Knowledge Modelling for Deductive Web Mining

Vojtěch Svátek, Martin Labský and Miroslav Vacura

Department of Information and Knowledge Engineering,
University of Economics, Prague, W. Churchill Sq. 4, 130 67 Praha 3, Czech Republic
phone: +420-224095462, fax: +420-224095400
e-mail: {svatek,labsky}@vse.cz, vacuram@cuni.cz

Abstract. Knowledge-intensive methods that can altogether be characterised as deductive web mining (DWM) already act as supporting technology for building the semantic web. Reusable knowledge-level descriptions may further ease the deployment of DWM tools. We developed a multi-dimensional, ontology-based framework, and a collection of problem-solving methods, which enable to characterise DWM applications at an abstract level. We show that the heterogeneity and unboundedness of the web demands for some modifications of the problem-solving method paradigm used in the context of traditional artificial intelligence.

1 Introduction

The short history of knowledge modelling is usually viewed as consisting of three periods: first, with prevalent *problem-solving modelling* and stress on traditional AI applications (from the beginning of 80s to mid 90s), second, with growing interest in *domain ontologies* and expansion to more mundane computing environments (second half of 90s), and third, in which ontologies (and to lower degree, problem-solving models) are most often built and used in connection with the *semantic web* initiative (from about 2000). While the initial concept of semantic web assumed manual creation of knowledge annotations, most recent research efforts are aimed towards large-scale *automated* [8] or at least *semi-automated* [10] annotation of websites. This makes the rich variety of website *data structures* such as HTML trees or link topologies, as well as the *processes* of their analysis, primordial knowledge modelling targets. Knowledge models of website analysis applications and tools (in combination with ontologies of the respective problem domains) could facilitate their reuse and reconfiguration for novel tasks, and thus significantly contribute to the deployment of semantic web. However, 'universal' *problem-solving methods* (PSMs), e.g. those from CommonKADS [17], are not easy to use for this purpose. This is mainly due to the heterogeneity and unboundedness of the web space, which acts as 'case data' for knowledge-based reasoning. Instead, as starting point for our knowledge modelling effort, we propose a *multi-dimensional, ontology-based framework*, which enables to characterise apparently dissimilar website analysis applications at the knowledge level. The framework is further enriched with a collection of *more specialised PSMs.*

In section 2 we explain the notion of deductive web mining (DWM) and discuss its knowledge modelling challenges. In section 3 we briefly present the history of our own DWM project named *Rainbow*, as initial motivation for DWM knowledge modelling. In section 4 we describe the four-dimensional (TODD) model of DWM as main contribution of the paper. In section 5 we discuss the applicability of traditional PSMs on typical DWM tasks, and attempt to formulate a collection of PSMs custom-tailored for DWM. In section 6 we describe several existing DWM applications using the TODD model. Section 7 surveys some related projects, and section 8 wraps up the paper.

2 Deductive Web Mining and its Knowledge-Level View

To our knowledge, there has been no concise term that would overarch various methods aiming at automated analysis (or, semantic annotation) of the web space. We therefore suggest a novel label, that of *Deductive Web Mining* (DWM). Our use of adjective 'deductive' is only meant as contrast to *inductive* web mining (i.e. Web Mining in the usual sense); it should thus not be rigorously identified with *deduction in formal logic*[1]. It is inspired by the notion of Deductive *Text* Mining introduced by Kodratoff [12] as synonym to Information Extraction and explained as 'finding instances of a predefined pattern in a set of texts'. Deductive Web Mining is however, in our interpretation, not just another word for Web Information Extraction. While Information Extraction from the web typically amounts to extraction (from another viewpoint, semantic annotation) of *texts* embedded in web pages, our notion of DWM covers all activities where *pre-existing patterns are matched with web data*, be they of textual, graph-wise or, say, bitmap nature. DWM thus subsumes Web Information Extraction, and differs from Inductive Web Mining, which aims at discovery of *previously unseen*, *frequent* patterns in web data. This does not mean that the 'pre-existing patterns' in DWM have necessarily been hand-crafted: inductive learning of patterns (or analogous structures/models) is merely viewed as an activity separate from DWM ('reasoning').

The knowledge engineering research on problem-solving modelling during the 90s was quite systematic, and it is likely to have covered most typical *ways of reasoning* in knowledge-based applications, in a relatively domain-neutral fashion. The aspect in which DWM reasoning might differ is thus merely related to the nature of underlying *data*. Traditional PSMs typically deal with rather compact *objects* with a restricted number of *features*. This holds not only for 'System Analysis' tasks (referring e.g. to the CommonKADS library [17]) such as Diagnosis or Assessment, but also for 'System Synthesis' tasks, e.g. for input components in Planning or Scheduling. If larger structures appear, it is typically only as *static roles* (in KADS sense), and they are relatively *homogeneous*, such as causal networks in Diagnosis. In contrast, the World-Wide Web is a single

[1] Web data structures are 'symptoms' of underlying 'causes' (intentions of website designers), i.e. the reasoning in DWM is rather *abductive*. Yet, some known variants of deduction, such as the *default logic*, might be highly relevant for DWM.

but extremely *large structure*, consisting of an enormous number of *heterogeneous* and *intertwined* objects, such as pages, hyperlinks, HTML trees, blocks of free text, URLs or bitmap images. There is no clear notion of *object-feature relation*; instead, there are the (sometimes substitutable) notions of parthood and adjacency linking *pairs of objects.* Inference targeted on a single object may easily become infinite, and certainly comprises aspects of *recursion* (at the task-subtask level). As we show in section 5, the recursive nature of DWM may even question the strict dichotomy of (non-atomic) *tasks* and *atomic inferences*, imposed by KADS-style modelling.

