Persistent Identifiers and Their Limitations in a Dynamic Web

Marcos Da Silveira[1,*,†], Cédric Pruski[1,†]

[1]*Luxembourg Institute of Science and Technology,*
5 avenue des hauts-fourneaux, L-4362 Esch-sur-Alzette, Luxembourg

Abstract

The trend in the internet is to publish more data enriched with semantic descriptions (e.g., FAIR metadata) and contextual links (e.g., Knowledge graphs), evolving from connected facts to a complete knowledge about digital objects. Persistent identifiers (PID) play an important role in this environment by providing long-lasting and unique identifiers for these digital objects, enabling FAIRness, and facilitating disambiguation and connection between them. However, PIDs are static while digital objects can be dynamic, which constraints the scenario where PIDs can be adopted without potentially generating misinterpretation of the status of the digital objects. In this paper, we analyze the limitations of existing PID approaches and we discuss the needs for dealing with dynamic environments. This discussion drove our proposal of the Persistent and Time-unique Identifier (PTID), that adds temporal information into PIDs and has an hierarchical structure to store versions of dynamic digital objects. We also present the impact of using PTID to the maintenance process of knowledge graphs.

Keywords

PID, Knowledge graph, Digital Objects, FAIR

1. Introduction

The unambiguous identification of digital objects (DO) is of utmost importance for data and knowledge intensive tasks and the identifier is a fundamental component for a metadata to FAIRly describe a DO. For this reason, academia and industry are joining forces to create rules and infrastructures to better and persistently identify DOs, to publish them in registries, and to promote the reuse of these PIDs to reference DOs. The role of a PID registry is to keep findable and accessible information about a DO, including the URL address where this object can be found. There are several schemes and technologies to build and store identifiers [1], but the management of their evolution and the guarantee that their format and properties can be maintained over time is still an open issue. This problem has been the focus of discussions of communities investigating the interoperability between PID systems, such as the Research Data Alliance (RDA) work groups on Aligning and coordinating national PID strategies, Open

SEMANTICS 2022 EU: 18th International Conference on Semantic Systems, September 13-15, 2022, Vienna, Austria
*Corresponding author.
†These authors contributed equally.
marcos.dasilveira@list.lu (M. D. Silveira); cedric.pruski@list.lu (C. Pruski)
https://marcao02.github.io/ (M. D. Silveira); https://www.researchgate.net/profile/Cedric-Pruski (C. Pruski)
0000-0002-2604-3645 (M. D. Silveira); 0000-0002-2103-0431 (C. Pruski)

Science Graphs Interest Group, Data Description Registry Interoperability (DDRI), Scholarly Link Exchange (Scholix), etc.

With the advance of AI technologies and the adoption of FAIR principles, DO are becoming more complex and dynamic resources having several dynamic metadata describing them, impacting the description and identification of the whole DO. We focus our work on these complex and dynamic DO, and our analysis takes into account several perspectives of DOs, such as when a change occurred (i.e., time-stamps), how the change was detected (i.e., what data changed inside of the new version of the DO), how the change was characterized (e.g., the evolutionary relationships created to link the versions as well as the taxonomy used to label this link), and how to refer to the different status of the DO (i.e., to specific versions of the DO). Our case study is the maintenance process of knowledge graphs (KGs), thus, we are also interested on how to extend existing strategies for PID creation in order to facilitate our analysis and to propose better maintenance process for KGs.

Referencing a DO is not a new problem and several solution exist. The Digital Object Identifier (DOI) [1] is the most widely used one. However, a DOI is not adapted for DOs that have versions because: 1) it does not allow to link the versions; 2-) it does not indicate what is the current version of a DO. In fact, DOI is advised for DO that are immutable over time. Extensions to DOI were proposed, such as in Zenodo [2], which proposes an hierarchical structure composed of "Version DOIs" and "Concept DOIs". The former identifies specific versions while the latter identifies a collection of DOIs. The Memento protocol [2] is another option to store version of DOs in the Web. Each copy of the DO is stored with a timestamp and they share the same URL. Searching for a version requires from users to provide the date and the URL that s/he is looking for. The limitation of these two approaches is that they store the whole changed DO in each version and there is no information about what changed or how. A traditional versioning control system, like Git [3], provides a transparent history of changes, but the versions' identifiers are not persistent. PIDs like the one proposed by ORCID iD[4] give more flexibility to the author of the content, allowing updating the information about a person without requiring a new PID for this person. However, it does not allow referencing to a specific version of the person description.

