# The NESTOR Model: Properties and Applications in the Context of Digital Archives

Nicola Ferro and Gianmaria Silvello

Department of Information Engineering, University of Padua, Italy
{ferro, silvello}@dei.unipd.it

**Abstract.** We present and describe the NEsted SeTs for Object hieRarchies (NESTOR) Model that allows us to model, manage, access and exchange hierarchically structured resources. The NESTOR Model is based on two set data models which can be put in relation with the tree data structure. We present these models highlighting their properties and the relationships with the tree.
We present a concrete use case based on archives that are fundamental and challenging entities in the digital libraries panorama. Within the archives we show how an archive can be represented through set data models, and how their properties can be used in this context; in particular, we focus of the problem of finding the lowest common ancestor.

## 1 Introduction

In Digital Libraries objects are often organized in hierarchies to help in representing, managing or browsing them. For instance, the documents in an archive are organized in a hierarchy divided into fonds, sub-fonds, series, sub-series and so on. Representing, managing, preserving and sharing efficiently and effectively the hierarchical structures is a key point for the development and the consolidation of Digital Library technology and services.

In this paper we provide a further analysis of the *NEsted SeTs for Object hieRarchies (NESTOR)* model which defines two set data models that we call: the "Nested Set Model (NS-M)" and the "Inverse Nested Set Model (INS-M)" [7]. These models are defined in the context of the ZFC (Zermelo-Fraenkel with the axiom of Choice) axiomatic set theory [3], exploiting the advantages of the use of sets in place of a tree structure. The foundational idea behind these set data models is that an opportune set organization can maintain all the features of a tree data structure with the addition of some new relevant functionalities. We define these functionalities in terms of flexibility of the model, rapid selection and isolation of easily specified subsets of data and extraction of only those data necessary to satisfy specific needs.

In this work we focus on the operations that we can perform in these set data models and we provide a use case in order to clarify the possible applications of the models as well as of their operations. In particular, we focus on the archives because they are one of the main organizations of interest for Digital Libraries;

they are a meaningful example of the need to support document management and access. The fundamental characteristic of archives resides in their internal hierarchical organization that constitutes a challenge for their representation, managing and, exchange as well as for their manipulation and querying.

In the presentation of the NESTOR Model, we concentrate on the INS-M and on how it can be used to model an archive and its resources. Furthermore, we analyze how we can define the lowest common ancestor in a hierarchy modeled by means of the INS-M. We highlight the problem of finding of the lowest common ancestor because it is an intrinsically beautiful and widely studied problem as well as a frequently performed operation in the archival context.

This paper is organized as follows: Section 2 introduces the background concepts on which this work is based; we introduce the basic set-theoretical concepts we are going to exploit and a brief definition of the tree data structure. Furthermore, we describe the basic principles of the archival practice and the standards to model and describe an archive in a digital environment. Section 3 presents the formal definition of the INS-M and proves the theorems defining how we can map a tree in the INS-M and vice versa. Moreover, we introduce a proposition showing the correlation between some operations in the tree and in the INS-M. Section 4 details how it is possible to define the lowest common ancestor in the INS-M. Section 5 presents a use case based on the archives where the INS-M properties are exploited; in particular, we explain how the INS-M has been adopted and exploited in the context of the SIAR (*Sistema Informativo Archivistico Regionale*) project. Lastly, in Section 6 we draw some final remarks.

## 2 Background

### 2.1 Set Theory: Collections of Subsets

We assume the reader to be confident with the basics of set theory that we cannot extensively treat here for space reasons [9]. The formal basis of this work is based on the concept of "Collection of subsets" that we introduce starting from the well-know concept of power set.

Let $E$ be a set, we denote with $\mathcal{P}(E)$ the set containing all and only the subsets of $E$, that is, a set $A$ belongs to $\mathcal{P}(E)$ if and only if it belongs to $E$. $\mathcal{P}(E)$ is called the **power set** of $E$. We understand that if $E$ is a set, then there exists a set (collection) $\mathcal{P}$ such that if $A \subseteq E$, then $A \in \mathcal{P}$. The power set of a set $E$ contains all the subsets of $E$, thus any collection of sets $\mathcal{C}$ composed by some subsets of $E$ is a subcollection of the power set $\mathcal{P}(E)$, that is: $\mathcal{C}(E) \subseteq \mathcal{P}(E)$. Let us consider a set $E$ and a collection of subsets $\mathcal{C}(E)$, we say that $\{H, K\} \in \mathcal{C}$ are incomparable, say $H||K$, is $H \nsubseteq K \wedge K \nsubseteq H$.

The following definition points out an important construction that we are going to exploit extensively in this work which is the *collection of proper subsets/supersets*.

**Definition 1** *Let $\mathcal{C}$ be a collection of sets and $A \in \mathcal{C}$ be a set. We define $\mathcal{S}^+(A) = \{B \in \mathcal{C} \ : \ B \subset A\}$ to be the* ***collection of proper subsets*** *of $A$*

*in $\mathcal{C}$. We define $\mathcal{S}^-(A) = \{B \in \mathcal{C} \ : \ A \subset B\}$ to be the* ***collection of proper supersets*** *of $A$ in $\mathcal{C}$.*

It is worthwhile for the rest of the work to introduce a formal definition of "family of subsets".

