Building Linked Open Date Entities for Historical Research

Go Sugimoto

Austrian Centre for Digital Humanities and Cultural Heritage, Austrian Academy of Sciences, Sonnenfelsgasse 19, Vienna, Austria
go.sugimoto@oeaw.ac.at

Abstract. Time is a focal point for historical research. Although existing Linked Open Data (LOD) resources hold time entities, they are often limited to modern period and year-month precision at most. Therefore, researchers are currently unable to execute co-reference resolution through entity linking to integrate different datasets which contain information on the day level or remote past. This paper aims to build an RDF model and lookup service for historical time at the lowest granularity level of a single day at a specific point in time, for the duration of 6000 years. The project, Linked Open Date Entities (LODE), generates stable URIs for over 2.2 million entities, which include essential information and links to other LOD resources. The value of date entities is discussed in a couple of use cases with existing datasets. LODE facilitates improved access and connectivity to unlock the potential for the data integration in interdisciplinary research.

Keywords: Historical date · Entity model · URI · Lookup · Time Ontology in OWL

1 Introduction

Time is one of the most fundamental concepts of our life. The data we deal with often contain time concepts such as day and year in the past, present, and future. There is no doubt that historical research cannot be done without notations of time. On the other hand, the advent of Linked Open Data (LOD) has changed the views on the possibility of data-driven historical research. Indeed, many projects have started producing a large number of LOD datasets. In this strand, entity linking has been considered as a critical ingredient of LOD implementation. Digital humanities and cultural heritage communities work on co-reference resolution by means of Named Entity Linking (NEL) to LOD resources with an expectation to make connections between their datasets and other resources [1–4]. It is often the case that they refer to globally known URIs of LOD such as Wikidata and DBpedia for the purpose of interoperability. Historical research datasets include such fundamental concepts as “World War I” (event), “Mozart” (person), “the Dead Sea Scrolls” (object), “the Colosseum” (building), and “Kyoto” (place). However, rather surprisingly, time concepts/entities are not fully discussed in this context. One reason is that the availability of LOD entities are limited to meet the needs of historians. Moreover,

© Springer Nature Switzerland AG 2021
E. Garoufallou and M.-A. Ovalle-Perandones (Eds.): MTSR 2020, CCIS 1355, pp. 323–335, 2021.
https://doi.org/10.1007/978-3-030-71903-6_30
they may not be well known and use cases are largely missing. It is also likely that entity linking is simply not executed. In the following sections, we discuss those issues and solutions in detail.

The primary goal of this paper is to foster LOD-based historical research by modelling and publishing time concepts/entities, called “Linked Open Date Entities (LODE)”, which satisfies the preliminary requirements of the target users. In particular, we 1) design and generate RDF entities to include useful information, 2) provide a lookup and API service to allow access to the entities through URI, 3) illustrate a typical implementation workflow for entity linking (“nodification”), and 4) present use cases with existing historical resources.

2 Related Work and Unsolved Issues

Firstly, we examine published temporal entities in LOD. In terms of descriptive entities, DBpedia holds entities including the 1980s, the Neolithic, the Roman Republic, and the Sui dynasty. PeriodO\(^1\) provides lookups and data dumps to facilitate the alignment of historical periods from different sources (the British Museum, ARIADNE, etc.). Semantics.gr has developed LOD vocabularies for both time and historical periods for SearchCulture.gr \(^5\). However, the lowest granularity is early, mid, and late period of a century. As descriptive time entities are already available, this article concentrates on numeric time entities that could connect to the descriptive ones. In this regard, DBpedia contains RDF nodes of numeric time such as 1969. They hold literals in various languages and links to other LOD resources, and can be looked up. However, year entities\(^2\) seem to be limited in the span between ca. 756 BC and ca. AD 2071, while years beyond this range tend to be redirected to the broader concepts of decade. Moreover, there seem to be no or only few entities for a month and day of a particular year. SPARQL queries on Wikidata suggest that the year entities are more or less continuously present between 2200 BC and AD 2200.\(^3\) Year-month entities seem to be merely available for a few hundred years in the modern period\(^4\), and day-level entities are scarce.\(^5\) Situations are normally worse in other LOD datasets.\(^6\) Therefore, it is currently not possible to connect datasets to the time entities comprehensively for a day and month, or a year in the remote past. This is not satisfactory for historical research. For instance, we could easily imagine how important time information could be in a situation in which the day-to-day reconstruction of history in 1918 during World War I is called for. The same goes for prehistory or medieval history, although lesser time precision would be required.