3 Background: the *Rainbow* Project

The *Rainbow*[2] project represents a family of more-or-less independent web-mining projects undertaken by the same research group[3]. Their unifying principles are commitment to *web-service* (WSDL/SOAP) front-end and agreement on shared *upper-level ontology.* Furthermore, for each *application*, the developers involved also agree on a *domain* and share the source of training/testing *data.* Otherwise, the *formal principles* of analysis methods vary (from linguistic through statistical to e.g. graph theory), and so does the *representation of data*, also nicknamed as 'web view' (such as free text, HTML trees or link topology). In this way, the natural complementarity and/or supplementarity of information inferable from different types of web data can be exploited.

Three application areas have been attacked so far: recognition of web *pornography*, extraction of information about *companies* and extraction of product offers from *bicycle* catalogues. All applications can be characterised as DWM, mostly complemented with inductive learning of patterns. Different *pornography-recognition* services, specialised in image bitmap analysis, HTML structure analysis, link topology analysis, META tag analysis and URL analysis, have been executed more-or-less standalone. Empirical tests however proved that the synergy of different methods significantly improves recognition accuracy [22]. Very simple analysis of *company information* (at the level of single pages) was designed to be executed and integrated via a web browser plug-in, which displayed the structured list of extracted information in a side bar [19]. Finally, the application specialised in *bicycle offer extraction* is currently being sewn together, including (in addition to 'core' DWM tools)

- *the full-text database engine AmphorA* [13], storing web pages as XHTML documents, in a native XML database, as source-data back-end,
- a simple *control procedure* (hard-coded in Java), calling individual DWM tools, and integrating and saving the results,

[2] Stands for 'Reusable Architecture for INtelligent Brokering Of Web information access'. Beyond the acronym (shared with a host of other research projects), the name is motivated by the idea that multiple independent tools for analysis of web data should synergistically 'shed light' on the web content, in a similar way as the different colours of the rainbow join together to form the visible light.

[3] Knowledge engineering group at the University of Economics, Prague.

– the RDF repository *Sesame* [4] for storing the results corresponding to a 'bicycle-offer' ontology (RDF Schema), and, finally,
– an (HTML+JSP) *semantic query interface* with pre-fabricated templates, shielding the user from the underlying RDF query language[4] and enabling a simple form of navigational retrieval [20].

Results of experiments carried out with various DWM tools within the *Rainbow* project were summarised in [19], while [18] refers about synergistic evaluation of multiple tools in a 'company-profile-extraction' task. More information can be found at the project homepage `http://rainbow.vse.cz`. In the rest of the paper, we will use applications of *Rainbow* (beside other applications reported in the literature) to illustrate our knowledge modelling concepts.

4 The *TODD* Framework for Deductive Web Mining

Our framework should enable to position any DWM tool or service within a four-dimensional space. The dimensions of the space correspond to the following:

1. Abstract *task* accomplished by the tool. So far, we managed to characterise any concrete DWM task as instance of either:
   – *Classification* of a web object into one or more pre-defined classes.
   – *Retrieval* of one or more web objects.
   – *Extraction* of desired information content from (within) a web object.

   The *Classification* of an object takes as input its identifier and the list of classes under consideration. It returns one or more classes. The *Retrieval* of desired objects takes as input the (syntactic) *type* of object and *constraints* expressing its class membership as well as (part–of and adjacency) relations to other objects[5]. It outputs the *identifiers* (addresses based on URIs, XPath expressions and the like) of relevant objects[6]. The *Extraction* task takes as input the class of information to be extracted and the scope (i.e., an object) within which the extraction should take place[7]. It outputs some (possibly structured, and most often textual) *content*. In contrast to Retrieval, it does not provide the information about precise location from where the content was extracted[8].

[4] We currently use SeRQL, as generic language of *Sesame*, mainly because of its support for optional path expressions – a useful feature when dealing with incomplete information typically obtained via DWM.

[5] For example: "Retrieve (the XPath addresses of) those HTML tables from the given website that are immediately preceded with a possible 'Product Table Introduction Phrase' (containing e.g. the expression `product*`)".

[6] In the description of this as well as other tasks, we omit auxiliary information on output, such as numerical measures of relevance or uncertainty. These are also typically output by DWM applications, including those developed in *Rainbow*.

[7] For example: "Extract the occurrences of Company Name within the scope of given Company Website".

[8] This is of course merely a knowledge-level view, which does not discourage relevant DWM applications from remembering such information for technical purposes.

