Coining goldMEDAL: A New Contribution to Data Lake Generic Metadata Modeling

Étienne Scholly
Université de Lyon, Lyon 2,
UR ERIC & BIAL-X
Lyon, France
etienne.scholly@bial-x.com

Pegdwendé N. Sawadogo
Université de Lyon, Lyon 2,
UR ERIC
Lyon, France
pegdwende.sawadogo@univ-lyon2.fr

Pengfei Liu
Université de Lyon, Lyon 2,
UR ERIC
Lyon, France
pengfei.liu@eric.univ-lyon2.fr

Javier A. Espinosa-Oviedo
Université de Lyon, Lyon 2,
UR ERIC-LAFMIA lab
Lyon, France
javier.espinosa@imag.fr

Cécile Favre
Université de Lyon, Lyon 2,
UR ERIC
Lyon, France
cecile.favre@univ-lyon2.fr

Sabine Loudcher
Université de Lyon, Lyon 2,
UR ERIC
Lyon, France
sabine.loudcher@univ-lyon2.fr

Jérôme Darmont
Université de Lyon, Lyon 2,
UR ERIC
Lyon, France
jerome.darmont@univ-lyon2.fr

Camille Noûs
Université de Lyon, Lyon 2,
Laboratoire Cogitamus
Lyon, France
camille.nous@cogitamus.fr

ABSTRACT

The rise of big data has revolutionized data exploitation practices and led to the emergence of new concepts. Among them, data lakes have emerged as large heterogeneous data repositories that can be analyzed by various methods. An efficient data lake requires a metadata system that addresses the many problems arising when dealing with big data. In consequence, the study of data lake metadata models is currently an active research topic and many proposals have been made in this regard. However, existing metadata models are either tailored for a specific use case or insufficiently generic to manage different types of data lakes, including our previous model MEDAL. In this paper, we generalize MEDAL's concepts in a new metadata model called goldMEDAL. Moreover, we compare goldMEDAL with the most recent state-of-the-art metadata models aiming at genericity and show that we can reproduce these metadata models with goldMEDAL's concepts. As a proof of concept, we also illustrate that goldMEDAL allows the design of various data lakes by presenting three different use cases.

1 INTRODUCTION

While the big data revolution has shaken up the entire field of data management and analytics, new concepts have emerged to meet these new challenges. Data lakes belong to such new concepts. First introduced by James Dixon, a data lake is a vast repository of raw and heterogeneous data from which various analyses can be performed [4]. Data lakes quickly gained popularity and several teams started to address research issues [13, 15]. A key one is efficient metadata management for avoiding data lakes to turn into unexploitable data swamps [10, 11, 16, 19, 22].

However, most metadata management proposals in the literature [1, 8, 14], and their associated implementations, give few details on the way data are conceptually organized and are thence hardly reusable. Thus, other researchers proposed more theoretical approaches named metadata models. Such approaches aim to provide detailed guidelines to metadata system design, while being generic, i.e., flexible and adaptable to many use cases. Yet, data lake generic metadata modeling is still an open research issue. A feature-based assessment indeed shows that none of the existing metadata models is generic enough, including our own MEtadata model for DAta Lakes (MEDAL) [20].

To address this genericity issue, we introduce goldMEDAL, a revision of our MEDAL model. We define goldMEDAL through a classical three-level modeling process (i.e., conceptual, logical and physical). We choose a formal representation to avoid ambiguity but also provide a UML representation for readability. The logical level is a translation of the concepts using graph theory. Eventually, we describe three different physical models as proofs of concept. Furthermore, to highlight goldMEDAL's genericity, we show that the concepts of our metadata model help model state-of-the-art metadata models from the literature.

The remainder of this paper is organised as follows. Section 2 reviews and discusses existing data lake metadata models. Section 3 presents goldMEDAL's conceptual and logical models. Section 4 illustrates how goldMEDAL generalises other data lake metadata models and how it can be used to implement different data lakes. Finally, Section 5 concludes this paper and hints at future research.

2 RELATED WORKS

Metadata management plays a vital role in data lakes. Indeed, in the absence of a fixed schema, data querying and analyses depend on an efficient metadata system. Several approaches help manage metadata in data lakes. However, only a few of them provide enough detail to ensure reusability. We refer to them as metadata models. In this section, we review state-of-the-art metadata models (Section 2.1) and compare them with respect to genericity (Section 2.2).

2.1 Metadata Models for Data Lakes

GEMMS (Generic and Extensible Metadata Management System) is a pioneer generic metadata model for data lakes [17]. GEMMS features two abstract entities: *data file* and *data unit*. A data file represents a generic data source. A data unit represents an identifiable data element inside a data source. Each data file is composed of a set of data units (e.g., a spreadsheet file is composed of a set of sheets). Data files and data units can be enriched with atomic or complex metadata values. However, GEMMS requires information on data structure to operate. Thus, making it unsuitable for working with unstructured data.

Ground is another generic metadata model [9] that can be used for modeling metadata in data lakes (although not specifically designed for that). Ground tracks *data context* (metadata) at three levels: 1) metadata properties, 2) data usage history and 3) data versioning. Although more extensive than GEMMS, Ground (as well as GEMMS) does not take in charge data linkage even though this type of metadata has been identified as relevant in data lakes [6, 20].

Based on GEMMS' data file and data units concepts, The model of Diamantini et al. adds *similarity links* between data units to indirectly link data files [3]. However, their model does not include important metadata such as data versioning and usage tracking as compared to Ground.

Similar to Diamantini et al., Ravat and Zhao propose a model where each data file can be associated with atomic and complex metadata [18], including metadata properties, data history and links with other data files. The main contribution of this model is the notion of *zone* metadata. Many data lake architectures consider the existence of zones (e.g., raw data zone, processed data zone) [7, 18]. Zone metadata specifies the zones where data is located. However, Ravat and Zhao's model cannot simultaneously represent different data granularity levels as previous models do [3, 17].

MEDAL represents data through three main concepts: *data objects*, *representations* and *versions* [20]. Data objects correspond to GEMMS' data files. Representations correspond to the result of transformed objects. Versions represent objects updates. Both, representations and versions, are materialized in the data lake. Thus, MEDAL gives alternative ways to track data linkage and zone metadata through the concepts of versions and representations, respectively. MEDAL also supports linkage metadata through categorizations and similarity links. However, MEDAL does not support multiple data granularity levels either.

Finally, HANDLE (Handling metAdata maNagement in Data LakEs), uses the generic concept of *data entity* to represent both, data files and parts of data files, which helps HANDLE support any granularity level [5]. In HANDLE, each data entity is associated with tags that represent zones, granularity levels or categorizations. HANDLE can also connect data entities together through containment links (e.g., between a table and a tuple). HANDLE provides concepts that subsume most of the concepts of the previous metadata models.

2.2 Genericity of Metadata Models

A generic metadata model should adapt to any data lake use case. As each use case requires specific metadata management features, we consider that the most abundant features a metadata model supports, the most generic it is. Therefore, features are a suitable way to compare metadata models.

To the best of our knowledge, there exist two feature-based comparisons of data lake metadata models in the literature. We introduced six relevant features: semantic enrichment, data indexing, data polymorphism, data versioning, link generation and usage tracking [20]; while Eichler et al. identified three other features: metadata properties, zone metadata and the support of multiple granularity levels [5].

