ORGANOGRAPHS
Multi-faceted Hierarchical Categorization of Web Documents

Rodrigo Dias Arruda Senra and Claudia Bauzer Medeiros
Institute of Computing, University of Campinas, UNICAMP, Campinas, SP, Brazil

Keywords: Organographs, Hierarchical content categorization, Bookmarking, Classification.

Abstract: The data deluge of information in the Web challenges internauts to organize their references to interesting content in the Web as well as in their private storage space off-line. Having an automatically managed personal index to content acquired from the Web is useful for everybody, but critical to researchers and scholars. In this paper, we discuss concepts and problems related to organizing information through multi-faceted hierarchical categorization. We introduce the *organograph* as a mechanism to specify multiple views of how content is organized. Organographs can help scientists to automatically organize their documents along multiple axes, improving sharing and navigation through themes and concepts according to a particular research objective.

1 INTRODUCTION

The organization, archival and sharing of digital content generated by scientists is important in eScience research - e.g. reports, algorithms and data. Scientists must be able to efficiently organize and disseminate their knowledge, not only within a project, but also with the community at large, where the preferred collaboration environment is the Web. This complicates document management, since each scientist (or group) may use different document standards, storage mechanisms and vocabularies. Such issues are nowadays a prime research topic in *Web Science* – a novel research domain which is concerned with the Web as the primary object of interest.

Continuing our research (Senra and Medeiros, 2009) about data sharing in eScience, this paper concerns with information organization and collaboration on the Web. We are interested in the *organization* of scholarly digital content (documents used by scientists), through automatic hierarchical structuring, multifaceted filtering and sharing. *Hierarchical Structuring* is a pervasive approach towards information organization. It is the cognitve pattern we use to organize everything (e.g., files, messages). Although ubiquitous, we often create our hierarchies manually and in an *ad hoc* fashion. We choose filesystems as the pivot artifact to discuss our approach to hierarchical structuring, considering that digital content is often encapsulated, stored and shared as files.

The issues for filesystems can be transported to other manifestations of the hierarchical structuring pattern. First issue, the membership relation between files and directories, is often static and manually defined by the user on a per-file-instance basis. Second, the hierarchy implicitly defines a content categorization, thus some given file frequently can only be placed in a single point inside the hierarchy. We refer to this issue as the *single category* problem, further exploring it in section 2.2. Above all, the organization of a directory hierarchy is not shared dissociated from the content itself, e.g., people do not exchange hierarchies of empty folders even though there is valuable knowledge encoded in their structuring. We only share directory trees associated with content, because we lack the tools to categorize our content according to a foreign directory hierarchy and vice-versa. We refer to this issue as the *content-recategorization* problem, and discuss it in section 2.4.

The Web aggravated these issues. It is often easier and quicker to find a piece of information in the Internet, than to locate the one already available in our local computer or Intranet. Not only is this an inefficient way of managing content, but it can lead to problems such as needless duplication, loss of quality and provenance mishaps. The information available "off-line" (i.e. in our local computer or Intranet) is not the same information available in the Internet. Although not necessarily as fresh, in some aspects local information can be richer. It has been filtered, possibly transformed, can be sensitive or private, and may

Dias Arruda Senra R. and Bauzer Medeiros C..
ORGANOGRAPHS - Multi-faceted Hierarchical Categorization of Web Documents.
DOI: 10.5220/0003319205830588
In *Proceedings of the 7th International Conference on Web Information Systems and Technologies* (WEBIST-2011), pages 583-588
ISBN: 978-989-8425-51-5

no longer be available on the Web.

This paper proposes an approach towards the automatic organization of documents generated and used by scientists. Our approach enables this organization along multiple axes, thus facilitating its sharing and dissemination. Our hypothesis is that hierarchies can be shared in isolation from their generative collection, and used to organize other (non-generative) collections.

2 APPROACHES TO CONTENT ORGANIZATION

We are interested in *information organization*, and we define it as: "the structuring of information units by a group of agents according to a set of consensual and shared principles to achieve a defined goal". Our definition has five elements. The *agents* represent the users and software artifacts that interact. The *information units* (IUs) represent the granularity at which content is encapsulated and manipulated. The *structure* is defined by intermediate aggregators that partition the set of IUs; these aggregators are often related to content categorization or clusterization. The *principles* are the algorithms and knowledge bases that drive the structuring process. The *goal* is usually to improve how the agents obtain access to the desired IUs. In this paper, we refer to *file* and *document* interchangeably as concrete instances of IUs; similarly, *directories* are instances of hierarchical aggregators to enforce structuring upon the IUs.