**Definition 2** *Let $A$ be a set, $I$ a non-empty set and $\mathcal{C}$ a collection of sets of $A$. Then a function $\mathcal{A} : I \to \mathcal{C}$ is defined to be a* ***family*** *of subsets of $A$. We call $I$ the* ***index*** *set and we say that the collection $\mathcal{C}$ is* ***indexed*** *by $I$.*

It is possible to use the extended notation $\{A_i\}_{i \in I}$ to indicate the family of subsets $\mathcal{A} : I \to \mathcal{C}$. The notation $A_i \in \{A_i\}_{i \in I}$ means that $\exists\ i \in I \mid \mathcal{A}(i) = A_i$. In the rest of the work to indicate a family of subsets $\mathcal{A} : I \to \mathcal{C}$ we will use the shorthand notation $\{\mathcal{A}_I\}$.

A frequently used concept is the one of **subfamily**: We indicate with $\{\mathcal{A}_J\}$ the subfamily of $\{\mathcal{A}_I\}$ defined as its **restriction** to $J \subseteq I$ and we say that $\{\mathcal{A}_J\} \subseteq \{\mathcal{A}_I\}$.

### 2.2 The Tree Data Structure

The most common and diffuse way to represent a hierarchy is the tree data structure, which is one of the most important non-linear data structures in computer science [11]. We define a tree as $T(V, E)$ where $V$ is the set of nodes and $E$ the set of edges connecting the nodes. $V$ is composed by $n$ nodes $V = \{v_1, \ldots, v_n\}$ and $E$ is composed by $n-1$ edges. If $v_i, v_j \in V$ and if $e_{ij} \in E$ then $e_{ij}$ is the edge connecting $v_i$ to $v_j$, thus $v_i$ is the parent of $v_j$. In this context it is convenient to talk about *inbound edges* and *outbound edges* of a node.

**Definition 3** *Let $T = (V, E)$ be a rooted tree and $v_i \in V$ be a node of the tree, then we define its:*

**Inbound set** *to be* $E^-(v_i) = \{v_j \in V \mid e_{j,i} \in E\}$.
**Outbound set** *to be* $E^+(v_i) = \{v_j \in V \mid e_{i,j} \in E\}$.
**Inbound degree** *to be* $|E^-(v_i)|$[1].
**Outbound degree** *to be* $|E^+(v_i)|$.

We define with $\Gamma^+(v_i)$ the set of **all the descendants** of $v_i$ in $V$ (including $v_i$ itself); vice versa $\Gamma^-(v_i)$ is the set of **all the ancestors** of $v_i$ in $V$ (including $v_i$ ifself). We shall use the set $\Gamma$ in the following of this work, so it is worth underlining a couple of recurrent cases. Let $v_r \in V$ be the root of a tree $T(V, E)$ then $\Gamma^-(v_r) = \{v_r\}$ and $\Gamma^+(v_r) = V$.

Furthermore, by means of this newly described notation, we can formally define the important concept of **lowest common ancestor**. The lowest common ancestor of nodes $v_j$ and $v_k$ in a tree is the ancestor of $v_j$ and $v_k$ that is located farthest from the root [2].

[1] For all nodes $v_i \in V$ such that $v_i \neq v_r$ where $v_r$ is the root, $|E^-(v_i)| = 1$.

**Definition 4** *Let $T(V, E)$ be a tree and $v_j, v_k \in V$ be two vertices. Then we define $v_t$ to be the* ***lowest common ancestor*** *of $v_j$ and $v_k$ ($\mathsf{lca}(v_j, v_k) = v_t$) if:*

$$v_t \in \Gamma^-(v_j) \cap \Gamma^-(v_k), \text{ and} \tag{2.1}$$

$$\nexists v_w \in V, w \neq t \mid (v_w \in \Gamma^-(v_j) \cap \Gamma^-(v_k)) \wedge (v_w \in \Gamma^+(v_t)) \tag{2.2}$$

The first condition imposes that $v_t = \mathsf{lca}(v_j, v_k)$ must be a common ancestor of $v_j$ and $v_k$; the second condition says that cannot exist a vertex that is not $v_t$ which is nearer than $v_t$ to both $v_j$ and $v_k$.

### 2.3 Archives

An archive represents the trace of the activities of a physical or juridical person in the course of their business which is preserved because of their continued value. Archives have to keep the context in which their records have been created and the network of relationships between them in order to preserve their informative content and provide understandable and useful information over time [8].

The context and the relationships between the documents are preserved thanks to the hierarchical organization of the documents inside the archive. Indeed, an archive is divided by fonds and then by sub-fonds and then by series and then by sub-series and so on – see Figure 1a for an example; at every level we can find documents belonging to a particular division of the archive or documents describing the nature of the considered level of the archive (e.g. a fond, a sub-fonds, etc.). The union of all these documents, the relationships and the context information permits the full informational power of the archival documents to be maintained. The archival documents are analyzed, organized, and recorded by means of the *archival descriptions* [12] that have to reflect the peculiarities of the archive [4].