\(^1\) https://perio.do (accessed July 20, 2020).
\(^2\) See also https://en.wikipedia.org/wiki/List_of_years. (accessed July 20, 2020).
\(^3\) Currently the lower and upper limit would be 9564 BC and AD 3000.
\(^4\) A SPARQL query only returns 218 hits between AD 1 to AD 1600, while 5041 entities are found between AD 1600 and 2020.
\(^5\) A SPARQL query returns no hit before October 15 1582 (on the day when the Gregorian calendar was first adopted), and only returns 159691 hits between AD 1 to AD 2020.
\(^6\) For example, rare cases include https://babelnet.org/synset?word=bn:14549660n&details=1&lang=EN. (accessed July 20, 2020).
Secondly, we look for ontologies in order to represent temporal information in RDF. [6] study TimeML to annotate historical periods, but its XML focus is out of our scope. Time Ontology in OWL [7] reflects classical works of [7, 8] and [9], overcoming problems of the original OWL-Time [8] that defined instant (point of time) and interval (period of time), but limited itself to the Gregorian calendar [10]. Thus, the use of different temporal reference systems (e.g. the Jewish calendar, radiocarbon dating) for the same absolute point in time can be modelled nicely [11]. The specifications also state some advantages of their approach over a typed literal, echoing the vision of our proposal (Sect. 3.1). In the Wikidata ontology, two streams of temporal concepts are present. One is concepts for the unit of time, or time interval, including millennium, century, decade, year, month, and day. The other is considered as the instances of the former. For example, the second millennium is an instance of millennium, while August 1969 is an instance of month. In the field of historical research, CIDOC-CRM [9] has similarity to Time Ontology in OWL, defining temporal classes and properties influenced by [8]. [12] apply Time Ontology in OWL for the ancient Chinese time, which demonstrates the importance of developing ontologies for non-Gregorian calendars.

Thirdly, a few examples are found along the line of data enrichment and entity linking. During the data aggregation process of Europeana, data enrichment is performed [2]. Some Europeana datasets include enriched date information expressed via edm:TimeSpan in relation to a digital object. [10] It contains URIs from semium.org, labels, and translations. [11] Those URIs connect different resources in the Europeana data space. A time concept links to broader or narrower concepts of time through dcterms:isPartOf. Another case is Japan Search. [12] In its data model, schema:temporal and jps:temporal function as properties for time resources. [13] The SPARQL-based lookup service displays time entities such as https://jpsearch.go.jp/entity/time/1162 and https://jpsearch.go.jp/entity/time/1100-1199, which often contain literal values in Japanese, English, and gYear, as well as owl:sameAs links to Wikidata and Japanese DBpedia. The web interface enables users to traverse the graphs between time entities and cultural artifacts in the collection.

3 Implementing the Linked Open Date Entities

3.1 Notification

We shall now discuss why RDF nodes are beneficial. Time concepts in historical research datasets are normally stored as literal values, when encoded in XML or RDF. In fact, those

