for multimedia description. For instance, multimedia analysis ontology was presented in [7] where it is used for the assisted semantic video object detection. In [11], the authors are publishing ontology for dynamic video scene understanding. MPEG-7 standard [10] became the standard for the all video description works.

In the field of a document description, Eriksson published the paper [8] where he is storing RDF description in PDF files. RDF description contains the annotations of the document objects. However, the visual information is not included in this description. For the logical description of particular parts of a document, the *SALT ontology*[1] may be used. It defines the concepts such as the document content, headline, table, image, etc. This ontology was primary developed for LaTeX documents.

The above mentioned models and ontologies create the base for the document description based on visual features proposed in this paper.

III. Levels of document description

Document model can be created at different levels of perception. In the figure 1, we are proposing four levels of the document description where the leftmost part of the figure defines types of document description.

Our approach to the document modeling is based on an assumption that several steps of analysis must be applied to each document in order to obtain a complete information about the document contents and its visual features.

The individual levels of the document processing are illustrated in figure 1. The whole process starts with document rendering. The output of rendering is called a box model of the document and it basically describes the positions of the individual pieces of the document content on the resulting page and their visual features. Afterwards, the box model is processed by a segmentation algorithm that discovers the visual organization of the page, i.e. basic consistent visual blocks and their relationships. As the next step, the discovered visual blocks may be assigned a role in the document such as header, main content, etc. In some cases, for example in information extraction applications, the areas may be even assigned to concepts from a specific domain (e.g. a date, a personal name, a speech title, etc.) Each level of the process has a strong link between the lower and the higher level. The lower level provides the input for the current level. The output of the current level provides the input for the higher level. The HTML source code and the visual features defined using CSS are the inputs of the rendering process that starts the whole processing. Each level has its characteristic output.

Our proposed document model is based on ontologies that correspond to the above mentioned steps of the document processing. The box model is described using a *box model ontology* where a *Box* is defined as a base element. The segmentation result is described using a *segmentation ontology* that contains an *Area* as a base element. Finally, the logical roles of the detected areas may be described using the *semantic ontology* that establishes the relationships between elements and their logical meaning. Optionally, a *domain ontology* may be used for assigning the individual parts of the document to particular concepts.

We have defined the above mentioned *box model ontology* and *segmentation ontology* using the standard OWL language. The *semantic ontology* is defined by semantic rules in the Semantic Web Rule Language (SWRL). Each element of the document is assigned classes that based on the semantic rules.

The following subsections briefly describe the ontologies of the individual document description levels.

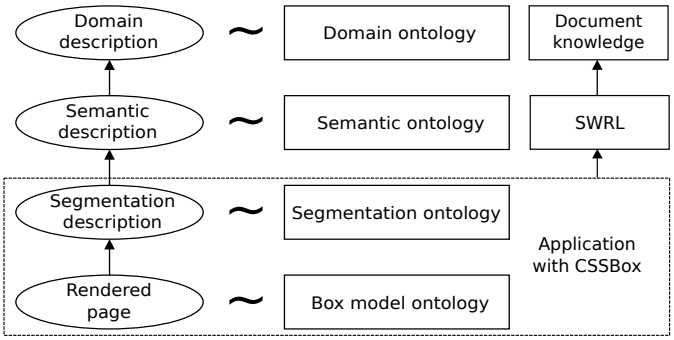

Fig. 1. Levels of document processing

A. Box model ontology

This ontology describes a rendered document where the rendering is based on the source data presented in the HTML document. Visual features are defined by Cascading Style Sheets (CSS).

Box denotes a base element of the rendered document and it follows the box definition from the CSS formatting model [2]. It is characterized as a rectangular area with certain position on the resulting page, width, height and its visual features such as colors and font properties. The boxes are organized in a hierarchical structure that is derived from the source Document Object Model (DOM).

The design of the *Box model ontology* is presented in the figure 2 A). The ontology classes are based on the class *Rectangle* with its characteristic size, position and visual features. The used visual features follow the suggestions from article of Burget and Burgetová [3]. Class *Border* denotes one side border of the particular object.