2.1 Basic Concepts

Organization as part of Information Retrieval (IR), involves several tasks (Jackson and Moulinier, 2002) – e.g., sorting, summarizing, indexing. We are particularly interested in *Classification* or *Categorization* – the assignment of IUs to a pre-established set of classes or categories defined by the agents. Methods for categorization differ in the form of the classifier, the technique for training, and the representation of the IU, e.g, see (Weigend et al., 1999) on text. While the categorization task assumes an existing set of classes(categories), clustering aims to create or discover a set of classes(clusters).

Another important concept is the notion of ontology ontology (Uschold and Gruninger, 1996): an explicit and rigorous specification of a conceptualization, that organizes some knowledge domain. For many applications, ontologies are mostly hierarchical, containing all the entities and their relations, usually restricted to *is-a* and *part-of*. Ontologies can describe IUs, categories, and relationships between them.

2.2 The *Single Category* Problem

The way we organize digital documents in hierarchies still follows the metaphors of the physical world: archives, drawers and folders. The majority of tools used to organize documents relies on such single taxonomic organization patterns - e.g., file managers or browsers, e-mail clients, or software navigational menus. When the digital space inherited this pattern from the physical world, it also inherited some unnecessary limitations. For instance, many software artifacts restrict a digital document to be placed in a single directory in the filesystem hierarchy, or an email message to be stored in a single folder.

Digital libraries are presented as a means to solve this issue – by allowing multifaceted content organization using links and metadata structures. Other means of circumventing this limitation is the use of copies, hard-links (a.k.a clones) and soft-links (a.k.a symbolic links). While these mechanisms support content multi-categorization, they also introduce new problems. Copies increase storage space, complicate consistency maintenance and may cause redundant processing. Links change the hierarchy traversal from tree-based into a graph-based procedure that often introduces problematic cycles. Furthermore, links introduce a duality: sometimes they are treated transparently, other times not – causing an identity problem between the link and its referred object. Thus, archiving systems and digital libraries still lack flexible content organization support, and suffer from a rigid content structuring.

2.3 Alternatives to Rigid Hierarchical Structuring

In opposition to the hierarchical organization strategy, there are other two approaches to organize content: folksonomies and full-text management engines. The *full-text search* approach abolishes classes and categories altogether. Objects are indexed by their textual content and retrieved by a subset of keywords. The object collection cannot be browsed by topic (or any other property) because it is unstructured, only ranked result sets can be iterated. Furthermore, full-text mechanisms do not support multimedia content, which is increasingly common.

Folksonomies (or social *tagging*) is a recent response to the demands of content organization. In this paradigm, content is annotated and categorized by

multiple labels or tags that form a cloud of unstructured categories, represented by words or short phrases. Any object can be associated with any number of tags, therefore belonging to multiple categories, solving the *single category problem*. Tags can evolve dynamically, being used to browse and retrieve IUs. Therefore, folksonomies preserve classes or emulate categories, but not their inter-relations and structure. In order to minimize the lack of structure, some tagging systems (e.g.Delicious and Connotea) use co-occurrence of tags for navigational purposes. Folksonomies have other limitations (Giannakidou et al., 2008) that restrict their usability, such as: tag validation, uncontrolled vocabularies, spamming and term ambiguity and redundancy.

Summing up, taxonomies suffer from the *single category problem*, full-text search provides no structuring nor categories, and folksonomies have unstructured and unrelated categories.

2.4 The Content-recategorization Problem

As mentioned in section 1, hierarchies still lack the desirable property of dynamic and flexible membership relation between content and category. One of our goals is to allow hierarchical organizations to be shared in isolation from the content they categorize. First we need to distinguish the *generative collection* from a *subordinate collection*. A **generative collection** is the set of IUs from which the agents derived a hierarchical categorization scheme. A subordinate collection is the reciprocal entity, i.e., any set of IUs subject to a specific (generated) classification scheme.

For example, suppose some researcher has a hierarchical collection of articles. The folders represent categories, and the union of all articles are the generative collection. There are two recategorization scenarios. The first is to fit her articles to an external categorization, for instance according to the Library of Congress Subject Headings (LCSH). In this case, each heading from LCSH would become a folder (category), and the researcher's articles would become the subordinate collection to be organized according to this new hierarchy. The second scenario is to fit other documents to her own hierarchy scheme, for example to browse a colleague's collection as if it were organized with her own personal classification scheme.