---

[7] https://www.w3.org/TR/owl-time/ (accessed July 20, 2020).
[8] https://www.w3.org/TR/2006/WD-owl-time-20060927/ (accessed July 20, 2020).
[9] https://www.cidoc-crm.org/ (accessed July 20, 2020).
[10] The DPLA Metadata Application Profile (MAP) also uses edm:TimeSpan (https://pro.dp.la/hubs/metadata-application-profile) (accessed July 20, 2020).
[11] See an example record at https://www.europeana.eu/portal/en/record/9200434/oai_baa_onb_at_8984183.html. For example, https://semium.org/time/1900 represents AD 1900. (accessed July 20, 2020).
[12] https://jpsearch.go.jp/ (accessed July 20, 2020).
[13] https://www.kanzaki.com/works/ld/jpsearch/primer/ (accessed July 20, 2020).
literals are often descriptive dates, such as “early 11th century”, “24 Aug 1965?”, “1876年”, and “1185 or 1192”, to allow multilingualism, diversity, flexibility, and uncertainty [5]. [6] report that less than half of dates in the ARIA database of Rijksmuseum are 3 or 4 digit year. Sometimes literal values are more structured and normalised like 1789/7/14. However, they could be only a fraction. The syntax of “standardised” dates also varies in different countries (10/26/85 or 26/10/85). The tradition of analogue data curation on historical materials may also contribute to this phenomenon to a certain extent. Whatever the reasons are, literals in RDF have three major disadvantages over nodes: a) new information cannot be attached, b) they are neither globally unique nor referable, and c) they cannot be linked. Since LOD is particularly suited to overcome those shortcomings, literals alone may hinder historical research in the LOD practices. This is the forefront motivation of the transformation of literals with or without data type into nodes/entities/resources. We may call it “nodification”. Figure 1 visualises a real example of nodification. ANNO14 and the Stefan Zweig dataset15 can be interlinked and the graph network is extended to other global LOD resources.

Fig. 1. Interlinking of nodes by nodification of two datasets

Some may still argue that nodification is redundant and/or problematic, because typed literals are designed for time expressions, and XMLSchema-based (e.g. xsd:date) calculations by queries cannot be done with nodes. But, this is not entirely true. First of all, the nodification of this project does not suggest a replacement of literals. When LOD datasets include typed literals, they can be untouched and/or fully preserved in rdfs:label of the new nodes. The temporal calculations are still fully supported, and are encouraged for mathematical operations. It is possible to use SPARQL to obtain not only dates in typed literals, but also dates without data types. It is also noted that the year entities in DBpedia do not seem to support data types for literals, thus arithmetic calculations may not be possible, while Wikidata does for the year, month and day entities.16 Secondly, as literals are intact, this proposal is a data enrichment and hence not a duplication. The enrichment provides additional possibilities to attach new information, which cannot be achieved by typed literals. Thirdly, a lookup service of LODE serves as a global and

---

14 https://anno.onb.ac.at/ (accessed July 20, 2020).
15 https://www.stefanzweig.digital (accessed July 20, 2020).
16 However, there would be a problem, because it sets January 1 as the value of xsd:dateTime for a time interval entity (e.g. 1987 is represented as 1987–01-01T00:00:00Z).
permanent reference point for the links across datasets. It encourages data owners to include the entity URIs in their datasets, so that they are able to connect to other datasets that refer to the same URIs. In addition, users often need data browsing before and/or without data querying, in order to understand the scope of data (e.g. data availability, coverage, connectivity, structure) by traversing graphs. Whilst the nodification offers an optimal use case for easy data exploration, literals have limited possibility. Lastly, without the nodification, LOD users have to connect datasets via date literals on the fly whenever they need to. Although it is possible to generate RDF nodes out of literals only when needed, URIs may not be assigned permanently for the nodes in this scenario. Therefore, long-term references are not assured. In addition, it is critical to openly declare and publish URIs a priori through the lookup. Otherwise, it is unlikely that NEL is conducted widely. In a way, the nodification also has a similar scope to materialisation with regard to pre-computing of data for performance. In Table 1, several advantages of our approach (preprocessed nodification) are outlined over 1) the use of only literals, and 2) on-demand nodification.

