Semantic Web 0 (2021) 1–0
IOS Press

Temporal Knowledge Representation for Historical Corpora: Application to the Henri Poincaré Correspondence Corpus

Nicolas Lasolle [a,b,*] Olivier Bruneau [a], Jean Lieber [b], Laurent Rollet [a]
[a] *AHP-PReST, Université de Lorraine, CNRS, Université de Strasbourg, France*
E-mails: nicolas.lasolle@univ-lorraine.fr, olivier.bruneau@univ-lorraine.fr, laurent.rollet@univ-lorraine.fr
[b] *LORIA, Université de Lorraine, CNRS, Inria, France*
E-mails: nicolas.lasolle@univ-lorraine.fr, jean.lieber@loria.fr

Abstract. The study of Henri Poincaré's life and works (1854-1912) has led to the constitution of a corpus which includes various document types (articles, books, reports, letters, etc.). There is a keen interest in his correspondence, which gathers scientific, administrative and private exchanges. Semantic Web technologies have been chosen to represent and to exploit corpus data: the RDF model, the RDFS knowledge representation language and the SPARQL query language. The existing representation does not include temporal knowledge, yet it is an important aspect when studying historical corpora. Different methods have been proposed by the Semantic Web community to include temporal knowledge in RDF databases. This article presents the temporal knowledge representation issues encountered in the exploitation of the Henri Poincaré correspondence corpus. Through examples, it compares existing methods and ontologies and details the implementation of the approach chosen for this corpus. A form-based interface has been developed to assist users in corpus data querying. The system relies on the use of a transformation rule mechanism to address two issues faced when dealing with temporal graphs: simplifying SPARQL query generation and reasoning with temporal data.

Keywords: Digital Humanities, Semantic Web, historical corpora, temporal representations, Henri Poincaré, SPARQL query transformation

1. Introduction

Jules Henri Poincaré was born in Nancy, France, in 1854 and he died in Paris in 1912. He is considered one of the last great universal scholars because of the major contributions he made to several disciplines. It is mainly for his works in mathematics (automorphic forms, topology) and in physics (*Three-body problem*) that he has acquired his reputation. He also had an interest in popularizing science and he contributed to the philosophy of science, as shown by his three books published at the beginning of the 20^{th} century: *La Science et l'Hypothèse* [1], *La Valeur de la Science* [2] and *Science et Méthode* [3]. During his academic career, Henri Poincaré was involved in numerous administrations and learned societies. He played a role in French societies such as the French Academy of Sciences and the *Bureau des longitudes* but also in international societies such as the Dutch Society of Sciences in Haarlem, the Hungarian Academy of Sciences or the American Philosophical Society. Within these different institutions, he held various positions ranging from corresponding member (as was the case for several foreign institutions) to the role of president (the *Bureau des longitudes* — in 1899, 1909 and 1910 — and the *Société astronomique de France* — from 1901 to 1903).

The Henri Poincaré correspondence corpus is composed of around 2200 letters, both sent and received, which are of interest to historians of science. The study and exploitation of this corpus is a long-term project which has resulted in the publication of several thematic volumes. As an example, a volume is dedicated to the

*Corresponding author. E-mail: nicolas.lasolle@univ-lorraine.fr.

correspondence of his youth [4] when he was a student at the *École polytechnique* and then at the *École des mines* (1873-1878). Another volume gathers his regular exchanges with the Swedish mathematician Gösta Mittag-Leffler,[1] which spread throughout his academic career [5]. Another aspect of the Henri Poincaré project is the online publishing of this corpus.

Accessible since September 2018, the Henri Poincaré website[2] was created and is maintained thanks to the *Omeka S* content management system [6]. This platform enables the web-publishing and sharing of cultural collections from institutions (museums, archive places, etc.). The website is hosted by Huma-Num [7], which proposes services for Digital Humanities projects. On the website, a letter is associated with a scan of the original document,[3] a transcription,[4] a critical apparatus and a set of meta-data. Meta-data can either refer to the letter as a physical document (date of writing, place of dispatch, sender, recipient, etc.) or to its content (scientific topics discussed, people quoted, etc.). In addition to the Omeka S infrastructure, Semantic Web technologies have been used to offer users editing tools and search engines. Meta-data are described using the RDF model, and the SPARQL query language allows the formulation of complex queries exploiting the links between the resources of the corpus. For example, one may want to retrieve letters sent to members of the Academy of Sciences between 1880 and 1885, and with optics as a topic. Please see [8] to get more details about the infrastructure related to this corpus.

