Access and Exchange of Hierarchically Structured Resources on the Web with the NESTOR Framework

Maristella Agosti, Nicola Ferro, Gianmaria Silvello
Department of Information Engineering
University of Padua, Italy
{agosti, ferro, silvello}@dei.unipd.it

***Abstract*—The paper addresses the problem of representing, managing and exchanging hierarchically structured data in the context of *Digital Library (DL)* systems in order to enhance the access and exchange DL resources on the Web.**

We propose the *NEsted SeTs for Object hieRarchies (NESTOR)* framework, which relies on two set data models — the "Nested Set Model (NS-M)" and the "Inverse Nested Set Model (INS-M)" — to enable the representation of hierarchical data structures by means of a proper organization of nested sets. In particular, we show how NESTOR can be effectively exploited to enhance *Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)* for better access and exchange of hierarchical resources on the Web.

***Keywords*-hierarchical structures; set data models; OAI-PMH; data access and exchange**

I. Introduction

The role of *Digital Library (DL)* in collecting, managing and preserving cultural heritage resources is increasingly important in several contexts. DL are not merely the digital counterpart of traditional libraries, rather they can be seen as tools for managing information resources of different kinds of organizations: from libraries, and museums to archives. In these different contexts, DL systems permit the management of wide and different corpora of resources which range from books and archival documents to multimedia resources. The different types of resources are often represented and managed with the use of metadata which contain a *Uniform Resource Identifier (URI)* for the resource on the Web or of a Web page from which the resource may be obtained. In addition to the management of metadata, DL systems offer advanced services, such as: multimedia and multilingual access, specialized services for e-learning and e-government or mash-up of resources into new information objects.

DL represent significant institutional investments, yet their resources may remain hidden in the Deep Web: even though they are accessible on the Web, they are often poorly integrated with mainstream Web applications and may be overlooked by major search engines [1], unless search engines make special accomodations for their protocol and access schemes. Moreover, DL systems have to manage and share resources differing in the media and in the structures in which they are organized. As a consequence DL systems have to face the interoperability issues related to the heterogeneous resources they manage and, possibly, exchange [2].

In the context of DL, the *de-facto* standards for metadata exchange are the *Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)* [3] and the *eXtensible Markup Language (XML)*[1]. The main reason is the flexibility of both the protocol and the markup language that support interoperability between DL and enhance their visibility on the Web. OAI-PMH promotes interoperability allowing metadata to be harvested between different repositories in a straightforward fashion, in order to create aggregated metadata collections and to enable the creation of advanced services on them. Furthermore, OAI-PMH permits us to enhance the presence of DL resources on the Web and to increase their visibility through search engines [4]. Although the interoperability and accessibility to DL resources is well-supported by the OAI-PMH, the effectiveness of this protocol in exposing and sharing DL resources on the Web can be limited by the hierarchical structure of these resources. Indeed, in DL resources are often organized in hierarchies to help in representing, managing or browsing them, like, for example, documents in an "archive" are organized in a hierarchy divided into fonds, sub-fonds, series, sub-series and so on. Also the internal structure of an object can be hierarchical, like a book organized in chapters, sections and subsections or a Web page composed of nested elements such as body, titles, subtitles, paragraphs and subparagraphs. XML is an important tool extensively adopted to represent digital objects such as metadata, text documents, and multimedia contents, which makes use of an intrinsically hierarchical structure.

Often these hierarchies are managed as a unique digital object and thus have a unique URI associated to them; for instance, a physical archive is usually described by a single metadata with a big and deep hierarchical structure, where every node of the hierarchy can contain a set of unique resources. These resources are embedded inside a hierarchical structure that permits us to maintain all the meaningful relationships with the other resources, but at the same time they are hardly reachable from the outside

[1] http://www.w3.org/XML/

978-0-7695-3801-3/09 $26.00 © 2009 IEEE
DOI 10.1109/WI-IAT.2009.110

without accessing the whole hierarchy. As a consequence these resources are exposed and shared by the DL throughout OAI-PMH as "monolithic" units thus providing only one access point to a whole hierarchy.