### 2.4 Digital Archives and the NESTOR Model.

In the digital environment archival descriptions are encoded by the use of metadata; these need to be able to express and maintain the structure of the descriptions and their relationships [8].

The standard format of metadata for representing the hierarchical structure of the archive is the *Encoded Archival Description (EAD)* [13], which reflects the archival structure and holds relations between entities in an archive. In addition, EAD has a flexible structure, encourages archivists to use collective and multilevel description, and has a broad applicability. On the other hand, the EAD permissive data model may undermine the very interoperability it is intended to foster and it must meet stringent best practice guidelines to be shareable and searchable [15]. Furthermore, an archive is described by means of a unique EAD file and this may be problematic when we need to access and exchange archival metadata with a variable granularity [5] by means of DL standard technologies like the *Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)*[2] [16].

[2] `http://www.openarchives.org/`

Several other modeling methodologies and metadata formats have been developed. Indeed, we may consider the "Tree-based Metadata" approach in which archives are described by a collection of lightweight metadata – e.g. Dublin Core Application Profiles[3] – one for each archival resource, connected one to the other by means of links to a third-party file – e.g. an external XML file – which maintains the archival structure [14]; alternative instantiations of this approach maintain the archival structure by means of an opportunely designed relational database [15]. Another possibility is to represent the archival structure by means of a collection of nested sets where each set represents an archival division and contains the metadata describing the resources belonging to that division [5]. This modeling methodology is based on the NESTOR Model which relies on two set data models called *Nested Set Model* (NS-M) and *Inverse Nested Set Model* (INS-M) [1]. Both these set data models, formally defined in the context of axiomatic set theory [10], can be used to model an archive by means of nested sets [7]. An extensive analysis of the NESTOR Model and its applications in the context of DL and archives can be found in [1]; in this paper we exploit the functionalities of the INS-M and thus we focus our presentation on this model.

The most intuitive way of understanding how the INS-M works is to see how a sample tree is mapped into an organization of nested sets based on the 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. We can represent in a straightforward way the INS-M by means of the "*DocBall representation*" [17] – see Figure 1b. It is worthwhile to understand how the DocBall is used because the graphical tool we are going to present is based on this idea. The DocBall is composed of a set of circular sectors arranged in concentric rings; each ring represents a level of the hierarchy with the center 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; for instance, referring to Figure 1b, it is possible to say that section "Series C" is a direct superset of section "Sub-Fonds B".

It has been shown [7] that an archive can be modeled by means of the INS-M and than instantiated in such a way that allows the use of the OAI-PMH architecture to enable a variable granularity access and exchange of the archival metadata. Furthermore, in [5] it has been described a methodology to map an EAD file into the NESTOR Model preserving the full informative power of the metadata. Mapping an EAD file into the NESTOR Model means that we dispose of a methodology that maps the EAD structure into the INS-M and a collection of lightweight metadata containing the content information retained by EAD. In this way the INS-M preserves the archival structure and the metadata belonging to its sets preserve the content of archival descriptions [5]. In the same way, this

[3] `http://www.dublincore.org/`

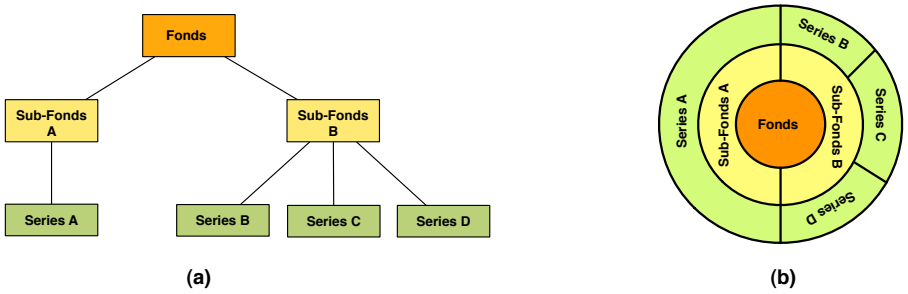


**Fig. 1.** The structure of a sample archive represented by: (a) a tree; (b) a Doc-Ball.

methodology is adopted with the "Tree-based metadata" approach, where the structure retained by an external XML file or by a relational database is mapped into the INS-M [1]. Thus, the INS-M can be used as a means to map archival metadata created by different systems in a common environment [5] as well as it can be adopted to model and describe an archive from scratch [7,1].