The corpus database is structured using an ontology defined within the AHP-PReST[5] laboratory. The current representation does not take into account temporal knowledge related to the statements described in the RDF database. However, this temporal knowledge is essential for the study of historical corpora. As an example, consider the fact that Henri Poincaré was a member of the French Academy of Sciences learned society. The related statement should be considered valid from his election date, on January 31, 1887.[6] Integrating temporal knowledge into RDF graphs is not an easy task and this field of research has aroused significant interest from the Semantic Web community. Several issues have been highlighted, but this article only considers the knowledge representation issue related to the association of a temporal element (e.g. an instant or a temporal interval) with an RDF statement. This explains the choice of an appropriate model to represent temporal knowledge and to query related statements using the SPARQL language. In particular, the proposed representation model should be easily integrated into the current database, which contains a significant number of triples (around 230 000). Another goal is to be able to reason with temporal knowledge. In line with this choice, research work has been carried out to improve information retrieval in temporal graphs.

The remainder of this article presents Semantic Web technologies used for representing and interrogating corpus data (Section 2). Next, temporal knowledge representation issues related to the exploitation of historical corpora are discussed. Several examples present the insufficiency of the current representation of the Henri Poincaré correspondence corpus (Section 3). A synthesis of existing models to represent temporal knowledge in RDF databases is then provided (Section 4). Four models are implemented and compared by using cases extracted from the Henri Poincaré corpus (Section 5). Associated with a dedicated Web interface, a SPARQL query transformation mechanism has been adapted to simplify the querying of temporal RDF graphs with the chosen representation (Section 6).

2. Preliminaries on Semantic Web technologies

This section introduces the Semantic Web technologies used to represent corpus data: the RDF model, the knowledge representation languages RDFS and OWL and the graph query language SPARQL.

2.1. *The* RDF *model*

Resource Description Framework (RDF) [9] is a knowledge representation model based on labeled directed graphs which is defined as a World Wide Web Consortium (W3C) standard. An RDF graph is composed of three kinds of nodes: *named resources*, *anonymous resources* and *literals*. A *named*

[1] Gösta Mittag-Leffler (1846-1927) was a Swedish mathematician who is famous for having introduced a theorem (now bearing its name) related to the representation of meromorphic functions by rational fraction series. In 1882, he created the *Acta Mathematica* journal where several mathematicians of the time were published, such as Georg Cantor, Sophie Kowalevski and Henri Poincaré.

[2] http://henripoincare.fr.

[3] Some are unavailable due to copyright rules.

[4] Around 40% of the letters are associated with a plain text transcription in XML or LaTeX.

[5] Archives Henri-Poincaré, Philosophie et Recherche sur les Sciences et les Technologies.

[6] Henri Poincaré was elected in the geometry section.

⟨letter11 sentBy henriPoincaré⟩
⟨letter11 sentTo göstaMittagLeffler⟩
⟨letter11 quotes charlesHermite⟩
⟨letter11 hasTopic "Non-Euclidean geometry"⟩
⟨letter11 hasWritingDate 1882-07-27⟩

Figure 1. A set of triples partially describing a letter.

resource is identified by an Internationalized Resource Identifier (IRI) and describes a class (e.g. `Person`, `Mathematician`, `Letter`), a property (e.g. `hasBirthDate`, `sentBy`, `hasWritingDate`) or a specific instance (e.g. `henriPoincaré`, `letter11`). An *anonymous resource* represents a resource that is not explicitly identified (*blank node*). A *literal* corresponds to a constant value of a specific datatype (e.g. `string`, `integer`, `date`).

Information is represented using statements, called triples, of the form ⟨*subject predicate object*⟩. The subject is a resource, the predicate is a property, and the object is either a resource or a literal. Figure 1 presents a set of RDF triples related to a letter of the Henri Poincaré correspondence corpus.