2. Type of *object* to be classified or retrieved[9]. The types, such as Document, Hyperlink, or Phrase, represent an upper-level of abstraction of web objects; any class considered in a DWM application should be subclass of such type. This is facilitated by the fact that types correspond to classes of our Upper Web Ontology (see below). The basic assumption is that the type of object is always known, i.e. its assignment is not by itself subject of DWM.
3. *Data type and/or representation*[10], which can be e.g. full HTML code, plain text (without tags), HTML parse tree (with/without textual content), hyperlink topology (as directed graph), frequencies of various sub-objects or of their sequences (n-grams), image bitmaps or even URL addresses.
4. *Domain* to which the task is specific. In this paper, we will consider the domains addressed by our as well as other analysed applications: company sites with product information (incl. specialisations to bicycle offer and casting industry), pornography sites and sites of computer science departments (plus the associated domain of bibliography).

We thus denote the framework as '*task-object-data(type)-domain*' (TODD). Its dimensions are to high degree independent, e.g. *object type* is only partially correlated with *data type*. For example, a document may be classified based on its HTML code, URL, META tag content or position in topology. Similarly, a hyperlink can be classified based on its target URL or the HTML code of source document (e.g. the menu structure containing the respective `<a>` tag). Clearly, not all points of the 4-dimensional space are meaningful. For instance, a META tag content cannot directly be used to classify a hyperlink, since the relation of a META tag (being a special class of HTML document fragment) to a hyperlink is only intermediated by a whole HTML document.

Table 1 demonstrates how the four-dimensional space of the TODD model can be visualised, on the example of services implemented within the *Rainbow* applications mentioned in section 3. Rows correspond to *object types* and columns to *data/representation types*. The fields/columns are filled with corresponding *task acronyms* (hyphen-)followed with *domain acronyms*. The *task acronyms* are C for classification, R for retrieval and E for extraction (the last always pertaining to a whole column). The *domain acronyms* are Ge for general domain (universal web analysis), Po for pornography recognition, Co for general company information extraction, and Bi for bicycle product information extraction. We omit potential columns that do not (yet) correspond to an implemented *Rainbow* service, such as HTML tree data type.

The TODD framework by itself does not offer any added value to DWM application design until augmented with appropriate *ontologies*. As example we can take the *Rainbow* system of ontologies[11], which is indeed structured in correspondence to the dimensions of the TODD framework. First, the three *abstract*

[9] Note that *extraction* aims at content and thus is not unambiguously associated with a particular object.

[10] We alternatively call this dimension 'web view', since the particular representation of data corresponds to a certain view of the complex structure of the web.

[11] Available in DAML+OIL at `http://rainbow.vse.cz`.

Table 1. *Rainbow* services in the TODD space

Data type / Object type	HTML code	Plain text	Frequen-cy	URL	Link topol-ogy	META tags	Image bitmap
Document collection					C-Po, R-Co		
Document			C-Po	C-Po, C-Co	C-Po	C-Po	
Document fragment	C-Po,						
Hyperlink				C-Co			
Phrase		C-Co					
Image							C-Ge, C-Po
Extraction	E-Bi	E-Co				E-Co	

tasks are modelled in a *task ontology*, in terms of their inputs and outputs. Furthermore, the distinction of *data types* and *domains* suggests decomposition into four layers of ontologies. The upper two layers, *Upper Web Ontology* and *Partial Generic Web Models*, are domain–independent and therefore reusable by applications from all domains. The lower two layers, *Partial Domain Web Models* and *Domain Web Ontology* add information specific to a given domain (e.g. product information or pornography). In addition, the top-most and bottom-most layers, *Upper Web Ontology* and any *Domain Web Ontology*, cover all data/representation types, while the middle ones consist each of multiple partial models; each partial model is specific for one data type such as HTML code or link topology. The remaining dimension of the TODD model, *object type*, is reflected in the internal structure of the Upper Web Ontology (see the UML diagram at Fig. 1). The content of each of the four layers of the *Rainbow* system of ontologies as well as their mutual relations and their development process are described in [15]. The ontologies are not yet used for automated reasoning; they merely provide a basis for information exchange among the developers of *Rainbow* tools and applications. Precisely defined *object types and classes* allow for synergy of different tools in the reasoning phase, while precisely defined *data type/representation* is important for sharing training/testing data in the learning phase of individual tools.

5 Problem-Solving Methods of Deductive Web Mining

The textual descriptions of three core *tasks* introduced in the previous section only specify the *input* and *output* of these tasks, in a style analogous to CommonKADS [17]. A natural next step towards reusable knowledge models thus

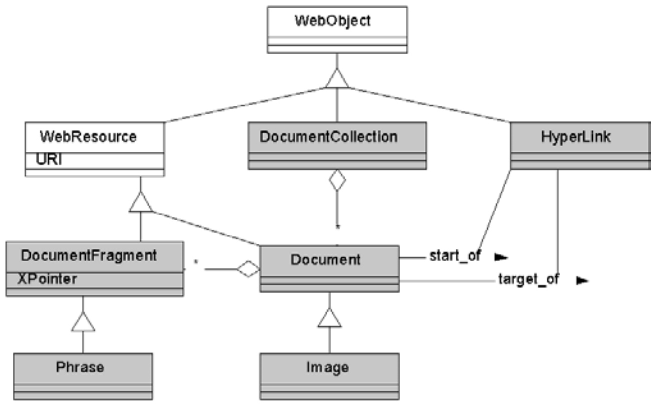


Fig. 1. Upper Web Ontology of *Rainbow*

seems to be the identification of appropriate *problem-solving methods*, and their representation in the form of inference and task diagrams.