The *Page* class represents the original document and contains the *sourceUrl* denoting a unique URL of document data property. There is an relationship property *belongsTo* between class Page and some rectangular class where each rectangular class belongs to specific document – Page.

Class *Box* is a subclass of class *Rectangle*. There are also some object properties that have a connection to its content – *containsObject* contains a common information about the objects in the page. *Image* class is an example of the specific object.

[1] http://salt.semanticauthoring.org/ontologies/sdo

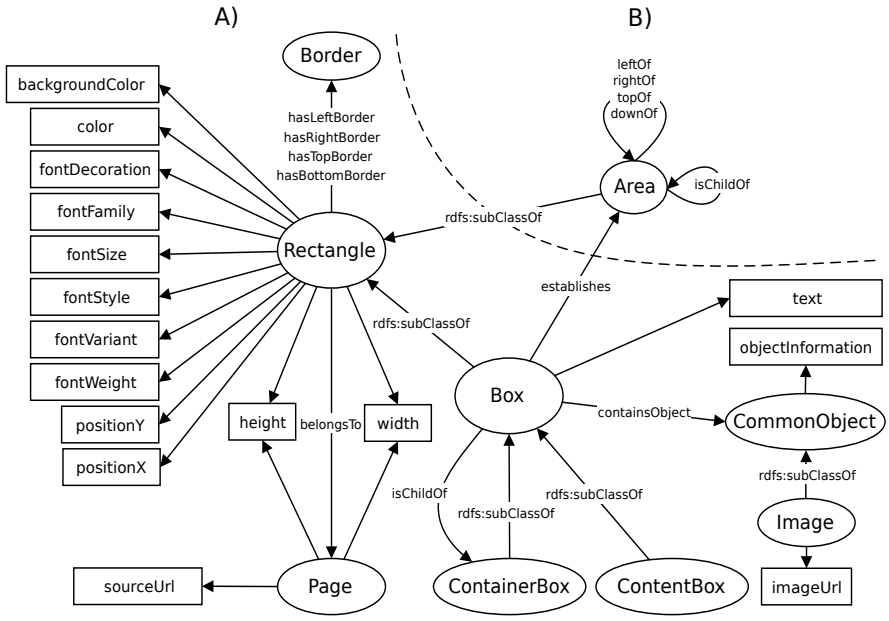

Fig. 2. A) Box model ontology B) Segmentation ontology

The *Box* can be specialized into the *ContainerBox* or *ContentBox* classes where *ContainerBox* represents the box nesting. *ContentBox* represents a Box that does not allow the box nesting; it contains the connections to final objects like images, common objects or textual information.

B. Segmentation ontology

The segmentation ontology represents the individual visually distinguished segments of the document contents in the page. One of the most popular visual segmentation algorithms for web pages was introduced by Microsoft research group in [6]. We can find a lot on modifications; one of them also was presented by Burget and Rudolfová in [4]. However, the ontology itself is not bound to a particular segmentation algorithm.

The segmentation ontology contains classes that are based on the rectangular shape. Segmentation algorithms may have different modifications for each author; however, the idea of the segmentation description is similar. For example, Burget and Rudolfová defined a basic element as *Area* [4]. It is visually autonomous and enables Area nesting.

The Segmentation ontology extends the Box model ontology. The basic *Area* class is defined as a specialization of the Rectangle class from Box model ontology. In the figure 2 B) we can see a segmentation ontology design.

The *Area* class represents the visual areas that are detected during the page segmentation. As was mentioned before, the areas enable hierarchical nesting. The ontology also allows the connection to the individual boxes from the Box model ontology. The *Area* class has the object property *childOf* that has a cyclic dependency. The connection between the area and the box is represented by the *establishes* object property. However, some areas may not correspond to any particular box since they may be created as a visual cluster of multiple boxes. As was mentioned in subsection III-A, box contains the connection to final document data.

C. Semantic ontology

This level of document description defines the parts of content with a specific role in the document. The semantic ontology processing is based on the segmented document.

There can be two types of the semantic document description defined: a manual or an automatic annotation. In the manual annotation, the meaning (a class) of the particular area of document is assigned by a user. The manually assigned classes may be used for training an automatic classifier which is able to assign the classes to new, previously unknown documents automatically. For instance, such approach was presented in [3].