Several syntaxes are available to create databases in RDF format. In the beginning, the XML syntax was the most frequently used but other syntaxes, such as the Turtle syntax [10], are now commonly used to store RDF databases. In this document, an abstract syntax close to the Turtle syntax is used to represent RDF triples.

2.2. RDF Schema

RDF Schema (RDFS) [11] is a knowledge representation language extending the RDF model. Several properties are used to structure data: `rdfs:subclassof` (resp. `rdfs:subpropertyof`) serves to create hierarchies between classes (resp. properties). For instance, the triple ⟨`Letter rdfs:subclassof Document`⟩ states that the concept of `Letter` is more specific than the concept of `Document`. `rdfs:domain` (resp. `rdfs:range`) can be applied to properties and add restrictions about the type of the resource in *subject* position (resp. *object*) for a triple. In the remainder of the article, short names are used for these properties: `a` for `rdf:type`, `subc` for `rdfs:subclassof`, `subp` for `rdfs:subpropertyof`, `domain` for `rdfs:domain` and `range` for `rdfs:range`.

An RDFS graph constitutes a knowledge base from which new knowledge can be generated. The RDFS deduction refers to the application of inference rules for extracting knowledge.

There are other knowledge representation languages available for RDF, such as OWL.

2.3. OWL

Web Ontology Language (OWL) [12] is a knowledge representation language which corresponds to the $\mathcal{SROIQ}(\mathcal{D})$ description logic. Many constructs are introduced by OWL to describe classes and properties and thus structure ontologies. Among these, we can highlight properties related to the equality concept (`owl:equivalentClass`, `owl:differentFrom`, etc.), to symmetry (`owl:inverseOf`) and to the conjunction of classes (`owl:intersectionOf`). OWL provides additional elements for reasoning about classes. Thus, it is possible to define the union (`owl:unionOf`), the complementary (`owl:complementOf`) and the disjunction (`owl:disjointWith`) of classes.

2.4. SPARQL

SPARQL is the language recommended by the W3C to query RDF graphs [13]. A common form of SPARQL query consists in a SELECT clause, containing one or several anonymous resources, followed by a WHERE clause, composed of a *graph pattern* divided into one or several *triple patterns*,[7] and a possible FILTER clause that is used to add constraints for the literal values associated with the properties. Consider the following informal query:

$\mathcal{Q} =$ | Give me the letters sent by Henri Poincaré between 1885 and 1890 and having *non-Euclidean geometry* as a topic.

This query can be expressed using SPARQL:

```
Q = | SELECT ?l
    | WHERE {
    |     ?l a letter .
    |     ?l sentBy henriPoincaré .
    |     ?l subject "Non-Euclidean geometry" .
    |     ?l hasWritingDate ?y .
    |     FILTER(YEAR(?y) >= 1885
    |        AND YEAR(?y) <= 1890)
    | }
```

3. Knowledge representation for historical corpora

The purpose of this section is to discuss issues related to the representation of historical knowledge.

3.1. Time and History

History is a social science based on the organization and interpretation of temporal data. It relies on a scientific methodology for analyzing sources and aims at producing a consistent discourse on the past likely to

[7] A triple pattern is a list of (subject, predicate, object) triples which can be written in different abbreviated ways.

shed light on present time debates [14]. Part of a historian's work therefore consists in producing chronologies and organizing his/her corpus of data into time periods adapted to the studied object. This diachronic work is undoubtedly validated by the rigorous mobilization of sources and the viewpoint of subject field specialists, but it is also a matter of choices involving a degree of subjectivity and/or contingency. The availability of some sources and their fragmentary or incomplete nature, the affiliation to a school or style of writing history (sociohistory, microhistory, structuralism, internalist or externalist approaches in the history of science), belonging to a specific field of historical research (political history, colonial history, history of science), the relative importance attributed to certain facts or actors, the adherence to a political conception of the role of historians: these are all parameters that will affect the diachronic organization of corpora and their interpretation. The importance given to the national narrative, which can be structured by political or ideological considerations which have nothing to do with history, can also play a role in the choice of periodization.