Each scenario means solving the hierarchical categorization problem. The content-recategorization problem is a variation of hierarchical classification because the content subject to classification is already classified according to a source hierarchy, which could be used to improve the classification process towards the new given target hierarchy. For a full discussion and survey about *hierarchical classification* see (Gordon, 1996). Many classification methods discussed in the literature are not fully automatic, requiring supervised training and user feedback – e.g., naive bayes, support vector machine, or neural networks.

If the categorization process is distributed across different people, or even done by a single person at different times, then categorizations will differ – e.g., (Bonifacio et al., 2000) have shown that community members keep their own perspective on a community repository. The reason why many community repository initiatives fail is due to the fact that a single (though shared) categorization scheme is not accepted or understood by the entire community. This reinforces the idea that single category approaches to classification are doomed to fail.

2.5 Categories for Categorization

We are interested in improving automation for different categorization tasks, which we call: first-time, refactor, shared, and synchronized categorization. *First-time categorization* is when we categorize an assorted digital document collection for the first time.

Refactor categorization is when we already have a categorized collection, but we need to either judge its coherence or refactor it. The quality and suitability of the categorization may vary for different user groups or for the same group doing tasks at different times.

Shared categorization is when we want to re-use the categorization scheme from a different group and apply it to our own content, or do the inverse task – using our own categorization scheme to browse the shared content from others.

Synchronized categorization is a stricter version of shared categorization, when two groups are simultaneously manipulating content and categorization. Each group may adopt a different categorization, and yet they must edit and evolve the same documents. Supposing that one group devised a proper (refactor) categorization for its purposes, then this group should be capable of applying that same categorization to any exchanged content. In other words, in the last two tasks what is sought is a solution to the *content-recategorization* problem.

3 THE ORGANOGRAPH FRAMEWORK

In order to accomplish those four categorization tasks, we present a semantically-grounded organizational

structure that we called *organographs*. An **organograph** is a user parametrization to a hierarchical categorization task, that is editable, persistable and shareable.

Organographs improve access to digital content because they are context-sensitive and built to meet specific user needs, preserving user familiarity with categories and their structure. Multiple organographs can be generated from the same collection given a different parametrization, and a single organograph could be applied to different and unrelated collections. Each generated organograph should be interpreted as a multi-faceted view (Dakka et al., 2007) of the generative collection. A concrete example of organograph is given in figure 1, which is detailed in section 3.3.

3.1 Use Case

Consider the following example: a researcher uses a hierarchy created to gather all material related to his research project. Some documents were generated locally, while others were retrieved from the Web. The document tree contains his publications, unpublished papers, some papers (categorized by subject) he already read, and papers to read. He wants to make this document tree available for his research group, including his own publications, but not his unpublished materials. Moreover, he wants to organize the resulting collection by publication date (year/month) and then by ACM's 1998 Computing Classification System (http://www.acm.org/about/class/1998). Notice that the classifying criteria are either based on attributes that are intrinsic to the IU's content (e.g.: publication date, text subject, annotations), or based on attributes dependent on the user context – being content-independent (eg.: read vs unread, published vs unpublished). In this example, the attributes published/not_published and read/not_read are implicitly given by categories in the original document organization, therefore annotations can be derived automatically using proper attribute extractors (Dakka and Ipeirotis, 2008). In other cases, annotations must be performed manually.

3.2 Methodology

Prior to constructing the target organizing hierarchy, all documents in the source collection are preprocessed and indexed to populate what we call the *attribute-space*. The **attribute-space** is a dictionary-like database whose keys are document IDs and whose values are records with heterogeneous schemas – because different documents might have distinct sets of attributes. It can be implemented on top of a NoSQL database or on top a relational database with a star-join (multi-dimensional) schema.It is built by applying all suitable information extractors available to all documents in the source collection. The cardinality of the attribute-space for a given document depends on the availability of information extractors applicable to the respective document type. Hence, it grows incrementally with the advent of new information extractors.