## 3 The Inverse Set Data Model and the Tree Data Structure

Now, we can define the Inverse Nested Set Model (INS-M):

**Definition 5** *Let $A$ be a set and let $\mathcal{C}$ be a collection. Then, $\mathcal{C}$ is an* ***Inverse Nested Set Collection*** *if:*

$$\exists! B \in \mathcal{C} \mid \forall K \in \mathcal{C}, B \subseteq K, \tag{3.1}$$

$$\forall H, K, L \in \mathcal{C} \mid H \subseteq K \wedge H||L \Rightarrow L \cap K = H \cap L. \tag{3.2}$$

Thus, we define an Inverse Nested Set Collection (INS-C) as a collection of subsets where two conditions must hold. The first condition (3.1) states that $\mathcal{C}$ must contain a bottom set, call it $B$, such that it is the common subset of all the sets in the collection. The second condition (3.2) states that if we consider two sets $K$ and $H$ such that $H$ is a subset of $K$, then it cannot exist a set $L$ incomparable to $H$, such that the intersection between $H$ and $L$ is not the same than the one between $K$ and $L$.

Let us see a couple of examples regarding the set operations in the INS-M.

**Example 1** *Let $\mathcal{C} = \{A, B, C\}$ be a INS-C, where $A = \{a, b\}$, $B = \{a, b, c, d\}$ and $C = \{a, b, c, d, e\}$.*

*In this example $B \subseteq C$. Then, $B \cup C = \{a, b, c, d, e\} = C$, $B \cap C = \{a, b, c, d\} = B$ and $C \setminus B = \{e\} \notin \mathcal{C}$.*

**Example 2** *Let $\mathcal{C} = \{A, B, C\}$ be a INS-F, where $A = \{a, b\}$, $B = \{a, b, c, d\}$ and $C = \{a, b, e\}$.*

*In this example* $C||B$. *Then,* $B \cup C = \{a, b, c, d, e\} \notin \mathcal{C}$, $B \cap C = \{a, b\} = A \in \mathcal{C}$ *and* $B \setminus C = \{c, d\} \notin \mathcal{C}$.

We show how a tree can be mapped into a INS-C and vice versa. The following theorem formalizes the intuitive explanation about the mapping of a tree into a INS-C that we have given before. Basically, every couple of nodes $v_j$ and $v_k$ is mapped into a couple of sets $J$ and $K$. If there exists an edge between $v_j$ and $v_k$, say $e_{j,k}$ then the the set $J$ created from $v_j$ is defined as a subset of the set $K$ created from $v_k$. The mapping between a tree and an INS-C reverses the idea described for the mapping of a tree into a NS-C; if a node is parent of another node in a tree, this is mapped into a set which is a subset of the set created from its child node. In Figure 2 we can see a tree mapped into the INS-M as defined by the next theorem.

**Theorem 1** *Let* $T = (V, E)$ *be a tree and let* $\mathcal{C}$ *be a collection of subsets where* $\forall v_i \in V, \exists! I = \Gamma^-(v_i)$. *Then* $\mathcal{C}$ *is an INS-C.*

*Proof.* In order to prove this theorem let us consider a family of subsets $\mathcal{V}_V : V \to \mathcal{C}$ where the set of nodes $V$ is its index set of the family and $\forall v_i \in V$, $V_{v_i} = \Gamma^-(v_i)$.

Let us prove condition 3.1 of Definition 5. Let $v_r \in V$ be the root of $T$. $\mathcal{V}_V(v_r) = V_{v_r} = \Gamma^-(v_r) = \{v_r\} \Rightarrow \forall v_j \in V, \Gamma^-(v_r) \subseteq \Gamma^-(v_j) \Rightarrow V_{v_r} \subseteq V_{v_j}$.

Let us prove condition 3.2 of Definition 5. Ab absurdo suppose that $\exists V_{v_k}, V_{v_h}, V_{v_l} \in \mathcal{V}_V \mid V_{v_h} \subseteq V_{v_k} \wedge V_{v_l} || V_{v_h} \Rightarrow V_{v_l} \cap V_{v_k} \neq V_{v_l} \cap V_{v_h}$.

This means that $\exists v_h, v_k, v_l \in V \mid \Gamma^-(v_h) \subseteq \Gamma^-(v_k) \wedge \Gamma^-(v_l) || \Gamma^-(v_h) \Rightarrow \Gamma^-(v_l) \cap \Gamma^-(v_k) \neq \Gamma^-(v_l) \cap \Gamma^-(v_h)$. $\exists v_j \in V \mid v_j \in (\Gamma^-(v_l) \cap \Gamma^-(v_k)) \wedge v_j \notin (\Gamma^-(v_l) \cap \Gamma^-(v_h)) \Rightarrow v_h \in \Gamma^-(v_k) \wedge v_j \in \Gamma^-(v_k) \wedge v_j \in \Gamma^-(v_l) \wedge v_j \notin \Gamma^-(v_h)$. This means that $v_k$ and $v_h$ must belong to the same branch of $T$; we know that $v_j \in \Gamma^-(v_l) \wedge v_j \in \Gamma^-(v_k)$, thus $v_k$ and $v_l$ must have $v_j$ as a common ancestor and $v_j \notin \Gamma^-(v_h)$. This means that $\{v_j, v_k, v_l\} \in \Gamma^+(v_h)$ but $\Gamma^-(v_l) || \Gamma^-(v_h) \Rightarrow d_V^-(v_l) > 1 \Rightarrow T$ is not a tree.□

Now we can see how an INS-M is mapped into a tree; the following theorem shows that if we map every couple of sets $A_j$ and $A_k$ in an INS-F into a couple of nodes $v_j$ and $v_k$ in a set of nodes $V$ such that there exists an edge $e_{j,k}$ in a set of edges $E$ if and only if $A_j$ is a direct subset of $A_k$ then the graph defined by the nodes in $V$ connected by the edges in $E$ is a tree.