Thus, while working on the same period, two historians will not necessarily develop the same narrative, will not necessarily identify the same discontinuities and may not give the same importance to a factual chronological sequence. In the same way, two historians of science working on the same subject may develop different chronologies to interpret the same foundational event, for example, because they give more or less importance to actors considered to be secondary. Thus, Clifford D. Conner [15] proposes a reinterpretation of the world history of science, inspired by the *popular history* movement initiated by Howard Zinn [16]. His work proposes a periodization which gives pride of place to little-known events or agents who were able, through their sometimes anonymous works, to pave the way for those of scholars enjoying a prominent place in the History of Science Pantheon (Galileo, Newton, Lavoisier, etc.). He also demonstrates that a periodization of the history of science cannot disregard the technical aspect of science and the role played by craftsmen and technicians in the evolution of the mastery and understanding of nature.

The temporalities are thus multiple and the proposed periodizations correspond to historical constructions aiming at providing a set of common reference points to apprehend the links which can exist between present, past and future [17]. It is therefore a major challenge to take these temporal elements into account when representing historical corpora knowledge. These temporal knowledge representation issues are discussed in the remainder of this section. Although the examples are related to the Henri Poincaré corpus, these problems are not specific to it.

3.2. *An ontology for the history of science*

Within the AHP-PReST laboratory, several research works are dedicated to the study and dissemination of Henri Poincaré's works. Through research around his scientific publications, it is possible to retrace his discoveries for and contributions to the different disciplines he cherished. In particular, this makes it possible to value the work of some of his colleagues who were conducting research on related themes. In addition, Henri Poincaré wrote numerous reports as part of his engagement in various institutions and learned societies.

There has been particular interest in the study of his correspondence. Why study this corpus? Several aspects need to be considered in order to answer this question. First, the study of a correspondence such as that of Henri Poincaré provides contextual elements related to the development of scientific theories. Indeed, at the end of the 19th century, the letter was an important vector of scientific information [18]. During his career, Henri Poincaré maintained an active correspondence with many of his French and foreign colleagues. For example, his correspondence with the Swedish mathematician Gösta Mittag-Leffler spanned his entire career and allows us to understand some of their respective works. Studying this corpus can also help to understand the organization of several administrations and academies. More generally, it can provide some key elements related to the social, political and cultural context of the time. Finally, this corpus presents Henri Poincaré in a different light from that put forward by the analysis of his work and thus introduces new biographical elements.

Henri Poincare corpus data is structured using an ontology[8] defined within the AHP-PReST laboratory. Its aim is to represent knowledge related to historical corpora, by modeling documents (articles, books, reports, letters, etc.), persons and institutions, places, etc.

This ontology has several limitations and is currently being restructured. The model suffers from ambiguities and redundancies, and it has become necessary to

[8]The term ontology refers to the set of concepts and properties used to represent the knowledge of a domain.

propose a more relevant alignment with existing ontologies: dcterms,[9] bibo,[10] bio,[11] rel,[12] etc. Improving and publishing this ontology would allow the scientific community to reuse it for other corpora in history. One of the main problems encountered during this restructuring work concerns the temporal validity of the facts, which was not addressed in previous versions of the ontology. The remainder of the article focuses on this aspect of the ontology restructuring.

3.3. *Timing facts*

Several facts from the correspondence corpus would be described in a more relevant way by including temporal elements. A first issue concerns the description of the residential addresses associated with the persons. In the current version of the ontology, the `livesAt` property is sometimes used to associate a place with a person. However, the meaning of this property is not clearly defined. Does this correspond to a main place of residence which could be interpreted as the place where the person resided for most of his/her life? Or is it the place of residence at the time of the person's death? In some cases, this may also correspond to the only known living place associated with the person. In a history of science context, it is necessary to consider knowledge that is sometimes incomplete because of the lack of resources related to a certain context. In the context of this corpus, it is common for a given address to correspond to a hypothetical place of residence based on the workplace associated with the person. To improve the `livesAt` property semantics, it is necessary to reflect in order to propose a better-structured set of properties. It would also be necessary to be able to associate several places with a person and to be able to distinguish them unambiguously. These issues require the addition of a temporal element to these relationships between people and places.