Considering that both the above sets of features are relevant, we propose to combine them for comparing the genericity of metadata models. Beyond simply unioning features, we merge data polymorphism with zone metadata, as these features both refer to the same concept. We also split link generation in two new features, namely similarity links and categorization, because some metadata models support only one of them. Eventually, we omit data indexing in this comparison, considering that indexing does not actually induce metadata modeling issues. Although indexing is definitely relevant to assess metadata systems [20], this feature seems less suited to metadata models.

All in all, we obtain a list of eight features that can serve to compare data lake metadata models and evaluate their genericity.

(1) Semantic enrichment
(2) Data polymorphism/multiple zones
(3) Data versioning
(4) Usage tracking
(5) Categorization
(6) Similarity links
(7) Metadata properties
(8) Multiple granularity levels

Table 1 highlights the features supported by all the models reviewed in Section 2.1. It shows that none of them support all the features we identify.

3 GOLDMEDAL METADATA MODEL

Section 2.1 establishes that, of the eight criteria used to compare data lake metadata models, none ticked all the boxes. In this section, we thoroughly describe goldMEDAL, a substantial evolution of MEDAL that generalizes its concepts while addressing all the features identified in Section 2.2.

A metadata model can be expressed "in the form of an explicit schema, a formal definition, or a textual description" [5]. In this paper, we choose a formal approach for the sake of precision. Yet, for the sake of readability and communication with possibly non-computer scientists, we also provide a semi-formal UML model. Moreover, we use a conventional data modeling approach that leverages a conceptual, a logical and a physical model, to demonstrate the actual implementation process of our metadata model.

Section 3.1 presents goldMEDAL's formal and semi-formal conceptual models. Section 3.2 details the translation of goldMEDAL's concepts into a logical, graph-based model. For the sake of clarity, the examples we use are the same examples in both sections, i.e., examples at the conceptual level are translated at the logical level. Eventually, example physical models, i.e., metadata models actually implemented in data lakes with goldMEDAL, are presented in Section 4.2.

3.1 Conceptual Model

In MEDAL, data items were considered either as *raw data*, or as *versions* or *representations* derived from raw data. The concepts of version and representation were used to express updated and transformed data, respectively. While modeling metadata for

Table 1: Features supported by data lake metadata models

Features ↓ \ Models →	GEMMS	Ground	Diamantini et al.	Ravat & Zhao	MEDAL	HANDLE	goldMEDAL
Semantic enrichment	✓	✓	✓	✓	✓	✓	✓
Polymorphism/multiple zones			✓	✓	✓	✓	✓
Data versioning		✓		✓	✓		✓
Usage tracking		✓		✓	✓	✓	✓
Categorization	✓	✓		✓	✓	✓	✓
Similarity links			✓	✓	✓	✓	✓
Metadata properties	✓	✓		✓	✓	✓	✓
Multiple granularity levels	✓		✓			✓	✓
Total	4/8	5/8	4/8	7/8	7/8	7/8	8/8

various data lakes, we found that more data items were possible, e.g., temporal representations. Thus, we decided to generalize any such concepts into a global concept named **data entity** in goldMEDAL.

Accordingly, we also generalized in goldMEDAL:

- *update* and *transformation* operations that served to track the lineage of representations and versions, respectively, as well as *parenthood relationships* that express fusion operations, into the concept of **process**;
- *similarity links* into the global concept of **link**.

Eventually, we retained in goldMEDAL the MEDAL concept of **grouping**, which notably allows multiple data granularity levels.

All the main goldMEDAL concepts (data entity, grouping, link and process) are characterized by attributes or properties that constitute their internal metadata.

3.1.1 Data Entity. Data entities are the basic units of our metadata model. They are flexible in terms of data granularity. For example, a data entity can represent a spreadsheet file, a textual or semi-structured document, an image, a database table, a tuple or an entire database. The introduction of any new element in the data lake leads to the creation of a new data entity.

Definition 3.1. The set of data entities is denoted $\mathcal{E} = \{e_i\}_{i \in \mathbb{N}^*}$.

3.1.2 Grouping. A grouping is a set of groups; a **group** brings together data entities based on common properties. For example, the raw and preprocessed data zones common in data lake architectures are the groups of a zone grouping. Another example is a grouping of textual documents according to the language of writing.

Definition 3.2. The set of groupings is denoted $\mathcal{G} = \{G_j\}_{j \in \mathbb{N}^*}$, with $G_j = \{\Gamma_{jk}\}_{k \in \mathbb{N}^*}$ and $\Gamma_{jk} \subseteq \mathcal{E}$ is a group.

Example 3.3. To get back to our previous examples, $\mathcal{G} = \{G_1, G_2\}$. $G_1 = \{\Gamma_{11}, \Gamma_{12}\}$ is the zone grouping, with Γ_{11} and Γ_{12} being the raw data and processed data zones, respectively. $G_2 = \{\Gamma_{21}, \Gamma_{22}\}$ is the language grouping, with Γ_{21} and Γ_{22} the groups corresponding to French and English languages, respectively. Note that the groupings G_j are deliberately not partitions of $\mathcal{E}$. Thus, a bilingual French-English document can belong to both groups Γ_{21} and Γ_{22}.

3.1.3 Link. Links are used to associate either data entities with each other or groups of data entities with each other. They can be oriented or not. They allow the expression of, e.g., simple similarity links between data entities or hierarchies between groups. For example, a temporal hierarchy month → quarter would have the months of January, February and March linked to the first quarter of a given year.

Definition 3.4. The set of links is denoted $\mathcal{L} = \{l_m\}_{m \in \mathbb{N}^*}$, with either:

- $l_m : \mathcal{E} \rightarrow \mathcal{E}$,
- $l_m : G_j \rightarrow G_{j'}$ and $j \neq j'$.

Example 3.5. Let us elaborate the sample hierarchy month → quarter. Let $G_3 = \{Jan, Feb, ..., Dec\}$ a grouping of data entities per month and $G_4 = \{Q1, Q2, Q3, Q4\}$ be a grouping of quarters in a year. Now, let us make explicit some data entities and their groups: $Jan = \{e_1, e_2\}$, $Fev = \{e_3\}$, $Mar = \{e_4\}$; $Q1 = \{e_1, e_2, e_3, e_4\}$. Link l_1 materializes the hierarchical link between groups G_3 and G_4: $Jan \xrightarrow[l_1]{} Q1, Feb \xrightarrow[l_1]{} Q1, Mar \xrightarrow[l_1]{} Q1$. Inversely, $Q1 \xrightarrow[l_1^{-1}]{} \{Jan, Feb, Mar\}$.

A functional notation may also be used: $l_1(Jan) = Q1, l_1(Feb) = Q1$, $l_1(Mar) = Q1, l_1^{-1}(Q1) = \{Jan, Feb, Mar\}$. Also note that $Q1 = Jan \cup Feb \cup Mar$.

3.1.4 Process. A process refers to any transformation applied to a set of data entities that produces a new set of data entities.

Definition 3.6. The set of processes is denoted $\mathcal{P} = \{P_n\}_{n \in \mathbb{N}^*}$, with $P_n = \{I_n, O_n\}$, $I_n \subseteq \mathcal{E}$ the set of input data entities of P_n and O_n the set of output data entities that is integrated into $\mathcal{E}$ ($\mathcal{E} \leftarrow \mathcal{E} \cup O_n$).