**Theorem 2** *Let* $\mathcal{C}$ *be a INS-C,* $V$ *be a set of nodes and* $E$ *be a set of edges where* $\forall v_j \in V, \exists! J \in \mathcal{C} \wedge \forall e_{j,k} \in E, \exists! J, K \in \mathcal{C} \mid J \subseteq K$. *Then* $T = (V, E)$ *is a tree.*

*Proof.* We have to prove that $(\exists!\ v_r \in V \mid |E^-(v_r)| = 0) \wedge (\forall v_j \in V, j \neq r, |E^-(v_j)| = 1)$. Ab absurdo suppose that $\exists v_r, v_k \in V \mid (|E^-(v_r)| = 0 \wedge |E^-(v_k) = 0)| \vee \exists v_j \in V \mid |E^-(v_j)| > 1$.

If $\exists v_r, v_k \in V \mid |E^-(v_r)| = 0 \wedge |E^-(v_k)| = 0 \Rightarrow \exists J, K \in \mathcal{C} \mid \mathcal{S}^-(J) \cap \mathcal{S}^-(K) = \emptyset \Rightarrow \nexists B \in \mathcal{C} \mid B \subseteq J \wedge B \subseteq K \Rightarrow \mathcal{C}$ is not an INS-C.

If $\exists v_j \in V \mid |E^-(v_j)| > 1 \Rightarrow \exists J, K, L \in \mathcal{C} \mid K \subseteq J \wedge L \subseteq J \wedge K \cap L = \emptyset \Rightarrow L \cap K = \emptyset \neq L \cap J = L \Rightarrow \mathcal{C}$ is not an INS-C.□

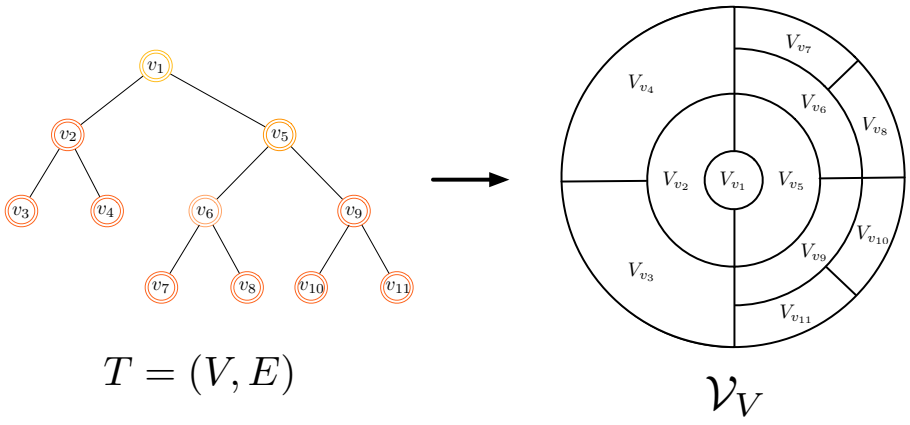


**Fig. 2.** A tree mapped into the INS-M.

The following proposition shows that the set-theoretic operations defined in the INS-M find a correspondent property in the tree.

**Proposition 3** *Let $T = (V, E)$ be a tree, $\mathcal{C}$ be a INS-F mapped from $T$, $J, K, L \in \mathcal{C}$ be three sets and $v_j, v_k, v_L \in V$ be the three correspondent nodes in $T$. Then:*

$$J \cup K = K \Leftrightarrow v_k \in \Gamma^+(v_j) \tag{3.3}$$

$$J \cap K = J \Leftrightarrow v_j \in \Gamma^-(v_k) \tag{3.4}$$

$$J \cap K = L \Leftrightarrow v_l \in \Gamma^-(v_k) \cap \Gamma^-(v_j) \tag{3.5}$$

*Proof.* Property 3.3. Let us prove ($\Rightarrow$). Ab absurdo suppose that $J \cup K = K \Rightarrow v_k \notin \Gamma^+(v_j)$. This means that $J \notin \mathcal{S}^+(K) \Rightarrow J \nsubseteq K \Rightarrow J \cup K \neq K$.

Let us prove ($\Leftarrow$). Ab absurdo suppose that $v_k \in \Gamma^+(v_j) \Rightarrow J \cup K \neq K$. $J \cup K \neq K \Rightarrow J \nsubseteq K \Rightarrow \Gamma^-(v_j) \nsubseteq \Gamma^-(v_k) \Rightarrow v_k \notin \Gamma^+(v_j)$.