For the problem of knowledge graph maintenance, we need a solution that combines the advantages of each of these methods (the hierarchical referencing structure, temporal information, the changes details, and the flexible content management). Building/updating knowledge graphs of DOs requires to establish links between DOs' metadata (including the metadata of each version) and it is frequently done by linking DOs' PIDs [3]. In this paper, we focus on the following requirements for maintaining a KG: (1) graphs may require a PID that represents all versions of the DO, (2) graphs may require specific versions of the DO, (3) DOs can be described by many properties that change frequently, (4) graphs may include information about types of changes between DOs' versions, (5) temporal information can be used to refine the graph's nodes or relationships.

In the next section, we will introduce the persistent and time-unique ID (PTID) and we will

[1] https://www.doi.org
[2] https://help.zenodo.org/#versioning
[3] https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control
[4] https://orcid.org

discuss how PTIDs can be used to facilitate the maintenance of knowledge graphs.

2. Persistent and Time-unique Identifier

Identifiers can have embedded information (also called "intelligent" identifiers) or not ("dumb" identifiers). Good practices advise to avoid embedding information in the PID and only use dumb identifiers. However, in some cases embedded information in the PID are necessary [2], for instance, time-stamps for memento protocol.

Our approach only use dumb identifiers and split the identifier in two parts: one for the collective entity and one for the version entity. This approach is named Persistent and Time-unique IDentifier (PTID) and it regroups the versioning, hierarchical, and distributed methods. The collective entity represents the abstract (and stable) part of the DO while individual versions are represented as "part_of" of the collective entity, and have the dynamic set of information. In our approach, the temporal information is added to the node (as an attribute) while the relationships do not have temporal information, but exist only during the period of validity of both (source and target) nodes. For instance, the analysis of the MeSH ontology (Medical Subject Headings) version 2009 and version 2010 shows that the concept number D001411 changed its label from *Bacillus stearothermophilus* into *Geobacillus stearothermophilus*. In the same period, new concepts were added to provide more precision on the definition of Bacillaceae. The changes are illustrated in figure 1 as an evolutionary relationship between concepts. Notice that, for readability reasons, we only add to the figure one specialization of the concept Bacillaceae.

If this scenario (figure 1) was represented by other cited approaches, we will notice that the Zenodo approach will provide a PID to each version of MeSH, the memento protocol will provide a PID to each versions of the concept D001411 (as they have a URL assigned, for instance, http://id.nlm.nih.gov/mesh/D001411), the GIT system will track the changes in the MeSH files (e.g., change label), and the ORCID ID approach will allow changing the label without changing the PID. None of them will be able to completely represent this scenario. Looking deeper into the problem, if we want to build a historical knowledge graph [4], we must avoid the situation where two different concepts (i.e., the two versions of the concept D001411) have the same identifier. However, this approach should also allow searching for the concept D001411 or for one of its versions. To address this problem, we propose to have a collective entity that has the ID D001411 and contains the stable description of the concept (i.e., RDF Unique Identifier http://id.nlm.nih.gov/mesh/D001411, Scope Note "A species of GRAM-POSITIVE ENDOSPORE-FORMING BACTERIA in the family BACILLACEAE, found in soil, hot springs, Arctic waters, ocean sediments, and spoiled food products", etc.) and have two different entities (versions) that contain the dynamic part of the concept (i.e., the labels). When querying for the concept D001411, the obtained answer will be all information of the collective entity plus the information of the current version of the entity (i.e., PTID:000-0328). However, if the query specifies the version date (e.g., 2005) or the version number (e.g., PTID:0000-0316) then the answer will be all information of the collective entity plus the label within the required version (or still valid in the specified date). Figure 1 illustrates this approach. The squares represent the the static set of information and the circles represent the dynamic one. The two different versions of the concept D001411 are linked through the evolutionary link "change_label" that indicates what