Example 3.7. Process P_1 splits a set of textual documents $D \subseteq \mathcal{E}$ into a set of text fragments $F \subseteq \mathcal{E}$. Here, $I_1 = D$ and $O_1 = F$.

3.1.5 UML model. Figure 1 features goldMEDAL's conceptual model as a UML class diagram. All the concepts of goldMEDAL, including group, are modeled as classes (data entity, grouping, group and process) or association classes (entity link and group link, which are labeled E-Link and G-Link in Figure 1, respectively).

Eventually, although they are not depicted in Figure 1, all classes and association classes bear attributes that model metadata properties. These attributes may be of any type, including lists, and of course vary with respect to use cases.

3.2 Logical Model

As MEDAL and HANDLE did, though at the physical level, we choose to design goldMEDAL's logical model as a graph, which is

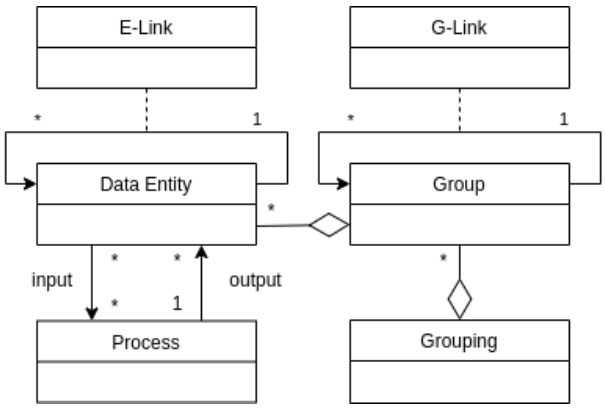

Figure 1: UML class diagram of goldMEDAL

particularly well-suited to depict relationships between different concepts.

Thus, in this section, we translate the concepts defined in Section 3.1 into graph nodes, edges and hyperedges, using the same indices, e.g., i, j, k... Moreover, we illustrate the translation with the examples used at the conceptual level. Finally, we also propose a graphic illustration of goldMEDAL's logical model.

3.2.1 Translation of Data Entity. Data entities are modeled by nodes that carry attributes.

Definition 3.8. The set of nodes is denoted $\mathcal{N} = \{n_i\}_{i \in \mathbb{N}^*}$. Each node $n_i \in \mathcal{N}$ carries attributes.

Example 3.9. A PDF file stored in the data lake can be represented by a node n_1.

3.2.2 Translation of Grouping. A group is represented by a non-oriented hyperedge, i.e., an edge that can link more than two nodes. A grouping is modeled by a set of hyperedges.

Definition 3.10. A hyperedge (a group) is denoted $\theta_{jk} \subseteq \mathcal{N}$, with $j, k \in \mathbb{N}^*$. Any θ_{jk} carries attributes.

Definition 3.11. The set of hyperedges of grouping j is denoted $H_j = \{\theta_{jk}\}$ and carries attributes. The set of hyperedge sets (set of groupings) is denoted $\mathcal{H}$.

Example 3.12. Let us translate Example 3.3. $\mathcal{H} = \{H_1, H_2\}$. $H_1 = \{\theta_{11}, \theta_{12}\}$ is the set of hyperedges representing the zone grouping, with θ_{11} and θ_{12} the hyperedges representing the raw data and processed data zones, respectively. $H_2 = \{\theta_{21}, \theta_{22}\}$ is the set of hyperedges representing the language grouping, with θ_{21} and θ_{22} the hyperedges representing the groups corresponding to French and English languages, respectively.

3.2.3 Translation of Link. Links may model relationships between either data entities (nodes) or groups (hyperedges). They are modeled by edges.

Definition 3.13. The set of edges is denoted $\mathcal{A} = \{a_m\}_{m \in \mathbb{N}^*}$, with any a_m being either:

- an edge, oriented or not, connecting two nodes. Then, $a_m = (n_i, n_{i'}) \in \mathcal{N}^2$;
- an oriented edge connecting two hyperedges. Then, $a_m = (\theta_{jk}, \theta_{j'k'}) \in H_j \times H_{j'}$.

In both cases, the edge carries attributes.

Example 3.14. To get back to the sample hierarchy month $\rightarrow$ quarters from Example 3.5, $H_3 = \{\theta_{Jan}, \theta_{Feb}, ..., \theta_{Dec}\}$ is a set of hyperedges representing a grouping of data entities per month. $H_4 = \{\theta_{Q1}, \theta_{Q2}, \theta_{Q3}, \theta_{Q4}\}$ is a set of hyperedges representing the grouping of quarters in a year. Let us make this explicit with instances. $\theta_{Jan} = \{n_1, n_2\}$, $\theta_{Fev} = \{n_3\}$, $\theta_{Mar} = \{n_4\}$; $\theta_{T1} = \{n_1, n_2, n_3, n_4\}$. Edge a_1 materializes the hierarchical link between H_3 and H_4: $\theta_{Jan} \xrightarrow[a_1]{} \theta_{Q1}, \theta_{Feb} \xrightarrow[a_1]{} \theta_{Q1}, \theta_{Mar} \xrightarrow[a_1]{} \theta_{Q1}$. Inversely, $\theta_{Q1} \xrightarrow[a_1^{-1}]{} \{\theta_{Jan}, \theta_{Feb}, \theta_{Mar}\}$.

3.2.4 Translation of Process. A process is modeled by an oriented hyperedge.

Definition 3.15. The set of oriented hyperedges modeling processes is denoted $\mathcal{Q} = \{\Pi_n\}_{n \in \mathbb{N}^*}$, with $\Pi_n = \{\Upsilon_n, \Omega_n\}$, $\Upsilon_n \subseteq \mathcal{N}$ being the set of input nodes of Π_n and Ω_n the a set of output nodes integrated to $\mathcal{N}$ ($\mathcal{N} \leftarrow \mathcal{N} \cup \Omega_n$). Any Π_n carries attributes.

Example 3.16. $\Pi_1 = \{\Upsilon_1, \Omega_1\}$ is an oriented hyperedge representing the process of splitting a set of textual documents (Example 3.7) represented by the set of nodes $N_D \subseteq \mathcal{N}$, into a set of text fragments represented by the set of nodes $N_F \subseteq \mathcal{N}$. Then, $\Upsilon_1 = N_D$ and $\Omega_1 = N_F$.

3.2.5 Sample Graph Representation. Figure 2 provides a schematic representation of the examples above. Let us introduce eight data entity nodes $\{n_i\}_{i \in [1,8]}$ colored in orange.

Example 3.12 is depicted on the left-hand side of Figure 2. Groups of H_1 are colored in purple, while H_2's are blue. We can see that n_1 and n_3 belong to the raw data group θ_{11}, while n_2 and n_4 are in the processed data group θ_{12}. Moreover, n_1, n_2 and n_3 are in the French language group θ_{21}, and n_4 is in the English language group θ_{22}.

Example 3.14 is represented at the center of Figure 2. Groups of H_3, namely θ_{Jan}, ..., θ_{Dec} are colored in green and groups of H_4 (θ_{Q1}, ..., θ_{Q4}) are colored in grey. Hyperedge a_1 connects groups of H_3 to H_4's.