For instance, the attribute-space for some x.pdf document has the schema: title, author, publication date, document type, word frequency vector, and other user defined keywords. The first four attributes are tagged (by the extractor) with their respective Dublin Core elements counterparts. Furthermore, x.pdf's attribute-space is augmented with attributes retrieved from datasource (the host filesystem), such as: document size, last access/modification date, owner, group, and access permissions. If originated from the Web, the attribute-space could be augmented with information extracted from the host site or the surrounding Web pages. Once the attribute-space is populated, the researcher possesses a *vocabulary of attributes* with which he can write organograph specifications that will guide the materialization of diverse multi-faceted views.

We propose four steps to construct organographs:

Step 1: apply information extraction techniques to the IUs (e.g. documents), factoring out attributes and properties (defining an *attribute space*). Each IU is assigned a unique ID that serves for indexation purposes. **Step 2:** automatically generate categories from a user-given organograph that either explicitly enumerates categories or provides generative rules that define them. User parametrization should anchor categories and their inter-relationships to concepts in ontologies. **Step 3:** run a categorization algorithm specified in the user-given organograph, resulting in the assignment of IU IDs (step 1) to the generated categories (step 2). **Step 4:** use virtualization and links to materialize the emergent categorization using the same structuring metaphor (e.g. directories) of the generative collection.

3.3 Concrete Organograph Example

We provide the organograph described in figure 1 to illustrate what we mean by user parametrization.

Line 1 defines an organograph instance with identification for persistence purposes. Line 2 defines the topmost level (root node). Lines 3-7 define the labels for the first level nodes, consisting of the 4-digit year present in the publication_date attribute. Lines 8-30

```
1: <organograph id="published-pdfs-by-year-month-acmsubject">
2:     <level name="root">
3:         <label>
4:             <datetime format="YYYY">
5:                 <extract attr="publication_date" node="http://purl.org/dc/elements/1.1/date"/>
6:             </datetime>
7:         </label>
8:         <level name="by_year">
9:             <label>
10:                <datetime format="MMM">
11:                    <extract attr="publication_date" node="http://purl.org/dc/elements/1.1/date"/>
12:                </datetime>
13:            </label>
14:            <level name="by_subject">
15:                <topical-classification
16:                  method="naive_bayes"
17:                  classes="http://www.acm.org/about/class/1998/acmccs98-1.2.3.xml">
18:                    <extract attr="text_body">
19:                        <datasource type="directory" uri="file://localhost/home/researcher">
20:                            <includes> <mime_type> application/pdf </mime_type> </includes>
21:                            <excludes> <user_tag> unpublished </user_tag> </excludes>
22:                        </datasource>
23:                    </extract>
24:                </topical-classification>
25:                <label> <extract attr="class_name" /> </label>
26:                <level name="hyperlinks">
27:                    <label> <extract attr="uri" /> </label>
28:                </level>
29:            </level>
30:        </level>
31:    </level>
32: </organograph>
```


Figure 1: Sample Organograph Specification.

define the second nested level, where lines 9-13 define similarly a 3-letter month label for this level's nodes based on the same attribute. Lines 14-29 define the third level, where lines 15-24 apply an algorithm to do topical clusterization according to ACM's CCS ontology. The choice of categorization algorithm is left for the user, in this example we chose "naive bayes" omitting from the example some necessary parameters such as the trainning set used. Line 19 defines the generative datasource collection subject to inclusive (line 20) and exclusive filters (line 21). Finally, lines 26-28 define the inner-most level nodes (tree leaves), consisting of hyperlinks to the IU's (documents) filtered.

The exact syntax and semantics of the domain specific language used to code the organograph lies outside the scope of this paper.

4 RELATED WORK

There are many research initiatives on tagging and text mining, on the Web, whose goal is document sharing. However, to our knowledge, ours is the first work that is geared towards organizing for sharing in a collaborative Web environment, in which each participating scientist can construct a personal organization scheme to allow other researchers to "see" organizations differently.

The CAIMAN system (Lacher and Groh, 2001) facilitates document exchange between geographically dispersed people. Each community member organizes his collection according to his own categorization scheme (ontology), then CAIMAN maps concepts in personal ontologies to concepts in a community ontology. (Bloehdorn et al., 2005) performed text mining experiments in the medical domain in which the ontological structures used were acquired automatically in an unsupervised learning process. They have shown that automatically learned ontologies and manually engineered ones, were both competitive and improved results on text clustering and classification tasks. (Giannakidou et al., 2008) propose an approach for social data clustering which combines semantic, social and content-based information. They devised an unsupervised model for efficient and scalable mining on multimedia social-related data, which leads to the extraction of rich and trustworthy semantics and the improvement of retrieval in a social tagging system. (Du and Chen, 2007) devised a desktop-based personal information management that further exploits a social networking environment for collaborative knowledge creation, integration and sharing, based on the integration of Semantic Web technologies and collaborative tagging. (Chen and Roberts, 2007) presented an architecture

capable of recovering the context of tags and drive emergent semantics, using them to organically build ontologies. The generated semantic hierarchy is used to enforce structure and semantics in collaborative tagging. That approach was adopted in practice in the *Online Open Publishing System*.