The *SALT ontology* is an example of a semantic ontology for documents because it provides a logical description of particular parts of the document. This ontology was primary developed for LaTeX documents. It defines classes such as *content*, *headline*, *table*, *image*, etc.

According to the visual features of areas that are defined in segmentation ontology, we can define rules for the detection of document content, menus, advertisements, headlines, etc. Semantic ontology is close to the domain ontologies because it defines the semantic parts of document. The details of semantic ontology design are presented in section IV.

D. Domain ontology

This ontology is defined for the particular application domain of the published information. For the documents from the given domain, the individual parts of the document described using the previously mentioned rendering, segmentation and semantic ontologies may be assigned to some concepts of the domain ontology.

The examples of domain ontologies are the FOAF ontology, Event ontology, calendar ontology, etc. Also in the context of conferences we can define a conference program ontology that has classes like date, time, topic, description and authors.

For mapping the individual document parts to the particular concepts of the domain ontology, different approaches may be used such as an approximate tree mapping [5], visual feature classification [3], Name Entity Recognition (NER) or any combination of these approaches.

IV. Semantic ontology

Semantic ontology is mainly based on to semantic rules that are defined by Semantic web rule language[2](SWRL). SWRL creates the extension of the OWL language. It is a combination of the OWL DL and OWL Lite sub-languages of the OWL Web Ontology Language with the Unary/Binary Datalog RuleML sub-languages of the Rule Markup Language.

Each proposed rule is based on the implication between an antecedent (body) and consequent (head). Whenever the conditions specified in the antecedent (body) hold, the conditions specified in the consequent (head) must also hold as well. The antecedent (body) and consequent (head) can be defined with zero or more atoms where each atom represents one assertion of the particular rule. Multiple atoms are treated as a conjunction. Details of the SWRL language are represented in the W3C standard in [15].

The proposed design of Semantic ontology can be divided into two logical parts where the first one describes the spatial position in the page and the second one is based on the semantic part of document. In fact, the semantic part reflects the advanced document knowledge that is similar to a human perception.

In the following subsections we can find detail information about both parts of Semantic segmentation.

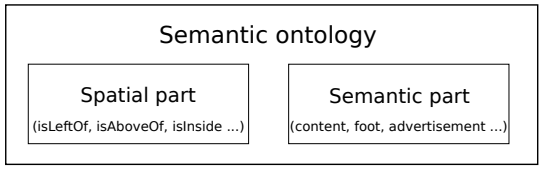

Fig. 3. Schema of semantic ontology

Clarification of the semantic document processing is presented on the figure 4. We can see that the processing depends on the RDF description based on segmentation ontology and on the defined semantic ontology. Since the semantic ontology is defined using semantic rules, we have to apply the reasoning to get the output. This is represented by SWRL processing. The examples of the rules are presented in the section V. Finally, the output of such processing gives the semantic information about the document elements. For the particular element there will be assigned the classes and object properties that is the element following.

[2]http://www.w3.org/Submission/SWRL/

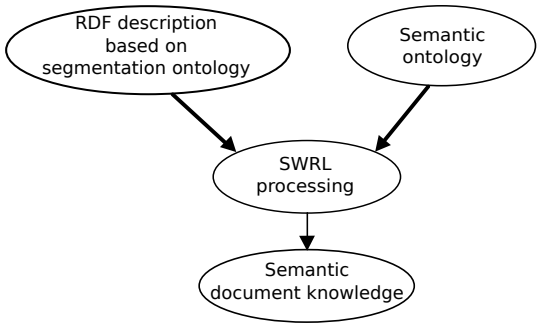

Fig. 4. Semantic ontology processing

A. Spatial part of semantic ontology

The spatial part of the Semantic ontology defines the spatial relationships between the particular document elements. In other words, it defines the visual organization of the web document. In this part there are defined the object properties like *isAboveOf*, *isLeftOf*, *isRightOf*, etc.

These spatial relationships create the base for the advanced knowledge discovery of the particular elements. The semantic part advantageously uses the spatial part in the semantic rule definition.