Finally, Example 3.16 is depicted on the right-hand side of Figure 2. n_5 is a textual document split in fragments n_6, n_7 and n_8. Π_1's input and output Υ_1 and Ω_1, respectively, are colored in yellow.

4 GOLDMEDAL ASSESSMENT

In this section, we discuss goldMEDAL's genericity. To this end, we show in Section 4.1 that all three most complete metadata models can be modeled with goldMEDAL. In Section 4.2, we present our ongoing implementation work of goldMEDAL on distinct use cases.

4.1 Comparison of State-of-the-Art Metadata Models with goldMEDAL

To evaluate goldMEDAL's genericity, we compare it with the three metadata models that are both the most recent and the most complete among metadata models, i.e., MEDAL, Ravat and Zhao's and HANDLE (Section 2.2).

For each comparison, we use a two-column table. The first column lists goldMEDAL's concepts, and the second column the corresponding concepts of the compared model. When any concept does not have an equivalent, it is marked with "—".

4.1.1 MEDAL vs. goldMEDAL. goldMEDAL's four main concepts help generalize all of MEDAL's concepts (Table 2). Data entity generalizes the concepts of version and representation. Grouping generalizes the concepts of object and grouping (in

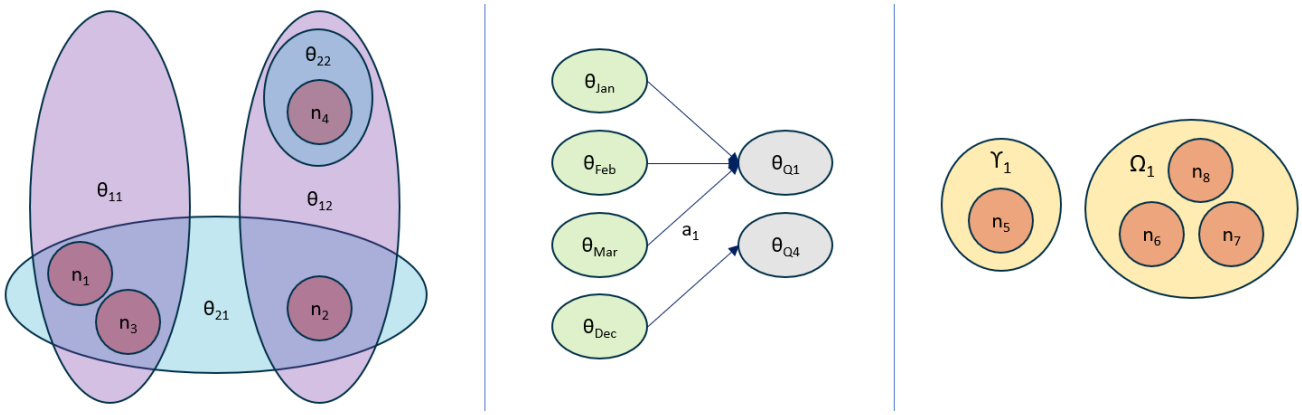

Figure 2: Sample goldMEDAL graph logical model

the sense of MEDAL). Link generalizes the concepts of similarity link. Finally, process generalizes transformation, update and parenthood relationship.

Table 2: goldMEDAL and MEDAL concepts

goldMEDAL	MEDAL
Data entity	Version, Representation
Grouping	Object, Grouping
Link	Similarity link
Process	Update, Transformation, Parenthood relationship

Note that we do not mention in this comparison global metadata existing in MEDAL. We indeed consider that elements such as logs or indexes mostly induce implementation rather than metadata modeling issues.

Yet, other forms of global metadata, namely semantic resources such as thesauruses and ontologies, can definitely be modeled with goldMEDAL using the node, grouping and link concepts.

4.1.2 Ravat and Zhao's Metadata Model vs. goldMEDAL. goldMEDAL can handle nearly all concepts of Ravat and Zhao's metadata model [18] (Table 3). Data entity generalizes the concept of dataset and all its subclasses, such as Datalake_Datasets or Source _Datasets. Grouping generalizes the concepts of keyword. Finally, link and process directly correspond to relationship and process, respectively.

Table 3: goldMEDAL and Ravat & Zhao concepts

goldMEDAL	Ravat & Zhao
Data entity	Dataset, Subclass
Grouping	Keyword
Link	Relationship
Process	Process
—	User, Access

However, two concepts of Ravat and Zhao's metadata model, namely user and access, have no explicit equivalent in goldMEDAL, though they could be classified as global metadata. Users and accesses can indeed be modeled as data entities and processes, respectively.

4.1.3 HANDLE vs. goldMEDAL. goldMEDAL can also generalize HANDLE's concepts (Table 4). Data entity generalizes both data and metadata, since a data entity is a representation of data that also contains metadata properties. Grouping generalizes three concepts: Categorization, ZoneIndicator, and GranularityIndicator. Finally, process has no direct match in HANDLE, although its authors show processes can be modeled through Action metadata instances of HANDLE's categorization extension [5].

Table 4: goldMEDAL and HANDLE concepts

goldMEDAL	HANDLE
Data entity	Data, Metadata
Grouping	Categorization, ZoneIndicator GranularityIndicator
Link	Link
Process	—

Handling multiple granularity levels as in HANDLE was not supported by MEDAL, so it was a design objective for goldMEDAL. Although there is no explicit granularity indicator in goldMEDAL, any data entity could have a granularity property. However, there is more efficient way by defining data entities on the finest possible granularity level. Then, coarser granularity levels are obtained with groupings. For example, if each data entity corresponds to a tuple in a relational database, then a grouping represent a set of tables.

4.2 goldMEDAL Physical Models

To show that goldMEDAL can model different business issues and manage various functionalities while remaining as simple as possible, we apply our metadata model to three different use cases. We also exemplify how goldMEDAL's logical model (Section 3.2) can be translated into different physical models.

4.2.1 Public Housing Data Lake. For social landlords (agents or agencies providing social housing), the use of data is nothing new, whether through business intelligence for patrimony

management or with data science methods for non-payment forecasting. However, landlords are facing two main problems. On the one hand, their analyses are conducted separately: in different environments, by different individuals and with different tools. This implies that collaborative work on the same data is impossible. On the other hand, landlords know how to use their data, but have much more difficulty capturing and exploiting "external" data. Yet their dwellings are located in environments with their own characteristics (transportation, climate, employment rate, education, etc.), which affect the attractiveness of the dwellings. Being able to combine this external information with landlords' data would be a real asset for understanding their patrimony.

A data lake can store both "internal" data from social landlords as well as "external" data gathered on the Internet. In addition, all types of analyses can be carried out from the data lake.

HOUDAL (public HOUsing DAta Lake). The data lake implemented for social landlords [21] is based on a Web application, and thus is composed of two major parts: the front-end (or client part) is the user interface for depositing new data, for creating new metadata and for consulting existing metadata; the back-end (or server part) features various services such as an API, the metadata system, data storage, and a user management service.

HOUDAL Metadata System. goldMEDAL's metadata model has been implemented into the Neo4J graph database management system[1]. Since Neo4J does not allow to have hyperedges, we create a node for each concept. Thus, entities, groups, groupings, links and processes translate as nodes, each bearing a label and attributes.