Among the three tasks, it is *Classification* that is most appropriate for comparison with existing PSM research. Classification problem solving was recently systematised by Motta&Lu [16]. Their taxonomy of classification problems is mainly derived from the presence (or absence) of a few key features:

1. *Whether the goal is to find one, all or the best solution.* This distinction can well be ported to the DWM context.
2. *Whether all observables are known at the beginning or are uncovered opportunistically (typically at some cost) during the problem solving process.* In DWM, the latter is typically the case (provided we interpret 'observables' as the web objects themselves); the cost is however only associated with download/analysis time, and its increase is smooth—unlike e.g. medical applications, where addition of a single examination may lead to abrupt increase of (financial or social) cost.
3. *Whether the solution space is structured according to a refinement hierarchy.* Presence of class hierarchy is quite typical in DWM; in the *Rainbow* project, it is reflected in concept taxonomies that constitute our ontology, see Section 4.
4. *Whether solutions can be composed together or each presents a different, self-contained alternative.* We believe that in DWM, elementary classification will mostly be carried out over disjoint classes, but can be superposed by multi-way classification with non-exclusive class taxonomies. We discuss this option below, in connection with the *refine* inference of Heuristic Classification.

Let us now present a collection of eight PSMs for DWM. It is rather tentative, yet seems to cover a large part of realistic cases; examples will be given in section 6.

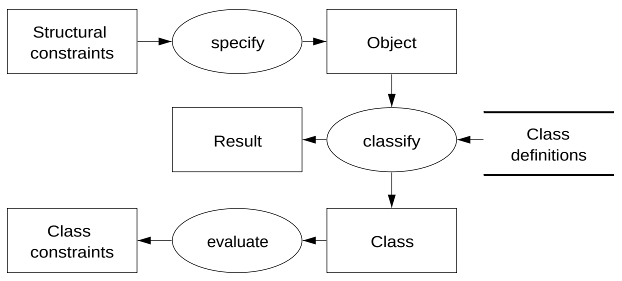
For *Classification*, we could consider three PSMs. *Look-up based Classification* amounts to picking the whole content of the given object (cf. the Overall Extraction PSM below), and comparing it with content constraints (such as look-up table), which yields the class; for example, a phrase is a Company Name if listed in business register. *Compact Classification* also corresponds to a single inference, it is however not based on simple content constraints but on some sort of computation (e.g. Bayesian classification), which is out of the scope of the knowledge modelling apparatus. Finally, *Structural Classification* corresponds to classification of an object based on the classes of related objects (sub–objects, super–objects and/or neighbours). It is thus decomposed to *retrieval* of related objects, their *individual classification*, and, finally, evaluation of *global classification patterns* for the current object. It is therefore *recursive*[12]: its 'inference structure' typically contains full-fledged (Direct) Retrieval and Classification tasks. We can also compare Structural Classification with the well-known Clancey's *Heuristic Classification* (HC) [6], consisting of *abstract*, *match* and *refine* inferences; note that HC was also chosen as the default PSM for classification by Motta&Lu [16], who defined all other models as its reductions. In (DWM) Structural Classification, the *abstract* inference is replaced with *classify* inferences applied on related (contained and/or adjacent) objects; this is due to the 'object-relation-object' (rather than 'object-feature-value') character of web data representation. The *match* inference from HC corresponds to 'evaluation of global classification patterns'. Finally, a *refinement* from general to case-specific solution might rather have the form of classification according to *multiple hierarchies* in DWM. The object is then assigned to the class that is defined as intersection of both original classes. For example, in the pornography application (section 6.1), an object classified as Image Gallery may also be independently classified as Scarce Text Fragment, which yields the class Porno Index.

For *Extraction*, there will be again three PSMs, rather analogous to those of Classification. *Overall Extraction* amounts to picking the whole content of the given object. *Compact Extraction* corresponds to a single inference based on possibly complex computation, which directly returns the content of specific subobject/s of the given 'scope' object. Finally, *Structural Extraction* corresponds to extraction of information from an object via focusing on its certain subobjects. Such objects have first to be *retrieved*, then lower-grained *extraction* takes place, and, finally, multiple content items possibly have to be *integrated*. Structural Extraction is thus equally recursive as Structural Classification.

Finally, let us first introduce two PSMs for the *Retrieval* task. The upper inference structure[13] at Fig. 2 corresponds to Direct Retrieval and the lower one to Index-Based Retrieval, respectively. The names of inferences (in ovals) are mostly borrowed from the CommonKADS library [17], while the knowledge

[12] The notion of recursion previously appeared in knowledge modelling literature, e.g. in the form of Systematic Refinement as form of classification [21]. Here, however, the problem is more severe, since a recursively processed data structure appears in a dynamic rather than static role (in the CommonKADS sense).

[13] We did not show inference structures for Classification and Extraction, due to limited space as well as due to incompatibility of their structural variants with the CommonKADS notation, see below.