The approach we present in this paper originates from the foundational idea that a set data model that supports an opportune set organization can maintain all the features of a tree data structure; we have developed a framework to be used to reconstruct the aggregations or the whole hierarchy in which the resources are organized maintaining the integrity of the structure and the relationships between the resources. The proposed framework named *NEsted SeTs for Object hieRarchies (NESTOR)* [5] permits variable granularity access and exchange to digital resources permitting each resource to be associated with a unique URI. The NESTOR framework operates on the basis of two set data models opportunely defined: the "Nested Set Model (NS-M)" and the "Inverse Nested Set Model (INS-M)". The two models are defined in the context of the ZFC (Zermelo-Fraenkel with the axiom of Choice) axiomatic set theory, exploiting the advantages of the use of sets in place of a tree structure.

New relevant functionalities that can be exploited by substituting a set organization with a hierarchical one are: flexibility, rapid selection and isolation of easily specified subsets of data, and extraction of only those data necessary to satisfy specific needs. The proposed models can work in conjunction with the OAI-PMH; the extension of OAI-PMH permits the exchange of data belonging to a hierarchy with a variable granularity without losing the relationships with the other data in the hierarchy.

The paper is organized as follows: Section II defines the two set data models and presents relevant mapping functions between data structures. Section III describes the functioning of OAI-PMH and shows how it can be used in conjunction with the NESTOR framework. Finally, section IV draws some conclusions.

II. The Set Data Models

We propose two set data models called *Nested Set Model* (NS-M) and *Inverse Nested Set Model* (INS-M) based on an organization of nested sets. The most intuitive way to understand how these models work is to relate them to the well-know tree data structure. Thus, we informally present the two data models by means of examples of mapping between them and a sample tree.

The first model we present is the **Nested Set Model** (NS-M). The intuitive graphic representation of a tree as an organization of nested sets was used in [6] to show different ways to represent tree data structure. An organization of sets in the NS-M is a collection of sets in which any pair of sets is either disjoint or one contains the other. In Figure 1 we can see how a sample tree is mapped into an organization of nested sets based on the NS-M.

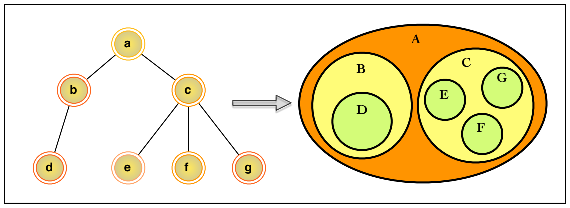

Figure 1. The mapping between a tree data structure and the NS-M.

From Figure 1 we can see that each node of the tree is mapped into a set, where child nodes become *proper subsets* of the set created from the parent node. Every set is subset of at least of one set; the set corresponding to the tree root is the only set without any supersets and every set in the hierarchy is subset of the root set. The external nodes are sets with no subsets. The tree structure is maintained thanks to the nested organization and the relationships between the sets are expressed by the set inclusion order. Even the disjunction between two sets brings information; indeed, the disjunction of two sets means that these belong to two different branches of the same tree.

The second data model is the **Inverse Nested Set Model** (INS-M). We can say that a tree is mapped into the INS-M transforming each node into a set, where each parent node becomes a subset of the sets created from its children. The set created from the tree's root is the only set with no subsets and the root set is a proper subset of all the sets in the hierarchy. The leaves are the sets with no supersets and they are sets containing all the sets created from the nodes composing tree path from a leaf to the root. An important aspect of INS-M is that the intersection of every couple of sets obtained from two nodes is always a set representing a node in the tree. The intersection of all the sets in the INS-M is the set mapped from the root of the tree.

Differently from the NS-M, the representation of the INS-M by means of the Euler-Venn diagrams is not very expressive and can be confusing for the reader. We can represent in a straightforward way the INS-M by means of the "*DocBall representation*". The DocBall representation is used in [7] to depict the structural components of the documents and can be considered as the representation of a tree structure; indeed, it has been used also to draw Web pages. We exploit the DocBall ability to show the structure of an object and to represent the "*inclusion order of one or more elements in another one*" [7]. The DocBall is composed of a set of circular sectors arranged in concentric rings as shown in Figure 2. In a DocBall each ring represents a level of the hierarchy with the center (level 0) representing the root. In a ring, the circular sectors represent the nodes in the corresponding level. We use the DocBall to represent the INS-M, thus for us each circular sector corresponds to a set.