Data entities. The different data files that populate the data lake are data entities. They can be either raw data files sent by landlords (often in comma separated value files) or reworked data, sometimes stored in various formats such as .pkl or .RData, for Python and R analyses, respectively. Each data entity has its node labeled :ENTITY and the entity's properties, such as file name or description, are stored in the node's attributes.

Groupings for Categorizing Data Entities. With HOUDAL, users can create as many groupings as necessary, and several groups for each grouping. Data entities can be linked to zero, one or several groups for each grouping. In Neo4J, groupings are modeled by nodes carrying a :GROUPING label. Groups are also nodes, carrying both a :GROUP label and the grouping's name as a second label, in order to facilitate querying. A data entity node (resp. group node) is linked to a group node (resp. grouping node) with an edge labeled with the grouping's name (resp. :GROUPING). With groups and groupings, users can, for example, determine whether it is internal or external data, or the data refinement level (zones), and so on.

Processes for Tracking Data Lineage. Like other goldMEDAL concepts, a process is also modeled by a node in Neo4J, bearing the :PROCESS label. A process can be a script for transforming or cleaning a data file, i.e., a data entity. If a data entity is the input of a process, there is an edge labeled :PROCESS_IN from the entity node to the process node. Inversely, an edge labeled :PROCESS_OUT from the process node to the entity node is created if a new data entity is generated by the process.

Example. Figure 3 presents a sample of metadata stored in Neo4J. Data entity nodes are colored in red. On both sides of the figure, a data entity node is highlighted: some of its attributes are depicted at the bottom in grey.

The left-hand side of Figure 3 gives an example of groupings. There are three groupings: a zone grouping, a format grouping and a granularity grouping. Each grouping has its group nodes, colored in green, purple and blue, respectively. Data entity nodes are connected to group nodes with an edge. For example, we can see that the highlighted data entity node (on the left) is a raw .csv file, and the granularity level is "Tenant", meaning that each line corresponds to a tenant. Note that in Neo4J, groupings are also modeled as nodes, but are not represented in this Figure.

An example of process is depicted on the right-hand side of Figure 3. The process node is colored in yellow. We can see that three data entity nodes are the process' input, and three data entity nodes are the process' output, meaning that they are generated by the process.

HOUDAL is operational and is currently being tested by social landlords. Nevertheless, we have many areas for improvement to work on, to make the application more robust and more user-friendly. In addition, we continue to discuss with social landlords to identify new needs, which could be the subject of future work to add a new feature to our data lake.

4.2.2 Textual and Tabular Data Lake. The AUDAL data lake is motivated by researchers in management science who want to analyze the effect of servicization (i.e., the transition from supplying products to supplying services) and digitization on small and medium sized companies' economic performance [2]. Source data are various textual documents (annual reports, press releases, websites, social media posts) and spreadsheet files featuring qualitative (e.g., stocks) and qualitative (e.g., degree of servicization) characteristics.

Metadata Management in AUDAL. AUDAL's metadata system is architectured in three levels. The first level manages data entities. Data entities, i.e., textual documents and spreadsheet tables, are categorized as *raw* and *refined*. Raw tables or documents are actually pointers to the corresponding files in their original format. Raw data entities store metadata properties, in the form of Neo4J node attributes, e.g., file author(s), date of creation, etc. Refined data entities are automatically generated from raw data entities. They are transformed so as to be exploited in analyses. More concretely, raw textual documents are refined into bag-of-word vectors or document embedding vectors stored in the MongoDB document-oriented database management system[2], and referenced from Neo4J nodes (Figure 4). Similarly, raw spreadsheet tables are refined in relational tables to benefit from SQL querying.

The second level in AUDAL's metadata system handles relationships between data items. We use two kinds of relationships in accordance with goldMEDAL concepts: groupings and (similarity) links. Some of the groupings relate to both tabular and textual data, e.g., groupings on the MIME type or data source. Conversely, others are relevant for only one type of data, e.g., the grouping on the language of documents. We materialize groupings in Neo4J through a set of nodes. Each grouping is a simple node with which all associated groups are linked. Then, groups are in turn linked to the corresponding data entities.

We define two types of links with respect to the type of data they relate to. *Document similarity links* express how much

[1] https://neo4j.com

[2] https://www.mongodb.com

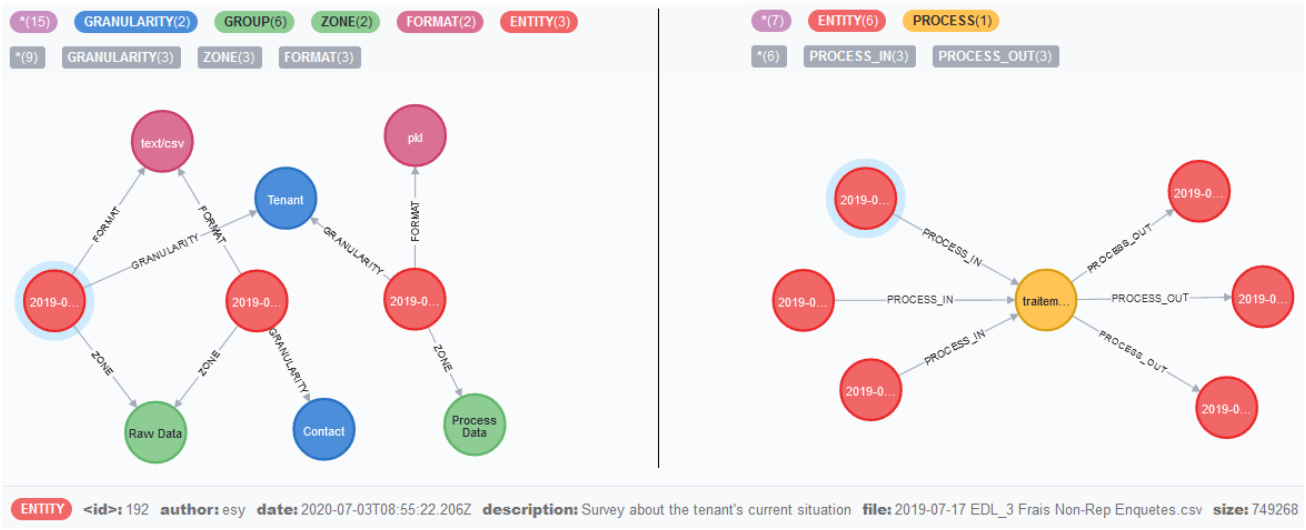

Figure 3: HOUDAL sample Neo4J metadata

two documents are similar. These links are materialized by non-oriented edges between data entity nodes in Neo4J. Similarly, we express links between tabular data with *Table joinability links*. Such links (labeled *PK_FK_LINK* in Figure 4) actually represent some automatically detected functional dependencies between columns from different tables. In Neo4J, table joinability edges are oriented.

Eventually, our model's third level is constituted of metadata used to speed up or enhance analyses. It includes indexes that allow and speed up keyword-based search on textual documents as well as spreadsheet files. These indexes are managed by ElasticSearch[3]. Moreover, AUDAL's metadata system also includes semantic resources, i.e., dictionaries and thesaurus. Such resources, stored in MongoDB, allow amongst other automatic query extension.