3.2 URI Syntax Principles

In order to execute the nodification, URIs are required. This section briefly highlights the design principles for the URI syntax of LODE. The date URIs consist of a base URI and a suffix. The base URI is set as https://vocabs.acdh.oeaw.ac.at/date/ as a part of an institutional vocabulary service, although it is misleading to be called vocabulary. As for the suffix, we follow the most widely accepted standard, ISO8601, which is the convention of many programming languages (SQL, PHP, Python) and web schemas (XMLSchema 1.1). The most common format should look like YYYY (2020), YYYY-MM (2020-01), and YYYY-MM-DD (2020-01-01). An important factor of adopting the subset of ISO8601 is that it can provide non-opaque numeric-based URIs. It enables human users to conjecture or infer any dates, including dates in a remote past and future, even if look ups are not available. In contrast, it is very hard for them to infer dates from opaque URIs such as the Wikidata URIs. ISO8601-based URIs are also language independent, as opposed to the DBpedia URIs. Those consideration helps researchers who deal with time spanning tens of thousands of years.

The use of ISO8601 also implies that the Gregorian calendar and proleptic Gregorian calendar are applied. The latter is the extension of the Gregorian calendar backward to the dates before AD 1582. Although ISO8601 allows it, the standard also suggests that there should be an explicit agreement between the data sender and the receiver about its use. Therefore, we provide a documentation to explain the modelling policy. In addition, the ISO8601 syntax is applied for BC/BCE and AD/CE, although there is complicated

---

17 Materialisation is the term used in Semantic Web to generate graphs based on inferences. Implicit knowledge is materialised in order to make it explicit for the purpose of query performance.
18 For instance, https://www.wikidata.org/entity/Q2432 represents AD 1984.
19 Full details are available at https://vocabs.acdh.oeaw.ac.at/date/ together with the syntax principles.
Table 1. Pros and cons of three approaches

|                                              | Preprocessed notification (our approach) | Only literals | On-demand notification |
|----------------------------------------------|------------------------------------------|---------------|------------------------|
| Connects to nodes in RDF                     | ☑                                        | ☒             | ☒                      |
| Includes literals                            | ☑                                        | ☑             | ☑                      |
| Possibility to add other information        | ☑                                        | ☒             | ☒                      |
| Time calculation by XSD typed literals       | ☑                                        | ☑             | ☑                      |
| Stable URIs                                  | ☑                                        | ☒             | ☒                      |
| Lookup service                               | ☑                                        | ☒             | ☒                      |
| Graph data browsing                          | Optimal                                  | Not optimal   | Not optimal            |
| Access and query performance                 | Depends (probably better)                | Depends       | Depends                |
| RDF data size (after enrichment)            | Bigger                                   | Smaller       | Smaller                |
| Data processing tasks for enrichment         | Much more<sup>b</sup>                    | No, or much less | No, or much less |

<sup>a</sup>Only on demand for selected datasets.

<sup>b</sup>This may not be a disadvantage, if one would like to execute other types of data enrichment and normalisation to improve data quality in parallel.

discussions and controversy.<sup>20</sup> Year Zero does not exist, thus “0000” means 1 BC<sup>21</sup> and “−0001” is 2 BC. More than 4 digits (“−13000”) allow time concepts in prehistory. As

<sup>20</sup>See [https://phabricator.wikimedia.org/T94064](https://phabricator.wikimedia.org/T94064). There are confusing specifications in XML Schema Part 2: Datatypes Second Edition ([https://www.w3.org/TR/xmlschema-2/#isoformats](https://www.w3.org/TR/xmlschema-2/#isoformats)), XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes ([https://www.w3.org/TR/xmlschema11-2/#dateTime](https://www.w3.org/TR/xmlschema11-2/#dateTime)), and the HTML living standard ([https://html.spec.whatwg.org/#global-dates-and-times](https://html.spec.whatwg.org/#global-dates-and-times)). (accessed July 20, 2020).

<sup>21</sup>As seen in Wikidata ([https://www.wikidata.org/entity/Q25299](https://www.wikidata.org/entity/Q25299)). (accessed July 20, 2020).
the syntax is the subset of ISO8601, exactly 3 digits (YYY) and 2 digits (YY) can be also used, representing a decade and century respectively.\textsuperscript{22}

In order to accommodate other calendars (e.g. Julian, Islamic) and dating systems (carbon-14 dating), one can add a schema name between the base URI and the date. For example, we could define URIs for the Japanese calendar as follows: https://vocabs.acdh.oeaw.ac.at/date/japanese_calendar/平成/31.