**Property 3.4**. The proof of this property is symmetric to the proof of Property 3.3.

**Property 3.5**. Let us prove ($\Rightarrow$). Ab absurdo suppose that $J \cap K = L \Rightarrow v_l \notin \Gamma^-(v_k) \cap \Gamma^-(v_j)$. This implies that $L \nsubseteq J \wedge L \nsubseteq K \Rightarrow L \notin \mathcal{S}^+(J) \wedge L \notin \mathcal{S}^+(K) \Rightarrow J \cap K \neq L$

Let us prove ($\Leftarrow$). Ab absurdo suppose that $v_l \in \Gamma^-(v_k) \cap \Gamma^-(v_j) \Rightarrow J \cap K \neq L$. This means that $L \nsubseteq J \wedge L \nsubseteq K \Rightarrow \Gamma^-(v_l) \nsubseteq \Gamma^-(v_k) \wedge \Gamma^-(v_l) \nsubseteq \Gamma^-(v_j) \Rightarrow v_l \notin \Gamma^-(v_k) \wedge v_l \notin \Gamma^-(v_j) \Rightarrow v_l \notin \Gamma^-(v_k) \cap \Gamma^-(v_j)$.□

Property 3.3 shows that if the union of two sets $\{J, K\} \in \mathcal{C}$ returns $J$ it means that $v_j \in V$ is a descendant of $v_k \in V$; this property is a direct consequence of the definition of INS-F. Property 3.4 shows that if the intersection of two sets $\{J, K\} \in \mathcal{C}$ returns $J$, it means that $v_j \in V$ is an ancestor of $v_k \in V$.

Property 3.5 points out an interesting result: if the intersection of two sets $J, K \in \mathcal{C}$ returns a third set $L \in \mathcal{C}$, then this set corresponds to a common ancestor $v_l$ of the nodes $v_j$ and $v_k$.

## 4 The Lowest Common Ancestor in the INS-M

An important operation performed in the tree data structure is to determine the *lowest common ancestor* ($\mathsf{lca}$) of two nodes. As a first thing let us define the lowest common ancestor in an INS-C.

**Definition 6** *Let $\mathcal{C}$ be an INS-C, and $J, K, L \in \mathcal{C}$ be three sets. $L = J \cap K$ is defined to be the **lowest common ancestor** between $J$ and $K$, say $\mathsf{lca}_{\mathcal{C}}(J, K) = L$.*

The relationship between the $\mathsf{lca}$ in a tree and in an INS-C can be easily determined by exploiting Theorem 1 which shows how to map a tree into an INS-C. Indeed, in the INS-M, the children of a node in a tree correspond to the supersets of the set mapped from that node in the INS-C mapped from the tree.

**Proposition 4** *Let $T = (V, E)$ be a tree, $v_j, v_k, v_l \in V$ be three nodes, $\mathcal{C}$ be a INS-F mapped from $T$ and $J, K, L \in \mathcal{C}$ be three sets. Then:*

$$v_l = \mathsf{lca}_V(v_j, v_k) \Leftrightarrow L = \mathsf{lca}_{\mathcal{C}}(J, K). \tag{4.1}$$

*Proof.* Let us prove ($\Rightarrow$). Ab absurdo suppose that $v_l = \mathsf{lca}_V(v_j, v_k) \Rightarrow L \neq J \cap K$. This implies that $L \nsubseteq J \vee L \nsubseteq K \vee (L \nsubseteq J \wedge L \nsubseteq K) \Rightarrow L \notin \mathcal{S}^+(J) \vee L \notin \mathcal{S}^+(K) \vee (L \notin \mathcal{S}^+(J) \wedge L \notin \mathcal{S}^+(K)) \Rightarrow v_l \notin \Gamma^-(v_j) \vee v_l \notin \Gamma^-(v_k) \vee (v_l \notin \Gamma^-(v_j) \wedge v_l \notin \Gamma^-(v_k)) \Rightarrow v_l \neq \mathsf{lca}_V(v_j, v_k)$.

Let us prove ($\Leftarrow$). Ab absurdo suppose that $L = J \cap K \Rightarrow v_l \neq \mathsf{lca}_V(v_j, v_k)$. This means that $(v_l \notin (\Gamma^-(v_j) \cap \Gamma^-(v_k))) \vee (\exists v_w \in V, v_w \neq v_l \mid (v_w \in (\Gamma^-(v_j) \cap \Gamma^-(v_k))) \wedge (v_w \in \Gamma^+(v_l)))$.

If $v_l \notin (\Gamma^-(v_j) \cap \Gamma^-(v_k)) \Rightarrow L \notin ((\mathcal{S}^+(J) \cup J) \cap (\mathcal{S}^+(K) \cup K)) \Rightarrow J \cap K \neq L$.