Analyses with AUDAL. AUDAL allows both data retrieval and content analyses. Data retrieval works in three different ways. The first way exploits indexes to allow term-based queries. It is effective for both textual documents and tabular data. AUDAL also provides navigation as a solution to discover data of interest. This is done by intersecting groups from different groupings. For example, such queries allow finding data from a specific source and created on a specified year. Finally, data can be retrieved using relatedness, starting from a specified data object and then finding the most related data, namely similar documents or joinable tables.

Content analyses are actually a way to aggregate data. In the case of textual documents, such analyses include document clustering or scoring with respect to a set of keywords and text concordance. Tabular data are exploited through SQL queries, the clustering of table rows and correlation analyses between columns.

4.2.3 Archaeological Data Lake. This data lake was designed during the course of the multidisciplinary project "Hyper thesaurus and data lakes: Mine the city and its archaeological archives" (HyperThesau) [2, 12]. Let us name it ArchaeoDAL, in echo to HOUDAL and AUDAL, though it was actually never called so.

Archaeological data may bear many different types, e.g., textual documents (excavation reports), images (photographs, drawings, plans...), sensor data, chemical analysis results, etc. Even structured data are often produced by various devices that are not compatible with each other. Moreover, the description of an archaeological object also differs with respect to users, usages and time. Thus, archaeologists use semantic resources such as thesauruses to interoperate data from various origins.

Physical Model of Data Entities. The implementation of ArchaeoDAL heavily relies on the Apache ecosystem. In particular, its metadata system rests on the Atlas[4] data governance and metadata framework. Atlas' objects match with goldMEDAL's data entities. In addition to metadata properties (in the form of key-value pairs), objects may also relate to terms from thesauruses, i.e., goldMEDAL links, and classifications, i.e., goldMEDAL groupings (Figure 5).

Moreover, we exploit Atlas' object types to fulfill domain-specific requirements regarding metadata properties. For example, in the HyperThesau project, users need not only semantic metadata to understand data contents, but also geographical metadata to know where archaeological objects were discovered. The benefits of having an object type system include:

- consistency: a universal definition of metadata can avoid terminological variations that may cause data retrieval problems;
- flexibility: a domain-specific type system helps define specific metadata for requirements in each use case;
- efficiency: with a given metadata type system, it is easy to write and implement search queries. Because names and types of all metadata properties are known in advance, we can filter data with metadata predicates such as *upload_date* > '10/02/2016'.

Physical Model of Processes. Atlas also includes a nice lineage feature that helps visualize chains of processes. For instance, Figure 6 represents a simple ingestion process of raw data stored in HDFS into a Hive table, where objects are symbolized by blue hexagons and the process by a green hexagon.