### 3.3 Modelling LODE in RDF

The first implementation of our RDF model should at least include entities at the lowest granularity level of a single day for the duration of 6000 years (from 3000 BC to AD 3000). From the perspectives of historians and archaeologists, day-level references would be required for this temporal range. The number implies that there will be over 2.2 million URIs, counting the units of the whole hierarchy from days to millennia. In any case, this experiment does not prevent us from extending the time span in the future.

Regarding the RDF representation of time entities, we adopt properties from Time Ontology in OWL, RDFS, and SKOS. However, there is a clear difference between LODE and Time Ontology in OWL. The former aims to create historical dates as stable nodes, rather than literals that the latter mostly employs. The latter also does not have properties expressing broader semantic concepts than years; decades, centuries, and millennia are not modelled by default. Therefore, we simply borrow some properties from the ontology for specific purposes, including \texttt{time:DayOfWeek}, \texttt{time:monthOfYear}, \texttt{time:hasTRS}, \texttt{time:intervalMeets}, and \texttt{time:intervalMetBy}. In LODE, the URLs of DBpedia, YAGO, Wikidata, semium.org, and Japan Search are included in our entities as the equivalent or related entities, where possible, especially for the entities of years and upward in hierarchy. Figure 2 illustrates a typical date entity for the day-level.

In order to generate 2.2 million entities, we have created dozens of Perl scripts, producing entities in RDF/XML for days, months, years, decades, centuries, and units of time, because of the complexity of generating the DBpedia and YAGO URIs as well as literal variations for different units of time. As there are only 6 millennia, they are manually created as the top level entities. The Perl library of DateTime\textsuperscript{23} is primarily used to calculate, for example, the day of a week, the day of a year, and the corresponding day of the Gregorian calendar in the Julian calendar. Some small functions are also developed to generate variations of descriptive dates in English and German and to calibrate entities for BC and AD as well as leap years.

The overall structure of various entities in LODE is visualised in Fig. 3. There were two choices to create links between the date entities. One is the SKOS vocabulary and the other is an ontology using RDFS/OWL. According to the SKOS Reference specifications\textsuperscript{24}, a thesaurus or classification scheme is different from a formal knowledge representation. Thus, facts and axioms could be modelled more suitably in an ontology.

---

\textsuperscript{22} For example, “196” means the 1960s, and “19” is the 19th century. They should not be confused with “0196” (AD 196) and “0019” (AD 19). Years less than 5 digits must be expressed in exactly 4 digits.

\textsuperscript{23} \url{https://metacpan.org/pod/DateTime} (accessed July 20, 2020).

\textsuperscript{24} \url{https://www.w3.org/TR/skos-reference/} (accessed July 20, 2020).
as the formal logic is required. The date entities seem to be facts and axioms, as we are dealing with commonly and internationally accepted ISO8601. However, from a historical and philosophical point of view, one could also argue that they are also heavily biased toward the idea of the Christian culture. Therefore, the decision to adopt SKOS or OWL was not as simple as it seemed. This paper primarily uses SKOS for two reasons: a) the implementation of a lookup service is provided by SKOSMOS which requires SKOS, b) it is preferred to avoid debates on the ontological conceptualisation of time for the time being. It is assumed that the even Wikidata ontology (Sect. 2) could be a subject of discussion. Moreover, there would be potential problems to use semantic reasoners, for example, due to the inconsistency of our use of decades and centuries.\footnote{For example, a popular usage is that a decade starts from a year ending in a 0 to a year ending in a 9, while a decade starts from a year ending in a 1 to a year ending in a 0 in a rarer version. Wikidata adopts the former, resulting the 0s and 0s BC consisting of only 9 years. A similar conflict of constructs exists for the use of century.} In this sense, SKOS is more desirable thanks to its simple structure and loose semantics.
Nevertheless, we adopted the same structure of the Wikidata ontology for interoperability reasons, by simply replacing all its proprietary properties with `skos:broad` and `skos:narrower` and a couple of Time Ontology in OWL. This builds a hierarchy of time concepts such as day, month, and year. Similarly, we separate the units of time from instances, with `rdf:type` connecting them. The sequence of the same time unit is encoded by `time:intervalMetBy` and `time:intervalMeets`.