If $\exists v_m \in V, v_m \neq v_l \mid (v_m \in \Gamma^-(v_j) \cap \Gamma^-(v_k)) \wedge (v_m \in \Gamma^+(v_l)) \Rightarrow v_l \in \Gamma^-(v_m) \Rightarrow L \subset M \Rightarrow (M \subseteq J \cap K) \wedge (L \subseteq J \cap K) \wedge (M \in \mathcal{S}^-(L) \Rightarrow J \cap K = M$. □

This proposition shows that if we map a tree into a correspondent INS-C also the nodes of the tree are mapped into sets in the collection and thus the $\mathsf{lca}$ between two nodes is mapped into the $\mathsf{lca}$ between the correspondent sets. Furthermore, we can see that the $\mathsf{lca}$ between two sets in the INS-M can be determined *by the intersection of the considered sets.*

**Example 3** *Let $T = (V, E)$ be a tree, and let $\mathcal{C}$ the INS-C mapped from $T$. In order to clearly understand the correspondence between the nodes of the tree and the sets of the collection, let us consider the family of subsets $\mathcal{V}_V : V \to \mathcal{C}$. If we consider the nodes $v_7$ and $v_{11}$ the $\mathsf{lca}_V(v_7, v_{11}) = v_5$ because the path $v_7 P v_1$ intersected with the path $v_{11} P v_1$ returns two nodes: $v_1$ and $v_5$; $v_1$ is the root and by definition its depth is 0, instead $v_5$ has depth 1 thus, it is the lowest common ancestor between $v_7$ and $v_{11}$.*

*We consider the sets $V_{v_7}$ and $V_{v_{11}}$ in $\mathcal{V}_V$ represented in Figure 2; we can see that $V_{v_1}$ is a common subset of both $V_{v_7}$ and $V_{v_{11}}$ as well as $V_{v_5}$. But $V_{v_1} \subset V_{v_5}$. Furthermore, $V_{v_7} \cap V_{v_{11}} = V_{v_5}$ which correspond to the node $v_5 \in V$ of the tree.*

*From this example we can see the correspondence between $\mathsf{lca}_V(v_7, v_{11})$ in $T$ and $\mathsf{lca}_{\mathcal{V}}(V_{v_7}, V_{v_{11}})$ in $\mathcal{V}_V$.*

## 5 Use Case: Modeling an Archive through the INS-M

The tree data structure is adequate to represent the structure of an archive because it properly represents the hierarchical relationships between the archival divisions – see Figure 1a; on the other hand, in a tree it is not straightforward to represent the documents belonging to each archival division. We can say that the tree can represent the structural aspects of an archive but it needs to be somehow extended in order to represent also the content – i.e. the archival resources.

One of the main features of the NESTOR Model is the possibility to express both the hierarchical structure by means of the nested sets and the content by means of the elements belonging to the sets. By means of the NESTOR Model, the archival divisions are represented as nested sets and the hierarchical relationships are retained by their inclusion order. On the other hand, the archival resources are represented as elements belonging to the sets – please see Figure 1b. The INS-M allows us to straightforwardly represent an archive; from the Theorem 1, we know that a tree can be mapped into a INS-F and thus we know that its expressive power is preserved by the INS-M. In this case we can see that the INS-M allows us to define a further level of expressiveness respect to the tree. Furthermore, the INS-M is well-suited for the archival practice; indeed, the idea of "set" shapes the concept of archival division which is a "container" comprising distinct elements that have some properties in common.

The use of the INS-M to model the archives enables their resources to be accessed and shared with a variable granularity in a distributed environment [1]. This is eased by the straightforward integration of the INS-M with the standard de-facto for metadata exchange in distributed environment which is the OAI-PMH [7]. A consequence of the possibility of instantiate the representation of the archives by means of INS-M into OAI-PMH is the further integration of archives in the digital library systems. For these reasons we chose to adopt the NESTOR Model a basic brick of the SIAR (*Sistema Informativo Archivistico Regionale*) system. [6].

The SIAR is a project supported by the Italian Veneto Region which aim is to design and develop a Digital Archive System. The main goal of the SIAR is to develop a system for managing and sharing archive metadata in a distributed environment. Furthermore, another SIAR objective is to develop an information system able to create, manage, access, share and provide advanced services on archival metadata. The design and development of the SIAR system rely on the NESTOR Model; indeed, the INS-M is adopted to model and represent the archives and the archival resources. In this work we do not present the system in details but we focus on the use of the INS-M to perform frequently requested operations on the archives that in particular regard the manipulation and the querying of the archival structure and of the archival resources.

In this context we focus on the querying of the archival structure and resources; in particular, we have seen that the relationships between the archival documents are as important as the documents themselves, thus it is necessary to easily exploit these relationships to infer information from the documents. One on the most important operation is to define the correlation between two

or more documents in the archive. The archivists have to be able to understand why two or more documents belong to the same archive and which is the document or the archival division that put them in relation. We can see that this operation can be modeled as a lowest common ancestor problem; indeed, two or more documents are in relation thanks to a common ancestor that connect them.