[3]https://www.elastic.co

[4]https://atlas.apache.org

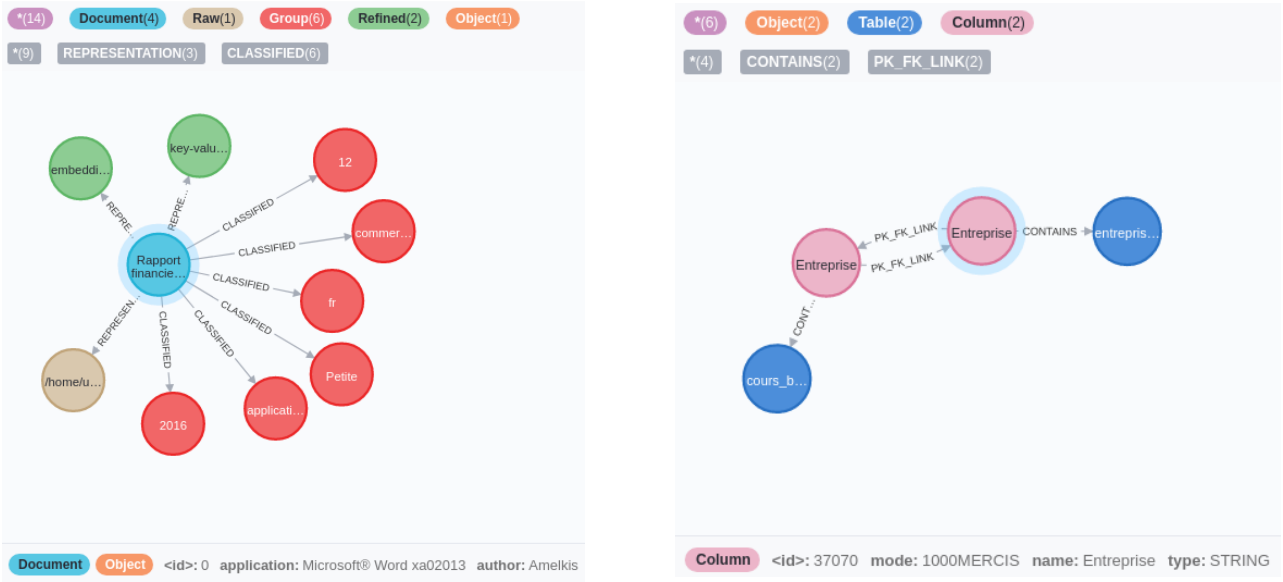

Figure 4: AUDAL sample Neo4J metadata

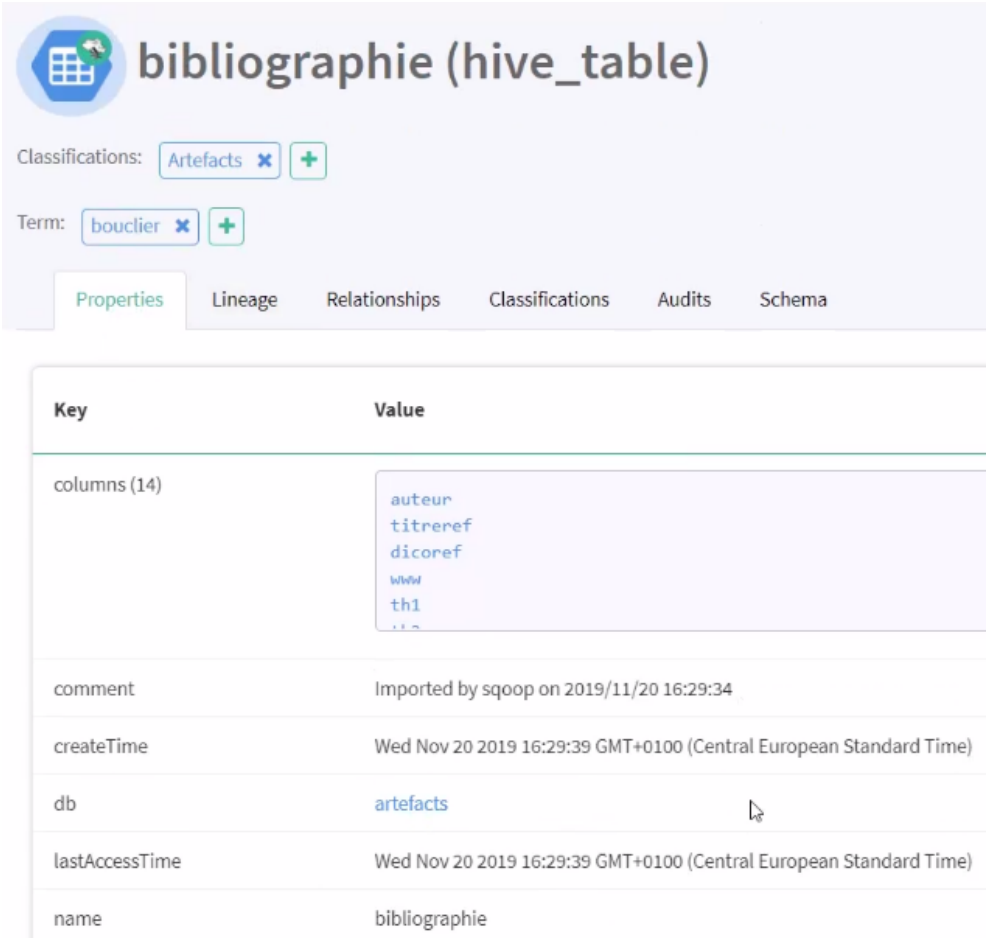

Figure 5: Sample Atlas object

Thesauruses and Links. The HyperThesau project heavily relies on thesauruses to organize data. A thesaurus consists of a set of categories and terms that help regroup data. In Atlas' glossary, a category may have only one parent. A category without a parent is called the root category. Conversely, a category may have several subcategories or terms. A term must have a parent

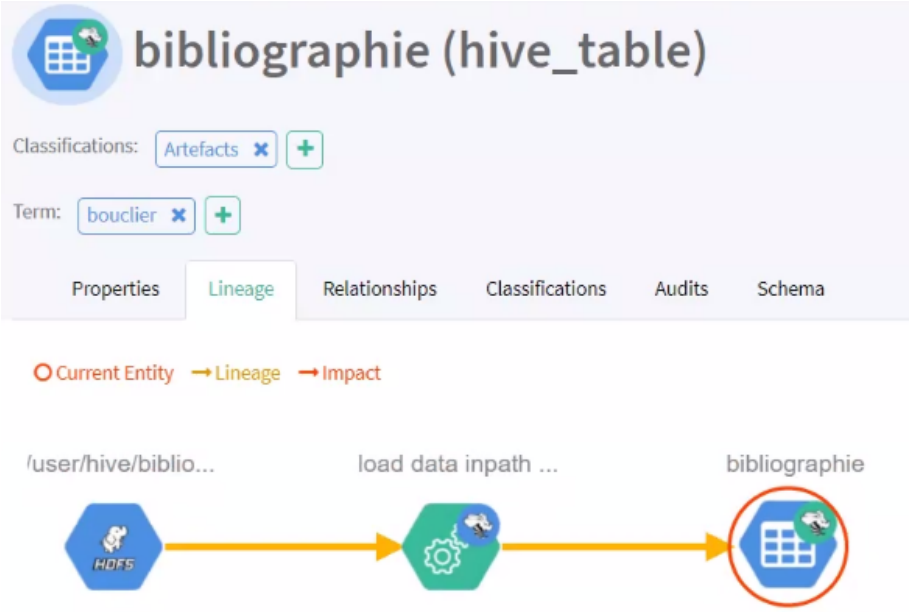

Figure 6: Sample Atlas lineage

category but no subcategory. A term may have relationships (i.e., goldMEDAL links) with other terms, e.g., related words, synonyms, antonyms, etc. Note that it would be easy to represent ontologies or taxonomies, too.

Eventually, we add specific links between data nodes associated with term nodes from the thesaurus. The left-hand side of Figure 7 displays an excerpt of the thesaurus. Figure 7 also shows how a term (*arme défensive*, i.e., defensive weapon) points to the corresponding metadata (short and long descriptions) and related terms.

5 CONCLUSION

In this paper, we introduced goldMEDAL, a generic data lake metadata model. goldMEDAL is based on four main concepts: data entity, grouping, link and process, which are defined at the conceptual and logical levels. These concepts interact altogether to support data lake metadata management requirements and they generalize almost all the concepts proposed in state-of-the-art metadata models : the concept of grouping supports the organization of data lakes in zones [18]; groupings allow managing multiple data granularity levels as in HANDLE [5].

Moreover, goldMEDAL supports all the features identified to compare data lake metadata models (Section 2.2), making it the most generic metadata model to the best of our knowledge.

Another particularity of goldMEDAL is the explicit possibility of data lineage tracing with the concept of process. goldMEDAL thus manages the dynamics of data, while the most recent metadata model from the literature, HANDLE [5], does not natively support it.

Eventually, we show as a proof of concept how goldMEDAL can be translated from conceptual and logical models to actual physical models with three different implementations of metadata models from distinct data lakes that feature both structured and unstructured data.

Future research and open issues include the "industrialization" of data lakes, i.e., providing a software layer, connected to the metadata system, which allows non-data or non-computer scientists to transform and analyze their own data in autonomy, just as dynamic reports are prepared on top of data warehouses for the use of business (i.e, non technical) users. However, such a software layer must not become yet another black box. In consequence, we must take great care of accompanying users in their appropriation of our analysis tools, not only by training, but also by interweaving research methodologies from computer science with business practices *by design*, in close collaboration with the partners.

Moreover, exploiting a data lake and its metadata system may contribute to open data and open science. A well-designed data lake should indeed readily enforce the four FAIR principles[5], i.e., findability, accessibility, interoperability and reusability. By adding an industrialization layer that allows non-data or non-computer scientist exploit the data lake, we can further improve accessibility *in a non-technical way*, i.e., not only through suitable communication protocols. FAIR principles are very appealing to researchers in humanities and social sciences, as illustrated by AUDAL (management sciences; Section 4.2.2) and ArchaeoDAL (archaeology; Section 4.2.3).

Finally, to the best of our knowledge, the maintenance of data lake metadata is a completely open issue. For instance, how to manage a new categorization of metadata? How to change or transform the metadata system when it hits some limits, whether technical or functional? What if metadata become big in the sense of voluminous big data? Should obsolete data be deleted, which is contrary to the principle of data lakes, and how to ensure that the metadata accessibility FAIR principle remains enforced when source data are no longer available?