In Figure 2 we can see the INS-M mapping of a sample

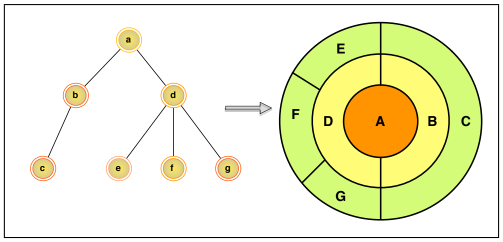

Figure 2. The mapping between a tree and INS-M by means of the DocBall representation.

tree by means of the DocBall representation. The root "a" of the tree is mapped into the set "A" represented by the inner ring at level 0 of the DocBall; at level 1 we find the children of the root and so on. With this representation a subset is presented in a ring within than the set including it. Indeed, we can see that the set A is included by all the other sets. If the intersection of two or more sets is empty then these sets have no common circular sector in the inner rings of the DocBall; in the INS-M this is not possible because the set representing the root (A) is common to all the sets in the INS-M.

It is worthwhile for the rest of the work to define some basic concepts of set theory: the family of subsets and the subfamily of subsets. However, we assume the reader is confident with the basic concepts of ZFC axiomatic set theory, which we cannot extensively treat here for space reasons.

Definition 1: Let A be a set, I a non-empty set and $\mathcal{C}$ a collection of subsets of A. Then a bijective function $\mathcal{A} : I \longrightarrow \mathcal{C}$ is a **family** of subsets of A. We call I the **index** set and we say that the collection $\mathcal{C}$ is **indexed** by I.

We use the following notation $\{A_i\}_{i\in I}$ to indicate the family $\mathcal{A}$; the notation $A_i \in \{A_i\}_{i\in I}$ means that $\exists\ i \in I \mid (\mathcal{A}(i) = A_i)$. We call **subfamily** of $\{A_i\}_{i\in I}$ the **restriction** of $\mathcal{A}$ to $J \subseteq I$ and we denote this with $\{B_i\}_{j\in J} \subseteq \{A_i\}_{i\in I}$.

Definition 2: Let A be a set and let $\{A_i\}_{i\in I}$ be a family. Then $\{A_i\}_{i\in I}$ is a **Nested Set** family if:

$$A \in \{A_i\}_{i\in I}, \tag{II.1}$$

$$\emptyset \notin \{A_i\}_{i\in I}, \tag{II.2}$$

$$\forall A_h, A_k \in \{A_i\}_{i\in I}, h \neq k \mid (A_h \cap A_k \neq \emptyset) \Rightarrow A_h \subset A_k \vee A_k \subset A_h. \tag{II.3}$$

Thus, we define a Nested Set family (NS-F) as a family where three conditions must hold. The first condition (II.1) states that set A which contains all the sets in the family must belong to the NS-F. The second condition states that the empty-set does not belong to the NS-F and the last condition (II.3) states that the intersection of every couple of distinct sets in the NS-F is not the empty-set only if one set is a proper subset of the other one.

In the same way we can define the Inverse Nested Set Model (INS-M):

Definition 3: Let A be a set and let $\{A_i\}_{i\in I}$ be a family. Then $\{A_i\}_{i\in I}$ is an **Inverse Nested Set** family if:

$$\emptyset \notin \{A_i\}_{i\in I}, \tag{II.4}$$

$$\forall \{B_j\}_{j\in J} \subseteq \{A_i\}_{i\in I} \Rightarrow \bigcap_{j\in J} B_j \in \{A_i\}_{i\in I}. \tag{II.5}$$

Thus, we define an Inverse Nested Set family (INS-F) as a family where two conditions must hold. The first condition (II.4) states that the empty-set does not belong to the INS-F. The second condition states that the intersection of every subfamily of the INS-F belongs to the INS-F itself.