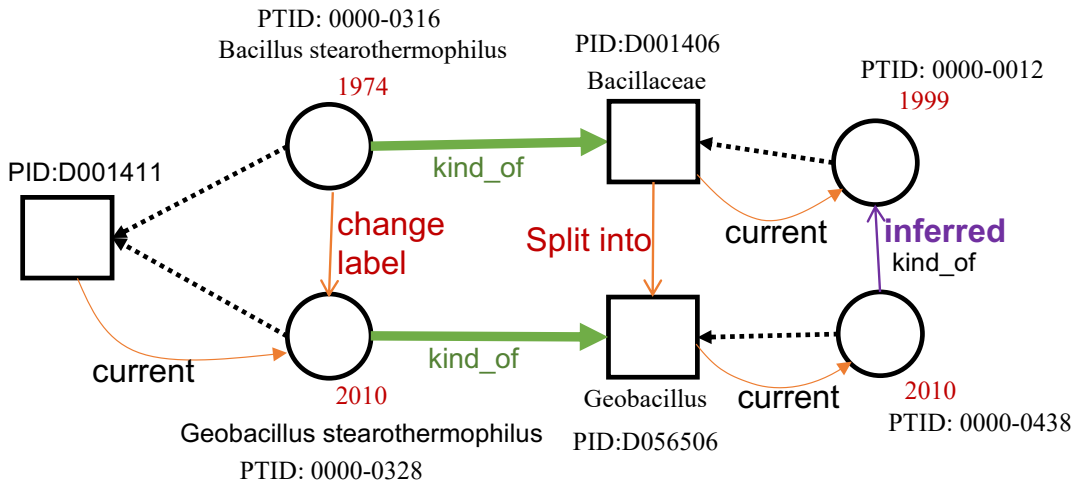

Figure 1: Example showing the use of PTID

kind of change was implemented between these versions. Other types of evolutionary links are also allowed. For instance, in 2010 the concept Bacillaceae was split into a more precise concepts (e.g., Geobacillus). Notice that the version entity of these concepts only have the creation date (by default this is a dynamic information) as properties, all other properties are static information and are part of the collective entity.

This approach avoids the duplication of stable information, but requires a clear definition of what is stable or dynamic information in the graph. The advantage of using KG to represent PTIDs is that each DO can use different criteria to define stable and dynamic information, and these criteria can change over time without affecting the other DOs. However, all versions of the same DO must respect the same criteria.

In brief, we showed that our PTID approach can be used when a change in a DO results into the creation of a new DO and when it does not. The PIDs are different for each DO's version, removing ambiguities and giving to users the possibility to refer to the collective entity or to each version of the DO (ensuring findability and accessibility to historical data). The DO can be described by several properties (including time stamps) and these properties are duplicated only if they are dynamic ones (saving storage spaces). Having the time stamps also give the possibility to query for information based on a validity period or on a precise date. Finally, the type of changes can be added to the properties of the DOs' versions as evolutionary links.

We also highlight that DOs can be created long time before having a PID. For instance, the concept "Bacillaceae" was created in 1975, but it was added to MeSH (and got a concept number) in 1999. Thus, KG builder can choose to add a snapshot of a DO from a specific date (different from the creation date). To record this information, we added the possibility to have a timestamp as a property of the node for the versions of the concept. The evolutionary relationships are important information for the maintenance process of a KG. We can use it to infer or to predict links between other concepts. For instance, if a collective entity was split into new specialized entities, thus we can infer the "kind of" relationship between the versions of these entities

(as shown in figure 1). In other words, if we search for patients with Bacillaceae, we can also add in the answer the patients that have Geobacillus. This example of analysis is part of our ongoing work on KG maintenance [4, 5], and from where the problem of identifiers raised up. The maintenance process that we will use PTID is composed of three main steps:

1. Compute the changes between versions. For that, we developed DynDiff [6], a method to compare two versions of an ontology or a KG, detect the changed elements, and combine them to identify more complex changes (e.g., move, merge, etc.).
2. Characterize the changes and label them according to a public accessible ontology. The terms used to label the evolutionary links come from the DynDiffOnto [6]. For instance, in the figure 1, to improve the readability of the figure, we labeled a evolutionary link as "change attribute", but, in reality, this link is named "changeOtherA", and the semantic description of this link can be found in DynDiffOnto.
3. Update the impacted DOs. We apply a pattern matching approach that was developed for a specific domain. Each pattern is associated with a set of rules that are applied to update the KG. The detailed description of these patterns and rules are out of the scope of this paper. However, to illustrate what kind of rules we apply, we textualized two rules: "a relationship is only duplicated if the source and target nodes are valid at the moment of the creation of the DO version", "the new version of a DO has the same parent as the old version of the DO", etc.

PTID was designed to reduce the number of manual actions. To illustrate the impact of PTID on our approach, let's assume that changes can only occur in the relationships or in the nodes of the KG. As a general rule, we adopt that a relationship has as origin and target collective entities (i.e., this relationship is valid for all versions of the DO). If this initial assumption is not true anymore (e.g., in 2010 the concept D001411 is not a "kind of" Bacillaceae anymore, it becomes a "kind of" Geobacillus). Thus, the relationship becomes a dynamic information and will link the versions of the concept D001411 to their target. The following triples replace the deleted relationship (D001411:PTID: 0000-0316, *kind_of*, D001406) and (D001411:PTID: 0000-0328, *kind_of*, D056506). The KG with this illustrative example is presented in figure 1, where the final version of the relationships are in green.

For changes in the nodes (e.g., in the label), we need to create a new version of the modified concept and copy all relationships from the previous version into the new one.

3. Conclusions

In this paper we analysed the limitations of using existing PIDs in dynamic environments such as the Web. This analysis focuses on environments where the information is represented through knowledge graphs and there is an intention of the KG manager to provide access to previous versions of the KG. Keeping snapshots of KGs stored in different datasets is not scalable, increase the time to find the information, and can be very expensive to store the data. Approaches that add hierarchical and/or temporal information to the PID are necessary to work with dynamic DOs, but to support the maintenance proceess of KGs we also need to have information about the changes. We proposed the persistent and time-unique identifier

(PTID) approach that split the identifier of a DO into two components: one to identify the stable part of the DO and another to identify the dynamic part. We discuss how this approach contributes to overcome the limitations of existing PIDs and we illustrate how PTIDs supports the implementation of KG maintenance tasks.

References

[1] M. Hellström, M. Johnsson, A. Vermeulen, Identification and citation of digital research resources, in: Towards Interoperable Research Infrastructures for Environmental and Earth Sciences, Springer, 2020, pp. 162–175.

[2] L. Gleim, L. Tirpitz, S. Decker, Http extensions for the management of highly dynamic data resources, in: European Semantic Web Conference, Springer, 2021, pp. 212–229.

[3] X. Chen, S. Dallmeier-Tiessen, R. Dasler, S. Feger, P. Fokianos, J. B. Gonzalez, H. Hirvonsalo, D. Kousidis, A. Lavasa, S. Mele, et al., Open is not enough, Nature Physics 15 (2019) 113–119.

[4] S. D. Cardoso, M. Da Silveira, C. Pruski, Construction and exploitation of an historical knowledge graph to deal with the evolution of ontologies, Knowledge-Based Systems 194 (2020) 105508. URL: https://doi.org/10.1016/j.knosys.2020.105508. doi:10.1016/j.knosys.2020.105508.

[5] S. D. Cardoso, M. Da Silveira, C. Pruski, C. Reynaud-Delaitre, Combining rules, background knowledge and change patterns to maintain semantic annotations, in: AMIA Annual Symposium proceedings, 2018, p. 505–514.

[6] S. Diaz Benavides, S. D. Cardoso, M. Da Silveira, C. Pruski, Dyndiff: A tool for comparing versions of large ontologies, in: Proceedings of SeWebMeDa workshop at ESWC conference, 2022.