ACKNOWLEDGEMENTS

E. Scholly's PhD is funded by BIAL-X[6]. P.N. Sawadogo's PhD is funded by the Auvergne-Rhône-Alpes Region through the

[5]https://www.go-fair.org/fair-principles/
[6]https://www.bial-x.com/

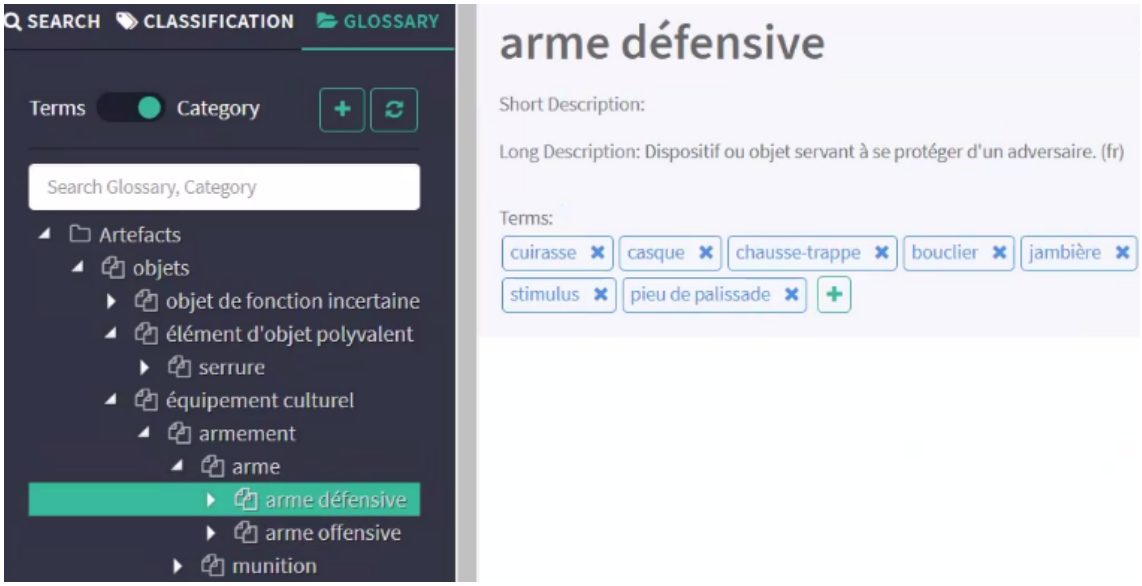

Figure 7: Sample Atlas thesaurus

AURA-PMI project. The HyperThesau project is funded by the Laboratory of Excellence "Intelligence of Urban Worlds" (IMU)[7].

REFERENCES

[1] Amin Beheshti, Boualem Benatallah, Reza Nouri, and Alireza Tabebordbar. 2018. CoreKG: A Knowledge Lake Service. *Proceedings of the Very Large Data Base Endowment (VLDB 2018)* 11, 12 (August 2018), 1942–1945.

[2] Jérôme Darmont, Cecile Favre, Sabine Loudcher, and Camille Noûs. 2020. Data Lakes for Digital Humanities. In *2nd International Digital Tools & Uses Congress (DTUC 2020), Hammamet, Tunisia*. ACM, New York, 38–41.

[3] Claudia Diamantini, Paolo Lo Giudice, Lorenzo Musarella, Domenico Potena, Emanuele Storti, and Domenico Ursino. 2018. A New Metadata Model to Uniformly Handle Heterogeneous Data Lake Sources. In *European Conference on Advances in Databases and Information Systems (ADBIS 2018), Budapest, Hungary*. 165–177.

[4] James Dixon. 2010. Pentaho, Hadoop, and Data Lakes. https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/.

[5] Rebecca Eichler, Corinna Giebler, Christoph Gröger, Holger Schwarz, and Bernhard Mitschang. 2020. HANDLE-A Generic Metadata Model for Data Lakes. In *International Conference on Big Data Analytics and Knowledge Discovery (DaWak 2020), Bratislava, Slovakia*. 73–88.

[6] Ashley Farrugia, Rob Claxton, and Simon Thompson. 2016. Towards Social Network Analytics for Understanding and Managing Enterprise Data Lakes. In *Advances in Social Networks Analysis and Mining (ASONAM 2016), San Francisco, CA, USA (IEEE)*. 1213–1220.

[7] Corinna Giebler, Christoph Gröger, Eva Hoos, Holger Schwarz, and Bernhard Mitschang. 2019. Leveraging the Data Lake - Current State and Challenges. In *International Conference on Big Data Analytics and Knowledge Discovery (DaWaK 2019), Linz, Austria*.

[8] Rihan Hai, Sandra Geisler, and Christoph Quix. 2016. Constance: An Intelligent Data Lake System. In *International Conference on Management of Data (SIGMOD 2016), San Francisco, CA, USA (ACM Digital Library)*. 2097–2100.

[9] Joseph M. Hellerstein, Vikram Sreekanti, Joseph E. Gonzalez, James Dalton, Akon Dey, Sreyashi Nag, Krishna Ramachandran, Sudhanshu Arora, Arka Bhattacharyya, Shirshanka Das, Mark Donsky, Gabriel Fierro, Chang She, Carl Steinbach, Venkat Subramanian, and Eric Sun. 2017. Ground: A Data Context Service. In *Biennial Conference on Innovative Data Systems Research (CIDR 2017), Chaminade, CA, USA*.

[10] Bill Inmon. 2016. *Data Lake Architecture: Designing the Data Lake and avoiding the garbage dump*. Technics Publications.

[11] Pwint Phyu Khine and Zhao Shun Wang. 2017. Data Lake: A New Ideology in Big Data Era. In *International Conference on Wireless Communication and Sensor Network (WCSN 2017), Wuhan, China (ITM Web of Conferences)*, Vol. 17. 1–6.

[12] Pengfei Liu, Sabine Loudcher, Jérôme Darmont, Emmanuelle Perrin, Jean-Pierre Girard, and Marie-Odile Rousset. 2020. Metadata model for an archeological data lake. *Digital Humanities Conference (DH 2020), Ottawa, Canada*.

[13] Cedrine Madera and Anne Laurent. 2016. The next information architecture evolution: the data lake wave. In *International Conference on Management of Digital EcoSystems (MEDES 2016), Biarritz, France*. 174–180.

[14] Hassan Mehmood, Ekaterina Gilman, Marta Cortes, Panos Kostakos, Andrew Byrne, Katerina Valta, Stavros Tekes, and Jukka Riekki. 2019. Implementing Big Data Lake for Heterogeneous Data Sources. In *International Conference on Data Engineering Workshops (ICDEW 2019), Macau SAR, China (IEEE)*. 37–44.

[15] Natalia Miloslavskaya and Alexander Tolstoy. 2016. Big Data, Fast Data and Data Lake Concepts. In *International Conference on Biologically Inspired Cognitive Architectures (BICA 2016), NY, USA (Procedia Computer Science)*, Vol. 88. 1–6.

[16] Christoph Quix and Rihan Hai. 2018. Data Lake. *Encyclopedia of Big Data Technologies* (2018), 1–8.

[17] Christoph Quix, Rihan Hai, and Ivan Vatov. 2016. Metadata Extraction and Management in Data Lakes With GEMMS. *Complex Systems Informatics and Modeling Quarterly* 9 (December 2016), 289–293.

[18] Franck Ravat and Yan Zhao. 2019. Metadata management for data lakes. In *European Conference on Advances in Databases and Information Systems (ADBIS 2019), Bled, Slovenia*. Springer, 37–44.

[19] Pegdwendé Sawadogo and Jérôme Darmont. 2021. On data lake architectures and metadata management. *Journal of Intelligent Information Systems* 56, 1 (2021), 97–120.

[20] Pegdwendé N Sawadogo, Etienne Scholly, Cécile Favre, Eric Ferey, Sabine Loudcher, and Jérôme Darmont. 2019. Metadata systems for data lakes: models and features. In *International Workshop on BI and Big Data Applications (BBIGAP@ADBIS 2019), Bled, Slovenia*. Springer, 440–451.

[21] Étienne Scholly. 2019. Business Intelligence & Analytics Applied to Public Housing. In *ADBIS Doctoral Consortium (DC@ADBIS 2019), Bled, Slovenia*. Springer, 552–557.

[22] Isuru Suriarachchi and Beth Plale. 2016. Crossing Analytics Systems: A Case for Integrated Provenance in Data Lakes. In *International Conference on e-Science (e-Science 2016), Baltimore, MD, USA (IEEE)*. 349–354.

[7] https://imu.universite-lyon.fr