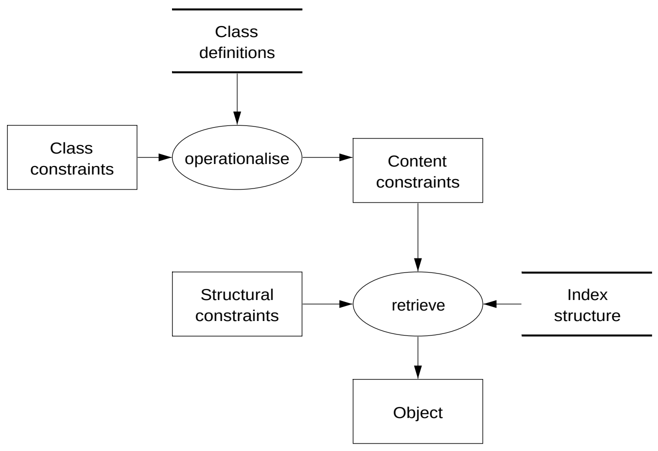


Fig. 2. Inference structures of Retrieval PSMs

roles are more DWM-specific. In *Direct Retrieval*, potentially relevant objects are first retrieved based on structural (parthood and adjacency) constraints, and then classified. Objects whose classes satisfy the class constraints are the output of the method. In the absence of class constraints, the method reduces to the 'specify' inference. In *Index-based Retrieval*, the (abstract) class constraints are first operationalised so that they can be directly matched with the content of objects. Then the objects are retrieved in an index structure (which is considered as separate from the web space itself), possibly considering structural constraints (provided structural information is stored aside the core index).

An interesting issue related to the representation of above PSMs is the possible interaction of different 'time horizons' in one application; static roles may become dynamic when changing the time scale. For example, a typical DWM application may first build an index of a part of the website (or learn class definitions from a labelled subset of objects), and then use the index to efficiently retrieve objects (or use the class definitions to classify further objects). This interaction deserves further study.

6 Example Descriptions of DWM Applications

Let us now describe concrete applications in terms of the TODD framework, including the mapping of tasks to PSMs. For this purpose, we will use an ad hoc semi-formal language with Prolog-like syntax. Its building blocks are *decompositions* of tasks ('heads of clauses') to ordered sequences of subtasks ('bodies of clauses'). Individual task descriptions ('literals') look as follows, respectively:

```
Cla?(<obj_var>, <obj_type>, <data_type>, <domain>, <classes>)
Ret?(<obj_var>, <obj_type>, <data_type>, <domain>, <constraints>)
Ext?(<obj_var>, <obj_type>, <data_type>, <domain>, <content>)
```

The 'predicate' (task name) corresponds to the first dimension in the TODD framework. An extra letter is used to distinguish the PSMs introduced in the previous sections: `ClaS` for Structural Classification, `ClaL` for Look-up based Classification, `ClaC` for Compact Classification; `RetD` for Direct Retrieval, `RetI` for Index-based Retrieval; `ExtS` for Structural Extraction, `ExtC` for Compact Extraction and `ExtO` for Overall Extraction. From the nature of the PSMs follows that each ClaS task can be decomposed to a structure including (among other) one or more subtasks of type Classification; analogously, each ExtS task can be decomposed to a structure including one or more subtasks of type Extraction. In the examples, the 'unification' of a 'goal' with a 'clause head' is always unique; the representation is only 'folded' for better readability.

The remaining three TODD dimensions are reflected by the 'arguments' `<obj_type>`, `<data_type>` and `<domain>`. `<obj_var>` is variable referring to the 'current' object of the task instance: input object in the case of Classification and output object/s in the case of Retrieval. We use object variables (and object types) even for Extraction; however, here they only refer to the scope of extraction, not to a 'current' object as in Classification and Retrieval. `<classes>` is the list of classes distinguished in the classification task (beside named classes, we use the symbol `@other` for a 'complement' class). `<constraints>` is the list of logical expressions determining the set of objects to be retrieved; they correspond to the knowledge roles Class Constraints (class membership restrictions) and Structural Constraints (parthood/adjacency restrictions). Finally, `<content>` is the list of types of content information to be extracted. For simplicity, the language does not consider the cardinality of input and output.

We first describe the *Rainbow* applications: pornography-recognition application [22] and two variants of bicycle offer extraction [20]. Then we, for better coverage, attempt to describe three DWM methods from the literature: the company website classification method by Ester et al. [9], the information extraction application for foundry websites by Krötzch & Rösner [14], and the bootstrapping approach to website information extraction by Ciravegna et al. [5]. The common aspect of all of them is the effort to overcome the limitations of single resource and/or single representation in web mining. However, the symbol-level principles of the methods are different: the first relies on probabilistic reasoning, the second on a mix of domain-specific heuristics, and the third on shallow NLP augmented with knowledge reuse. Due to limited space, we sometimes slightly simplify the structure of applications, without affecting their core principles.