A lookup service is implemented with SKOSMOS\(^\text{26}\) (Fig. 4). Once SKOS compliant RDF files are imported to a Jena Fuseki server, one can browse through a hierarchical view of the vocabulary and download an entity in RDF/XML, Turtle, and JSON-LD.

\(^{26}\) https://skosmos.org/ (accessed July 20, 2020).
4 Use Cases

One benefit of LODE is the capability of handling multilingualism and different calendars. In a use case of Itabi: Medieval Stone Monuments of Eastern Japan Database, one may like to align the Japanese calendar with the Western one, when expressing the temporal information in the dataset as LOD. A trouble is that most records hold the accurate date (i.e. day and month) in the Japanese calendar, and only the equivalent year in the Western calendar. Thus, while preserving the original data in literals, it would be constructive to use nodification and materialisation techniques to connect relevant date entities to the artifact (Fig. 5). LODE helps substantially in this scenario, because it allows us to discover the corresponding day both in the proleptic Gregorian calendar and the Julian calendar by inferences/materialisation, as well as the day of the week. The implementation is not possible yet, however, LODE plans to include mapping between the Japanese and Western calendar in the future. By extending this method, we could expect that LOD users can query LODE to fetch a literal in a specific language and use it for querying a full-text database that is not necessarily RDF-compliant, and does

Fig. 5. A record containing the Japanese and Western calendar (https://www.rekihaku.ac.jp/doc/itabi.html (accessed July 20, 2020)) (above) and an example of its simplified RDF model connecting date entities (below)

27 https://www.rekihaku.ac.jp/up-cgi/login.pl?p=param/ita2/db_param (accessed July 20, 2020).
not support the Western alphabet and/or calendars. Such a use case is not possible with literals alone.

A more typical pattern of nodification is data enrichment. The Omnipot project in our institute aims to create an extremely large knowledge graph by ingesting local and global LOD into one triple store. The project evaluates the connectivity of heterogeneous graphs through LODE and the usability of data discovery and exploration. During the nodification of 1.8 million literals in ANNO, not only dates but also data providers and media types are nodified. In this regard, the nodification is not a labour-intensive obstacle, but a part of a data improvement and NEL. A similar nodification is conducted for the Schnitzler-LOD datasets\textsuperscript{28} and PMB\textsuperscript{29} by Regular Expression. This practice verifies our approach with human inferable non-opaque URIs. Unlike the Wikidata URIs, the LODE and DBpedia URIs were embedded with little effort. The simplicity of implementation incentivises data owners to notify their data in the future.

Research Space\textsuperscript{30} displays incoming and outgoing node links in the Omnipot project automatically. Figure 6 showcases connections between them, via the 1987 entity. Users could interactively compare art objects in Europeana with art works in Wikipedia from the same year via Wikidata.\textsuperscript{31} This view is currently not possible with literals alone. By default many visualisation software offer a graph view enabling users to focus on nodes as a mean to traverse graphs. Therefore, they do not have to worry about query formulations. As it is not trivial to construct the same view by a query using literals, user friendliness should be considered as a selling point of nodification.

\textbf{Fig. 6.} Interactive LOD exploration through node links in Research Space

\textsuperscript{28} https://schnitzler-lod.acdh-dev.oeaw.ac.at/about.html (accessed July 20, 2020).

\textsuperscript{29} https://pmb.acdh.oeaw.ac.at/ (accessed July 20, 2020).

\textsuperscript{30} https://www.researchspace.org/ (accessed July 20, 2020).