The spatial part of object properties can be also categorized into the characteristic groups:

- *isLeftOf, isRightOf, isAboveOf, isBelowOf*
- *isHorizontal, isVertical*
- *isInsideOf, isOutsideOf*
- *isDirectNeighbour*

Of course, there can be another user defined object properties. The mentioned properties just provide the idea of the further possibilities.

B. Semantic part of semantic ontology

This part of the ontology is oriented on the logical (semantic) meaning of the particular document elements. The semantic part mainly defines the classes in semantic ontology. Each ontology class is characterized by semantic rules where many of them are containing the object properties defined in the spatial part of the semantic ontology. For instance, the content of document can be characterized as text that is introduced by Headline. Moreover Headline is defined as a text which has font size significantly higher than the average font size.

Similarly to spatial part of the ontology, we can define the characteristic groups of classes. There is no limitation in the class definition. The user can modify this ontology for a particular purpose. It can be used with advantage in the information retrieval because the process of class modification is really fast. The reasoned output can be also stored for further use in the RDF. The groups of classes are the following:

- *Layout-3cols, Layout-2cols*
- *Headline, Content, Foot*

- *List, Menu, Advertisement*
- *(user defined classes), etc.*

V. Examples of SWRL rule definition

This section wants to demonstrate an easy definition of the semantic rules based on the object properties of the ontology classes. It points to the flexibility and easy use in the information retrieval or knowledge discovery.

In the rule definition, there can be also used the built-in functions of SWRL. For instance, we can use functions like *add*, *pow*, *equal*, *greaterThan*, *substring*, *upperCase*, etc.

A. Example – isLeftOf (spatial part)

The example detects the objects that are on the left side from the others. In fact, it returns all elements that are placed on the right from the specific element. In the beginning there are obtained x-left-positions of A and B rectangle, the width of B rectangle, afterwards x-left-position of B and width of B are summed to x-right-position of B. Finally, if the x-left-position of A is greater than x-right-position of B there must be A rectangle left of B rectangle.

```
Rectangle(?a), positionX(?a, ?posLeftXa),
Rectangle(?b), positionX(?b, ?posLeftXb),
width(?b, ?widthb),
add(?posRightXb, ?posLeftXb, ?widthb),
greaterThan(?posLeftXa, ?posRightXb)
  -> isLeftOf(?b, ?a)
```

B. Example – content (semantic part)

This example follows the semantic part. It searches the main content of document. At first, we are searching an object that has the Headline type. Afterwards, we are searching the content below the headline. Such content cannot be on the left and also on the right – it must be right bellow. The font size of the Headline must be higher than in the content. And finally, the height of content must be higher than height of the headline.

```
Rectangle(?a),
Headline(?b), isBelowOf(?a, ?b),
booleanNot(isLeftOf(?a, ?b)),
booleanNot(isRightOf(?a, ?b)),
height(?a, ?heighta),
height(?b, ?heightb),
fontSize(?a, ?fontSizea),
fontSize(?b, ?fontSizeb),
greaterThan(?heighta, ?heightb),
greaterThan(?fontSizeb, ?fontSizea)
  -> Content(?a)
```

VI. Experimental evaluation

We have tested the suitability of the proposed ontological models for the description of real-world web documents where the main part was oriented to the articles of news portals especially from the Czech Republic. The schema of the whole process is in the figure 5. For rendering the documents and obtaining the box model, we have used the CSSBox[3] rendering engine. Subsequently, the segmentation model was created from the box model using our page segmentation algorithm based on [3]. The obtained models have been described using the proposed *box model ontology* and *segmentation ontology* and stored as the RDF data.

The process of semantic inferencing was tested in Protege 4.2 with Pellet reasoner 2.2.0.

We have successfully generated the box and segmentation models for the web documents. However, the accuracy of the semantic rule application has a strong binding to the accuracy of the used segmentation algorithms. When the segmentation algorithm fails to detect the visual organization of the document properly, the semantic rules cannot be successfully applied.