III. How to Exploit the NESTOR Framework in Conjunction with OAI-PMH

The defined set data models can be exploited to improve the data exchange between DL systems in a distributed environment. Our aim is to show how NESTOR enables OAI-PMH to cope with complex hierarchical structured objects without any losses in its basic features that are: flexibility, adaptability and non-invasivity. In order to explain how NESTOR can be used in conjunction with OAI-PMH it is worthwhile to describe a native function of the protocol: the selective harvesting. The selective harvesting is based on the concept of *OAI-set*, which enables logical data partitioning by defining groups of records. Selective harvesting is the procedure that permits the harvesting only of metadata owned by a specified OAI-set. In OAI-PMH a set is defined by three components: `setSpec` which is mandatory and a unique identifier for the set within the repository, `setName` which is a mandatory short human-readable string naming the set, and `setDesc` which may hold community-specific XML-encoded data about the set.

OAI-set organization may be hierarchical, where hierarchy is expressed in the `setSpec` field by the use of a colon [:] separated list indicating the path from the root of the set hierarchy to the respective node. For example if we define an OAI-set whose `setSpec` is *"A"*, its subset "B" would have *"A:B"* as `setSpec`. In this case "B" is a proper subset of "A": $B \subset A$. When a repository defines a set organization it must *include set membership information in the headers of the records* returned to the harvester requests. Harvesting from a set which has sub-sets will cause the repository to return the records in the specified set and recursively to return the records from all the sub-sets. In our example, if we harvest set A, we also obtain the records in sub-set B.

In OAI-PMH it is possible to define an OAI-set organization based on the NS-M or INS-M. This means that we can treat the OAI-sets as a Nested Set Family (NS-F) or as an Inverse Nested Set Family (INS-F). The inclusion order between the OAI-sets is given by its identifier which is a `<setspec>` value. In the following we describe how it is possible to create a Nested Set family of OAI-Set and afterward how the same thing can be done with an Inverse Nested Set family.

Let $\mathcal{O}$ be a Nested Set family and let I be the set of the <setspec> values where $i \in I = \{s_0 : s_1 : \dots : s_j\}$ means that $\exists\ O_j \in \{O_i\}_{i \in I} \mid O_j \subset \dots \subset O_1 \subset O_0$. Every $O_i \in \{O_i\}_{i \in I}$ is an OAI-set uniquely identified by a <setspec> value in I. The <setspec> values for the $O_k \in \{O_i\}_{i \in I}$ are settled in such a way to maintain the inclusion order between the sets. If an O_k has no superset its setspec value is composed only by a single value (<setspec>s_k</setspec>). Instead if a set O_h has supersets, e.g. O_a and O_b where $O_b \subset O_a$, its setspec value must be the combination of the name of its supersets and itself separated by the colon [:] (e.g. <setspec>$s_a : s_b : s_h$</setspec>). Furthermore, let $R = \{r_0, \dots, r_n\}$ be a set of records, then each $r_i \in O_j$ must contain the setspec of O_j in its header.

Throughout $\{O_i\}_{i \in I}$ it is possible to represent a hierarchical data structure, such as a tree, in OAI-PMH providing a granularity access to the items in the hierarchy and at the same time enabling the exchange of a single part of the hierarchy with the possibility of reconstructing the whole hierarchy whenever it is necessary. The NS-M fosters the reconstruction of the lower levels of the hierarchy; for instance, if a Service Provider harvests the subset representing a chapter it recursively obtains all the subsets of the chapter, which in this example are sections and subsections.