6.1 Pornography-Recognition Application

The upper level of the pornography-recognition process is an instantiation of the *Structural Classification* PSM as discussed in the previous section. In order to classify the whole website (i.e. document collection), symptomatic 'out-tree' topology structures are first sought; their sources (local hubs) can possibly be identified with 'index' pages with image miniatures. To verify that, the hub is examined for presence of 'nudity' PICS rating in META tags (Look-up Classification PSM), for presence of indicative strings in the URL, and its whole HTML code is searched for 'image gallery'-like structures with low proportion of text (which distinguishes pornography from regular image galleries). The analysis further concentrates on individual pages referenced by the hub, and attempts to identify a single dominant image at each of them. The images are then analysed by (bitmap) image analysis methods; in particular, the proportion of body colour and the central position of a dominant object are assessed. In the description, we omit the 'evaluation of global classification pattern' subtasks, for brevity; their inclusion would be straightforward.

```
ClaS(DC, DocCollection, _, Pornography, [PornoSite,@other]) :-
   RetD(D1, Document, topology, General, [D1 part-of DC, LocalHub(D1)]),
   ClaS(D1, Document, _, Pornography, [PornoIndex,@other]),
   RetD(D2, Document, topology, General, [D2 follows D1]),
   ClaS(D2, Document, _, Pornography, [PornoContentPage,@other]).
% classification of index page
ClaS(D, Document, _, Pornography, [PornoIndex,@other]) :-
   ClaL(D, Document, meta, Pornography, [PornoResource,@other]),
   ClaS(D, Document, url, Pornography, [PornoResource,@other]),
   RetD(DF, DocFragment, html-txt, General, [DF part-of D, ImgGallery(DF)]),
   ClaC(DF, DocFragment, freq, General, [ScarceTextFragment,@other]).
% classification of content page
ClaS(D, Document, _, Pornography, [PornoContentPage,@other]) :-
   ClaL(D, Document, meta, Pornography, [PornoResource,@other]),
   RetD(Im, Image, html-txt, General, [Im referenced-in D]),
   ClaC(Im, Image, image, Pornography, [PornoImage,@other]).
```

6.2 Bicycle Application

Navigational Data Access The start-up scenario for extraction of user-oriented information from bicycle-selling sites is centred around *navigation-based* access to individual pages; for the time being, we use URL analysis (admittedly, a weak method only suitable for initial prototype) for this purpose. Subsequently, statistical extraction (Hidden Markov Models) is applied to obtain structured information (products, company address), while phrasal patterns are applied on the (presumably) free text describing the overall company profile. All Retrieval tasks (for navigation-based access to pages as well as at the level of phrases in the last subtask) are mapped on the Direct Retrieval PSM. Most Extraction tasks shown correspond to Structural Extraction. However, at the lowest level, company address is obtained via Compact Extraction, and company description (sentences) are obtained via Overall Extraction. Product information is still a Structural Extraction; if we decomposed it further (not shown in the code), it would however consist of Compact Extraction followed with integration of individual information (names, prices and the like) to a more complex structure.

The real application also includes other types of analysis (topology, META tags, images). We however omit them for brevity.

```
ExtS(DC, DocCollection, _, Bicycle, [products, comp_addr, comp_descr]) :-
   ExtS(DC, DocCollection, _, Bicycle, [products]).
   ExtS(DC, DocCollection, _, Company, [comp_descr]).
   ExtS(DC, DocCollection, _, Company, [comp_addr]).
% extraction of product information from catalogue pages
ExtS(DC, DocCollection, _, Bicycle, [products]) :-
   RetD(D, Document, url, Company, [D part-of DC, ProductCatalogue(D)]),
   ExtS(D, Document, html, Bicycle, [products]).
% extraction of company address from the contact page
ExtS(DC, DocCollection, _, Bicycle, [comp_addr]) :-
   RetD(D, Document, url, Company, [ContactPage(D)]),
   ExtC(D, Document, html, Company, [comp_addr]).
% extraction of general company profile from the profile page
ExtS(DC, DocCollection, _, Bicycle, [comp_descr]) :-
   RetD(D, Document, url, Company, [D part-of DC, ProfilePage(D)]),
   RetD(P1, Phrase, text, Company, [P1 part-of D, ProfilePhrase(P1)]),
   RetD(P2, Phrase, text, General, [Sentence(P2), P1 part-of P2]),
   ExtO(P2, Phrase, text, General, [comp_descr]).
```

Index-Based Data Access Among alternative methods of page access, we are seriously considering the one taking advantage of the available full-text database engine (AmphorA), with its capability of term indexing combined with XML indexing. The parts of website suitable for detailed extraction can be efficiently detected via lexical indicators (e.g. phrases typically occurring nearby product catalogues): some sort of XML environment of the indicators can then be submitted to the extraction tool. Since the overall structure of the application is analogous to the previous one, we only show a fragment, in which Index-based Retrieval of indicative phrases plus Index-based Retrieval of 'mark-up environment' (in a native XML database storing the HTML trees) appears.

```
...
ExtS(DC, DocCollection, _, Bicycle, [products, ...]) :-
   RetI(P, Phrase, text, Company, [P part-of DC, ProductCataloguePhrase(P)]),
   RetI(DF, DocFragment, html-tree, General, [DF contains P]),
   ExtC(DF, DocFragment, html, Bicycle, [products]),
   ...
```

6.3 Website Mining by Ester et al.

The method is not knowledge-based: it relies on Bayesian classification of individual documents (wrt. topics) over the feature space of terms, and then again on Bayesian classification, this time of the whole website (i.e. document collection) over the feature space of individual document's topics. Hence, the overall task pattern amounts to Structural Classification similar to the pornography-recognition task, while the embedded (Bayesian) classifications are Compact.

```
ClaS(DC, DocCollection, _, Company, TopicSet) :-
   RetD(D, Document, topology, Company, [D part-of DC]),
   ClaC(D, Document, freq, Company, TopicSet),
   ClaC(DC, DocCollection, freq, Company, TopicSet).
```