By means of the INS-M in the SIAR system we can infer the context of two archival documents without navigating the whole archival hierarchy. In fact, by means of the INS-M when we need to find out the common archival division which contains two or more archival documents we just need to intersect the sets containing the selected documents. The intersection of these sets returns one of their common superset; thanks to Proposition 3 we know it belongs to the INS-C representing the archive and thanks to Proposition 4 we know it is the lowest common ancestor. This property gives us a way to calculate the lowest common ancestor between two elements in a hierarchy – i.e. two documents in an archive – without taking into account the whole hierarchy but just the sets at which these elements belong. The lowest common ancestor represents a relevant case where we exploit the relationships between the tree data structure and the INS-M and the ratio between the operations in a tree and the operations in a INS-C mapped from it. Moreover, the formal basis we defined provides us with the necessary consistency to manipulate and query the archival resources modeled in the INS-M as well as we would do in the tree data structure. This fact allows us to be consistent with the other data models and systems adopted to handle the archives and archival resources.

## 6 Final Remarks

In this paper we presented the NESTOR Model focusing on the Inverse Nested Set Model and its properties detailing its formal definition and the relationships with the tree data structure. In particular, we define the problem of calculating the lowest common ancestor in the INS-M comparing it with the same problem in the tree. We presented a concrete use case based on the archive showing how it is possible to model an archive throughout the INS-M and to apply the presented properties to query the archival resources. The use case is described in the context of the SIAR project.

## Acknowledgments

The work reported has been envisaged in the context of an agreement between the Italian Veneto Region and the University of Padua. EuropeanaConnect[4] (Contract ECP-2008-DILI-52800) and the PROMISE network of excellence[5] (contract n. 258191) projects, as part of the 7th Framework Program of the European Commission, have partially supported the reported work.

[4] `http://www.europeanaconnect.eu/`
[5] `http://www.promise-noe.eu/`

## References


1. A. Agosti, N. Ferro, and G. Silvello. The NESTOR Framework: Manage, Access and Exchange Hierarchical Data Structures. In *Proceedings of the 18th Italian Symposium on Advanced Database Systems*, pages 242–253. Società Editrice Esculapio, Bologna, Italy, 2010.
2. M. A. Bender, M. Farach-colton, G. Pemmasani, S. Skiena, and P. Sumazin. Lowest Common Ancestors in Trees and Directed Acyclic Graphs. *J. Algorithms*, 57:75–94, 2005.
3. B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order - 2nd Ed.* Cambridge University Press, Cambridge, UK, 2002.
4. L. Duranti. *Diplomatics: New Uses for an Old Science.* Society of American Archivists and Association of Canadian Archivists in association with Scarecrow Press, Lanham, Maryland, USA, 1998.
5. N. Ferro and G. Silvello. A Methodology for Sharing Archival Descriptive Metadata in a Distributed Environment. In *Proc. 12th Eur. Conf. on Research and Advanced Tech. for Digital Libraries*, pages 268–279. LNCS 5173, Springer, Germany, 2008.
6. N. Ferro and G. Silvello. Design and Development of the Data Model of a Distributed DLS Architecture for Archive Metadata. In $5^{th}$ *IRCDL - Italian Research Conference on Digital Libraries*, pages 12–21. DELOS: an Association for Digital Libraries, 2009.
7. N. Ferro and G. Silvello. The NESTOR Framework: How to Handle Hierarchical Data Structures. In *Proc. 13th Eur. Conf. on Research and Advanced Tech. for Digital Libraries*, pages 215–226. LNCS 5714, Springer, Germany, 2009.
8. A. J. Gilliland-Swetland. *Enduring Paradigm, New Opportunities: The Value of the Archival Perspective in the Digital Environment.* Council on Library and Information Resources, Washington, DC, USA, 2000.
9. P. R. Halmos. *Naive Set Theory.* D. Van Nostrand Company, Inc., New York, NY, USA, 1960.
10. Thomas Jech. *Set Theory.* Springer-Verlag, Berlin, Germany, 2003.
11. D. E. Knuth. *The Art of Computer Programming, third edition*, volume 1. Addison Wesley, Reading, MA, USA, 1997.
12. R. Pearce-Moses. *Glossary of Archival And Records Terminology.* Society of American Archivists, 2005.
13. D. V. Pitti. Encoded Archival Description. An Introduction and Overview. *D-Lib Magazine*, 5(11), 1999.
14. C. J. Prom and T. G. Habing. Using the Open Archives Initiative Protocols with EAD. In *Proc. 2nd ACM/IEEE Joint Conf. on Digital Libraries*, pages 171–180. ACM Press, USA, 2002.
15. C. J. Prom, C. A. Rishel, S. W. Schwartz, and K. J. Fox. A Unified Platform for Archival Description and Access. In *Proc. 7th ACM/IEEE Joint Conf. on Digital Libraries*, pages 157–166. ACM Press, USA, 2007.
16. H. Van de Sompel, C. Lagoze, M. Nelson, and S. Warner. The Open Archives Initiative Protocol for Metadata Harvesting (2nd ed.). Technical report, Open Archive Initiative, p. 24, 2003.
17. J. Vegas, F. Crestani, and P. de la Fuente. Context Representation for Web Search Results. *Journal of Information Science*, 33(1):77–94, 2007.