Our work also goes toward filling gaps in Web Science research, in the area of designing and developing infrastructures for collaboration on the Web. The term Web Science was first introduced by Berners-Lee. It has since given origin to large international research efforts, including The Web Science Trust (http://webscience.org/) or the Brazilian Institute for Web Science Research (http://webscience.org.br).

Formally, research in Web Science is concerned with the Web as the primary object of interest. In our work, this means among others concentrating on organographs as a means of sharing and exchanging knowledge. Furthermore, once document organizations are shared, the researchers can reuse each other's data – which is the essence of scientific collaboration – without having to concern themselves with establishing standards for document organization.

5 CONCLUSIONS

This paper presented a conceptual framework to automate information organization and support collaborative work on the Web. Our core proposal is the organograph – a persistent and shareable organization that emerges from automatic feature-extraction, classification and clustering. Though our discussion was centered in document sharing and reuse, our end-users are scientists that work cooperatively in some eScience domain. In such a context, documents refer not only to scientific papers and reports, but also data files containing experimental data, or images. Under this perspective, our proposal can be extended to other domains in which cooperation on the Web is required.

At the same time, we need to concern ourselves with the Web Science issue of visibility. It is not enough to share organographs, if we also want the documents to be visible beyond a research group. Indeed, the validation of scientific experiments requires reproducibility – and this means that documents associated with an eScience project must all, at the end, become available. This means that we must also consider some sort of Publication Directory, in which a group's (or a project's) organographs can be accessed by all interested in accessing the main results of that group. This kind of solution is part of our ongoing research. We will validate this concept using real applications that run in the Web and that have been implemented by our research group, in distinct scientific domains.

ACKNOWLEDGEMENTS

This work was supported by Fapesp, CNPq, CAPES and INCT in Web Science (CNPq 557.128/2009-9).

REFERENCES

Bloehdorn, S., Cimiano, P., and Hotho, A. (2005). Learning ontologies to improve text clustering and classification. In *From Data and Information Analysis to Knowledge Engineering: Proceedings of the 29th Annual Conference of the German Classification Society*.

Bonifacio, M., Bouquet, P., and Manzardo, A. (2000). A distributed intelligence paradigm for knowledge management. In *AAAI Spring Symposium Series 2000 on Bringing Knowledge to Business Processes*.

Chen, L. and Roberts, C. (2007). Semantic tagging for large-scale content management. In *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*.

Dakka, W. and Ipeirotis, P. G. (2008). Automatic extraction of useful facet hierarchies from text databases. In *ICDE*, pages 466–475.

Dakka, W., Ipeirotis, P. G., and Wood, K. R. (2007). Faceted browsing over large databases of text-annotated objects. In *ICDE*, pages 1489–1490.

Du, Y. and Chen, L. (2007). Using personalized knowledge portal for information and knowledge integration and sharing. In *SKG '07: Proceedings of the Third International Conference on Semantics, Knowledge and Grid*.

Giannakidou, E., Kompatsiaris, I., and Vakali, A. (2008). Semsoc: Semantic, social and content-based clustering in multimedia collaborative tagging systems. In *ICSC '08: Proceedings of the 2008 IEEE International Conference on Semantic Computing*.

Gordon, A. (1996). Hierarchical classification. *Clustering and classification*.

Jackson, P. and Moulinier, I. (2002). *Natural language processing for online applications: text retrieval, extraction, and categorization*. John Benjamins Publishing Company.

Lacher, M. and Groh, G. (2001). Facilitating the exchange of explicit knowledge through ontology mappings. In *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference*.

Senra, R. D. A. and Medeiros, C. B. (2009). SciFrame: a conceptual framework to describe data sharing in e-Science. *SBBD. III e-Science Workshop*.

Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The Knowledge Engineering Review*, 11(02).

Weigend, A. S., Wiener, E. D., and Pedersen, J. O. (1999). Exploiting hierarchy in text categorization. *Inf. Retr.*, 1(3).