6.4 Company Profile Extraction by Krötzch&Rösner

The overall scheme is similar to the bicycle application, except that product information is only extracted from tables (via heuristics), while phrasal patterns are used in a finer way, to extract not just sentences but names of either customers or quality certificates.

```
ExtS(DC, DocCollection, _, Foundry, [products, customers, certificates]) :-
   RetD(D, Document, html, Company, [D part-of DC, InfoPage(D)]),
   ExtS(D, Document, _, Foundry, [products]),
   ExtS(D, Document, _, Company, [customers]),
   ExtS(D, Document, _, Company, [certificates]).
% product information extraction
ExtS(D, Document, _, Foundry, [products]) :-
   RetD(DF, DocFragment, html, General, [DF part-of D, ContentTable(DF)]),
   ExtS(DF, DocFragment, html, Foundry, [products]).
% customer information extraction
ExtS(D, Document, _, Company, [customers]) :-
   RetD(P1, Phrase, text, Company, [P1 part-of D, CustomerPhrase(P1)]),
   RetD(P2, Phrase, parse-tree, General, [P2 depends-on P1]),
   ExtO(P2, Phrase, text, General, [customers]),
% certificate extraction
ExtS(D, Document, _, Company, [certificates]) :-
   RetD(P1, Phrase, text, Company, [P1 part-of D, QualityPhrase(P1)]),
   RetD(P2, Phrase, parse-tree, General, [CertName(P2), P2 depends-on P1]),
   ExtO(P2, Phrase, text, General, [certificates]).
```

6.5 Bootstrapping Information Extraction by Ciravegna et al.

The approach described in [5] heavily relies on knowledge reuse, thanks to the well-known redundancy of WWW information. We only describe the most elaborated part of the method, targeted at extraction of person names (additionally, various personal data and paper titles are extracted for the persons in question). First, potential names are cropped from the website, and checked against binary classification tools such as context-based named-entity recognisers (Compact Classification), as well as against public search tools (namely, online bibliographies, homepage finders and general search engines) that produce the same binary classification (person name - yes/no) as by-product of offering information on papers or homepages (i.e. Index-based Retrieval). Furthermore, for the results of general web search, the page from the given site is labelled as homepage if the name occurs in a particular (typically, heading) tag. The seed names obtained are further extended by names co-occurring in a list or in the same column of a table. Finally, potential person names from anchors of intra-site hyperlinks are added.

```
ExtS(DC, DocCollection, _, CSDept, [names]) :-
   RetD(P1, Phrase, text, General, [P1 part-of DC, PotentPName(P1)]),
   % named entity recognition for person names
   ClaC(P1, Phrase, text, General, [PName,@other]),
   % use of public search tools over papers and homepages
   RetI(P2, Phrase, freq, Biblio, P1 part-of P2, PaperCitation(P2)]),
   RetI(D, Document, freq, General, [P1 part-of D, D part-of DC, PHomepage(D)]),
   RetD(DF1, DocFragment, freq, General,
         [Heading(DF1), DF1 part-of D, P1 part-of DF1),
   ExtO(P1, Phrase, text, General, [names]),
   % co-occurrence-based extraction
   RetD(DF2, DocFragment, html, General,
         [ListItem(DF2), DF2 part-of DC, P1 part-of DF2]),
   RetD(DF3, DocFragment, html, General,
         [ListItem(DF3), (DF3 below DF2; DF2 below DF3)]),
   ExtS(DF3, DocFragment, text, General, [names]),
   RetD(DF4, DocFragment, html, General,
         [TableField(DF4), DF4 part-of DC, P1 part-of DF4]),
   RetD(Q, DocFragment, html, General,
         [TableField(DF5), (DF5 below DF4; DF4 below DF5)]),
   ExtS(DF5, DocFragment, text, General, [names]),
   % extraction from links
   RetD(DF5, DocFragment, html, General,
         [IntraSiteLinkElement(DF5), DF5 part-of DC]),
   ExtS(DF5, DocFragment, text, General, [names]),
   ...
% extraction of potential person names from document fragments
ExtS(DF, DocFragment, text, General, [names]) :-
   RetD(P, Phrase, text, General,
         [DF contains P, PotentialPersonName(P)]),
   ExtO(P, Phrase, text, General, [names]).
```

7 Related Work

In the *IBrow* project [1], operational PSM libraries have been for developed for two areas of document search/analysis: Anjewierden [3] concentrated on *analysis of standalone documents* in terms of low-level formal and logical structure, and

Abasolo et al. [2] dealt with information search in multiple external resources. Direct mining of websites was however not addressed; IBrow libraries thus do not cope with the problem of web heterogeneity and unboundedness, which motivated the development of the TODD framework. Partially related is also the *OntoWebber* project [11], in which a 'website ontology' was designed. It was however biased by its application on portal building (i.e. 'website synthesis'), and thus did not fully cover the needs of automated analysis; moreover, the problem-solving side of modelling was not explicitly addressed.

8 Conclusions and Future Work

We presented a general *ontology-based knowledge-level framework* for Deductive Web Mining (DWM), determined its role with respect to KADS-style problem-solving modelling, and endowed the core DWM tasks with corresponding *problem-solving methods*. Finally, we demonstrated the usability of the framework on several examples of our own as well as others' *applications*.