\textsuperscript{31} As Wikipedia is not LOD, only links to Wikipedia articles are shown and clickable.
5 Future Work and Conclusion

Future work would be more case studies which the use of literals alone cannot easily replicate. An RDF implementation of TEI\(^{32}\) could bring interesting use cases by normalising and nodifying date literals in various languages, scripts, and calendars in historical texts. In addition, LODE could align with the Chinese, Islamic, Japanese, and Maya calendars, and add more information about festivities and holidays that literals cannot fully cover. Consequently, event-based analyses by SPARQL may uncover unknown connections between people, objects, concepts, and places in a global scale. It could even connect to pre-computed astronomical events at various key locations such as the visibility of planets on a specific day, with which interdisciplinary research can be performed. Further, detailed evaluation of use cases is also needed. For instance, query performance and formulation, and usability could be measured and analysed more systematically. We are also fine-tuning the LODE model by properly modelling the concepts of instants and intervals based on the Time Ontology in OWL.

LODE attempts to solve two problems of existing LOD: a) It tries to meet the needs for greater coverage and granularity of date entities for historical research. b) By designing a simple model and suggesting a straightforward method of nodification, it helps to reduce the complexity of LOD by automatically connecting/visualising vital information in a big Web of Data. Although the research focused exclusively on cultural heritage, many science domains deal with some conception of time, and thus, this study could be an impetus to acknowledge the necessity and the impact of time entities in a broader research community. Since time is one of the most critical dimensions of datasets, we would be able to unlock more potential of LOD.

References

1. van Veen, T., Lonij, J., Faber, W.J.: Linking named entities in Dutch historical newspapers. In: Garoufallou, E., Subirats Coll, I., Stellato, A., Greenberg, J. (eds.) MTSR 2016. CCIS, vol. 672, pp. 205–210. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49157-8_18
2. Stiller, J., Petras, V., Gäde, M., Isaac, A.: Automatic enrichments with controlled vocabularies in Europeana: challenges and consequences. In: Ioannides, M., Magnenat-Thalmann, N., Fink, E., Žarnič, R., Yen, A.-Y., Quak, E. (eds.) EuroMed 2014. LNCS, vol. 8740, pp. 238–247. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13695-0_23
3. van Nispen, A.: EHRI Vocabularies and Linked Open Data: An Enrichment? ABB: Archives et Bibliothèques de Belgique - Archief- en Bibliotheekwezen in België 106, 117–122 (2019)
4. Zeng, M.L.: Semantic enrichment for enhancing LAM data and supporting digital humanities. Rev. Article. EPI 28 (2019). https://doi.org/10.3145/epi.2019.ene.03.
5. Georgiadis, H., Papanoti, A., Paschou, M., Roubani, D., Hardouveli, D., Sachini, E.: The semantic enrichment strategy for types, chronologies and historical periods in searchculture.gr. In: Garoufallou, E., Virkus, S., Siatri, R., Koutsomiha, D. (eds.) MTSR 2017. CCIS, vol. 755, pp. 211–223. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70863-8_20
6. de Boer, V., van Someren, M., Wielinga, B.J.: Extracting historical time periods from the Web. J. Am. Soc. Inf. Sci. 61, 1888–1908 (2010). https://doi.org/10.1002/asi.21378

\(^{32}\)https://tei-c.org/ (accessed July 20, 2020).
7. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM. 26, 832–843 (1983). https://doi.org/10.1145/182.358434
8. Allen, J.F., Ferguson, G.: Actions and Events in Interval Temporal Logic. In: Stock, O. (ed.) Spatial and Temporal Reasoning. pp. 205–245. Springer Netherlands, Dordrecht (1997). https://doi.org/10.1007/978-0-585-28322-7_7.
9. Hobbs, J.R., Pan, F.: An ontology of time for the semantic web. ACM Trans. Asian Lang. Inf. Process. (TALIP). 3, 66–85 (2004). https://doi.org/10.1145/1017068.1017073
10. Cox, S., Little, C., Hobbs, J., Pan, F.: Time ontology in OWL (2017). https://www.w3.org/TR/2017/REC-owl-time-20171019/
11. Cox, S.J.D.: Time ontology extended for non-Gregorian calendar applications. SW. 7, 201–209 (2016). https://doi.org/10.3233/SW-150187
12. Zou, Q., Park, E.G.: Modelling ancient Chinese time ontology. J. Inf. Sci. 37, 332–341 (2011). https://doi.org/10.1177/0165551511406063