In the same way we can apply the INS-M to OAI-PMH; let $\mathcal{U}$ be an Inverse Nested Set family and let J be the set of the <setspec> values where $j \in J = \{s_0 : s_1 : \dots : s_k\}$ means that $\exists\ U_k \in \{U_j\}_{j \in J} = U_k \subset \dots \subset U_1 \subset U_0$. In $\{U_j\}_{j \in J}$ unlike in $\{O_i\}_{i \in I}$ the following case may happen: Let $U_i, U_k, U_w \in \{U_j\}_{j \in J}$ then it is possible that $U_w \subset U_i$ and $U_w \subset U_k$ but either $U_i \nsubseteq U_k$ and $U_k \nsubseteq U_i$. If we consider $\{U_j\}_{j \in J}$ is composed only of U_i, U_k and U_w, the identifier of U_i is <setspec>s_i</setspec> and the identifier of U_k is <setspec>s_k</setspec>. Instead, the identifier of U_w must be <setspec>$s_j : s_w$</setspec> and <setspec>$s_k : s_w$</setspec> at the same time; this means that in $\{U_j\}_{j \in J}$ there are two distinct OAI-sets, one identified by <setspec>$s_j : s_w$</setspec> and the other identified by <setspec>$s_k : s_w$</setspec>. This is due to the fact that the intersection between OAI-sets in OAI-PMH is not defined set-theoretically; indeed, the only way to get an intersection of two OAI-sets is by enumerating the records. This means that we can know if an OAI-record belongs to two or more sets just by seeing whether there are two or more <setspec> entries in the header of the record. In this case the records belonging to U_w will contain two <setspec> entries in their header: <setspec>$s_j : s_w$</setspec> and <setspec>$s_k : s_w$</setspec>; note that only the <setspec> value is duplicated and not the records themselves.

With this view of OAI-PMH we can set a hierarchical structure of items as a well-defined nested set organization that maintains the relationships between the items just as a tree data structure does and moreover we can exploit the flexibility of the sets exchanging a specific subset while maintaining the integrity of the data. Throughout the NS-M and INS-M it is possible to handle hierarchical structures in OAI-PMH simply by exploiting the inner functionalities of the protocol; indeed, no change of OAI-PMH is required to cope with the presented set data models.

IV. Conclusions

The paper has introduced the two set data models — named "Nested Set Model (NS-M)" and "Inverse Nested Set Model (INS-M)" — which permit the organization of nested sets that enable the representation of hierarchical data structures, and the NESTOR framework, which is based on the two set data models, and which can be used in conjunction with the *Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)* adding new functionalities to it.

Acknowledgments

The work reported in this paper has been supported by a grant from the Italian Veneto Region. The study is also partially supported by the TELplus Targeted Project for Digital Libraries, as part of the eContentplus Program of the European Commission (Contract ECP-2006-DILI- 510003).

References

[1] H. Van de Sompel, C. Lagoze, M. Nelson, S. Warner, R. Sanderson, and P. Johnston, "Adding eScience Assets on the Data Web," in *Proc. of the Linked Data on the Web Workshop of the WWW2009 Conf.* CEUR Workshop Proceedings, 2009.

[2] N. Ferro and G. Silvello, "A Methodology for Sharing Archival Descriptive Metadata in a Distributed Environment," in *Proc. 12th European Conf. on Research and Adv. Tech. for Digital Libraries (ECDL 2008).* LNCS 5173, Springer, Heidelberg, Germany, 2008, pp. 268–279.

[3] H. Van de Sompel, C. Lagoze, M. Nelson, and S. Warner, "The Open Archives Initiative Protocol for Metadata Harvesting (2nd ed.)," Open Archive Initiative, p. 24, Tech. Rep., 2003.

[4] F. McCown, L. Xiaoming, M. L. Nelson, and M. Zubair, "Search Engine Coverage of the OAI-PMH Corpus," *IEEE Internet Computing*, vol. 10, no. 2, pp. 66–73, 2006.

[5] N. Ferro and G. Silvello, "The NESTOR Framework: How to Handle Hierarchical Data Structures," in *Proc. 13th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2009).* Springer, Heidelberg, Germany, 2009, in print.

[6] D. E. Knuth, *The Art of Computer Programming, third edition.* Addison Wesley, 1997, vol. 1.

[7] J. Vegas, F. Crestani, and P. de la Fuente, "Context Representation for Web Search Results," *Journal of Information Science*, vol. 33, no. 1, pp. 77–94, 2007.