Although the TODD framework seems to be useful for communication of (semi-formal) application descriptions among humans, it is desirable to proceed to its fully operational exploitation. We plan to develop a meta-level tool on the top of our *Rainbow* architecture, which would enable *formal verification* and even *semi-automated composition* of DWM applications similar to those referenced in section 6. The experience (and possibly even the tangible results) of the IBrow project might help us in this effort. We plan to transform our models to the UPML language (developed in IBrow) so as to align them with the most recent PSM research achievements. The current research on *semantic web service composition*[14] is also highly relevant.

Finally, a natural next step to DWM modelling would be to cover Inductive Web Mining as well. In a sense, our 'library' thus would be extended (taking analogy with KADS libraries [21], [17]) to cover not only System Analysis but also System Synthesis.

Acknowledgements

The research is partially supported by the grant no.201/03/1318 of the Czech Science Foundation. The authors would like to thank Annette ten Teije and Frank van Harmelen for invaluable comments on drafts of this paper.

References

1. IBROW homepage, `http://www.swi.psy.uva.nl/projects/ibrow`
2. Abasolo, C., Arcos, J.-L., Armengol, E., Gómez, M., López-Cobo, J.-M., López-Sánchez, M., López de Mantaras, R., Plaza, E., van Aart, C., Wielinga, B.: Libraries for Information Agents. IBROW Deliverable D4, online at `http://www.swi.psy.uva.nl/projects/ibrow/docs/deliverables/deliverables.html`.

[14] See e.g. `http://swws.semanticweb.org`.

3. Anjewierden, A.: A library of document analysis components, IBrow deliverable D2b. Online at `http://www.swi.psy.uva.nl/projects/ibrow/docs/deliverables/deliverables.html`.
4. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: An Architecture for Storing and Querying RDF and RDF Schema. In: ISWC 2002, Springer-Verlag, LNCS 2342.
5. Ciravegna, F., Dingli, A., Guthrie, D., Wilks, Y.: Integrating Information to Bootstrap Information Extraction from Web Sites. In: IJCAI'03 Workshop on Intelligent Information Integration, 2003.
6. Clancey, W. J.: Heuristic Classification. *Artificial Intelligence*, 27 - 3, 1985, 289-350.
7. Crubézy, M. Lu, W., Motta, E., Musen, M.A.: Configuring Configuring Online Problem-Solving Resources with the Internet Reasoning Service. *IEEE Intelligent Systems*, 2 (March-April):34-42, 2003.
8. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J., Zien, J.: SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In: Proc. WWW2003, Budapest 2003.
9. Ester, M., Kriegel, H.P., Schubert, M.: Web Site Mining: a new way to spot Competitors, Customers and Suppliers in the World Wide Web. In: Proc. KDD 2002.
10. Handschuh, S., Staab, S., Ciravegna, F.: S-CREAM - Semi-automatic CREAtion of Metadata. In: Proc. EKAW-2002, LNCS, Springer, 2002.
11. Jin, Y., Decker, S., Wiederhold, G.: OntoWebber: Model-Driven Ontology-Based Web Site Management. In: 1st International Semantic Web Working Symposium (SWWS'01), Stanford University, Stanford, CA, July 29-Aug 1, 2001.
12. Kodratoff, Y.: Rating the Interest of Rules Induced from Data and within Texts In: Proc. DEXA 2001, IEEE Computer Science Press, 265-269, 2001.
13. Krátký, M., Pokorný, J., Snášel, V.: Indexing XML Data with UB-trees, in: ADBIS 2002, Research Communications, Bratislava 2002.
14. Krötzch, S., Rösner, D.: Ontology based Extraction of Company Profiles. In: Workshop DBFusion, Karlsruhe 2002.
15. Labský, M., Svátek, V.: Ontology Merging in Context of Web Analysis. In: Workshop DATESO03, TU Ostrava, 2003.
16. Motta, E., Lu, W.: A Library of Components for Classification Problem Solving. In: Proceedings of PKAW 2000: The 2000 Pacific Rim Knowledge Acquisition, Workshop, Sydney, Australia, December 11-13, 2000.
17. Schreiber, G., et al.: Knowledge Engineering and Management. The CommonKADS Methodology. MIT Press, 1999.
18. Svátek, V., Berka, P., Kavalec, M., Kosek, J., Vávra, V.: Discovering company descriptions on the web by multiway analysis. In: New Trends in Intelligent Information Processing and Web Mining (IIPWM'03), Zakopane 2003. Springer-Verlag, 'Advances in Soft Computing' series, 2003.
19. Svátek, V., Kosek, J., Labský, M., Bráza, J., Kavalec, M., Vacura, M., Vávra, V., Snášel, V.: Rainbow - Multiway Semantic Analysis of Websites. In: 2nd International DEXA Workshop on Web Semantics (WebS03), Prague 2003, IEEE Computer Society Press.
20. Šváb, O., Svátek, V., Kavalec, M., Labský, M.: Querying the RDF: Small Case Study in the Bicycle Sale Domain. In: Workshop on Databases, Texts, Specifications and Objects (DATESO'04), online at `http://www.ceur-ws.org/Vol-98`.
21. Tansley, D.S.W., Hayball, C.C.: KBS Analysis and Design. A KADS Developer's Handbook. Prentice Hall 1993.
22. Vacura, M.: Recognition of pornographic WWW documents on the Internet (in Czech), PhD Thesis, University of Economics, Prague, 2003.