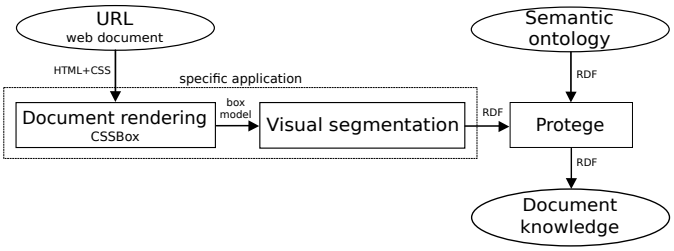

Fig. 5. Processing schema

VII. Conclusion

In this paper, we have presented the levels of the document description and an ontology- and RDF-based document model that allows to describe both the document contents and its visual features on several levels of abstraction. The main advantage of the RDF-based model is the possibility of the usage of existing formalisms and inferencing tools. For the query purpose we can use SPARQL[4] that provides retrieving and manipulating with RDF data.

Special attention was paid on the semantic ontology. We have shown the idea of the semantic rule definition based on SWRL. This approach can be effectively used in the information extraction, because the process of class definition (modification) is really easy and straightforward. The ontology based on semantic SWRL rules can be easily modified for the particular purposes; there are no limitations in the class modification. We have also presented examples of the particular rule definitions.

Acknowledgment

This work was supported by the research program MSM 0021630528, the BUT FIT grant FIT-S-11-2 and the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

[3] http://cssbox.sourceforge.net

[4] Protocol and RDF Query Language (http://www.w3.org/TR/rdf-sparql-query/)

REFERENCES

[1] Alpuente, M., Romero, D.: A Visual Technique for Web Pages Comparison. Electronic Notes in Theoretical Computer Science, 235 (April 2009), pp. 3-18.

[2] Bos, B., Lie, H. W., Lilley, C., and Jacobs, I.: Cascading style sheets, level 2, CSS2 specification: The World Wide Web Consorcium, 1998

[3] Burget, R., Burgetová, I.: Automatic annotation of online articles based on visual feature classification. International Journal of Intelligent Information and Database Systems, 5(4), 2011, pp. 338–360.

[4] Burget, R., Rudolfová, I.: Web page element classification based on visual features. 1st Asian Conference on Intelligent Information and Databases Systems ACIIDS, 2009, pp. 67–72.

[5] Burget, R.: Hierarchies in HTML Documents: Linking Text to Concepts. 15th International Workshop on Database and Expert Systems Applications, 2004, pp. 186–190

[6] Cai, D., Yu, S., Wen, J.-R., Ma, W.-Y.: Vips: A vision-based page segmentation algorithm. Microsoft Research, 2003.

[7] Dasiopoulou, S., Mezaris, V., Kompatsiaris, I., Papastathis, V.-K., Strintzis, M.G.: Knowledge-assisted semantic video object detection. Circuits and Systems for Video Technology, 15(10), 2005, pp. 1210–1224.

[8] Eriksson, H.: The semantic-document approach to combining documents and ontologies. International Journal of Human-Computer Studies, 65(7), 2007, pp. 624–639.

[9] Liu, W., Meng, X., Meng, W.: Vision-based web data records extraction. In: Proc. 9th International Workshop on the Web and Databases, 2006, pp. 20–25.

[10] Martínez, M.J., Koenen, R., Pereira, F.: MPEG-7: The Genetic Multimedia Content Description Standard. Siemens Corporate Research, 2002.

[11] Olszewska, J. I., Mccluskey, T. L.: Ontology-coupled active contours for dynamic video scene understanding. 15th IEEE International Conference on Intelligent Engineering Systems (INES), 2011, pp. 369–374.

[12] Penna, G.D., Magazzeni, D., Orefice, S.: Visual extraction of information from web pages. Journal of Visual Languages and Computing, 21(1), 2010, pp. 23–32.

[13] Wood, L. et al: Document Object Model (DOM) Level 1 Specification, Version 1.0, W3C Recommendation 1 October, 1998

[14] Yi, L., Liu, B., Li, X.: Eliminating Noisy Information in Web Pages for Data Mining. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Washington, DC, USA, 2003

[15] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004, http://www.w3.org/Submission/SWRL (visited 2.8.2013).