Another problem is related to the representation of relationships between people, which can evolve and be time-limited. Indeed, some family ties are true on a permanent basis (or at least, considering the time interval during which the two people are alive). Other forms of relationships are limited in time. This is the case, for example, for the marriage relationship, which has a start date and an end date. The fact that Henri Poincaré was married to Jeanne Louise Poulain d'Andecy is valid from April 20, 1881, the date of the wedding ceremony, to his death on July 17, 1912. Another form of knowledge, not represented at this time, concerns work relationships that could be referred to as *colleagues*. During his career, Henri Poincaré was a member of several institutions. He was also involved as a teacher in numerous institutions (*École polytechnique*, *École des mines*, University of Paris, etc.). As a result, he interacted with many people who could be described as colleagues. However, each of these relationships existed for a specific period of time. For example, Henri Poincaré worked with many people at the Faculty of Sciences in Paris where he held various chairs: *mécanique physique et expérimentale* (1884-1886), *physique mathématique et calcul des probabilités* (1886-1896), *astronomie mathématique et mécanique céleste* (1896-1912). He was thus able to interact with people such as Gaston Darboux, Émile Picard or Joseph-Alfred Serret. Temporal aspects should therefore be considered when using the underlying properties.

Another issue is to be able to consider unions of time intervals. For example, Henri Poincaré held various positions within the *Bureau des longitudes*.[13] A member from 1893, he was appointed secretary for the years 1896 and 1897. He was also president of the institution from January 1899 to January 1900 and a second time from February 1909 to December 1910 [19]. Thus, a person interested in the history of this institution could search for the moment when Henri Poincaré was its president. Such a search must return a time element corresponding to a finite union of disjoint time intervals: $[18\ \texttt{January}\ 1899, 20\ \texttt{January}\ 1900] \cup [17\ \texttt{February}\ 1909, 28\ \texttt{December}\ 1910]$.

Beyond the mere integration of temporal elements, another objective is to be able to set up reasoning in order to extract knowledge from corpus data. For instance, if a letter from the correspondence corpus does not have an explicit date of writing but the correspondent congratulates Henri Poincaré for his forthcoming wedding, it is possible to deduce that the letter was written after April 20, 1881. If, in the same letter, the correspondent

[9]https://www.dublincore.org/specifications/dublin-core/dcmi-terms/.
[10]http://www.bibliontology.com/.
[11]https://vocab.org/bio/.
[12]https://vocab.org/relationship/.

[13]This French scientific and technical academy was created in 1795 with the stated objective of regaining control of the seas by improving the system for determining longitudes [19]. Soon, the disciplines dealt with by the *Bureau des longitudes* expanded to include geodesy, celestial mechanics, cosmology, metrology, etc. This institution functioned as an academy alongside the Academy of Sciences.

announces to Henri Poincaré that he is coming to Paris for the International Electricity Exhibition,[14] the exhibition having taken place between August 10, 1881 and November 20, 1881, it becomes possible to associate the letter with a date of writing within the interval [20 `April` 1881, 20 `November` 1881]. Thereafter, perhaps other elements could lead to a narrower interval or the precise date of writing. This type of reasoning has frequently been used to estimate and sometimes determine the date of writing of certain letters in the Henri Poincaré correspondence corpus.[15]

For this knowledge representation work, the granularity is defined at the level of the day. This allows consideration of data associated with letters in the correspondence for which the day of writing is sometimes known. Depending on the context, this level of granularity could be too precise or too vague—for example, in the study of war battles, where the granularity should be in the order of an hour or even a minute. In history, it is sometimes very difficult to estimate the date associated with an event. Some dates are related to choices defined to ensure consistency within a chronology and correspond to the most probable hypothetical date with regard to research work.

4. Existing temporal data representation models

This section presents a summary of the main temporal knowledge representation models and ontologies for RDF data.

4.1. Including temporal data in RDF graphs

4.1.1. Temporal graphs and RDF reification

Temporal RDF [21] is one of the first approaches to represent temporal data within RDF databases. A temporal graph is composed of a set of temporal triples, labeled triples of the form $\langle s \;\; p \;\; o\rangle : [t]$ where t corresponds to the temporal element defined as the moment during which the fact associated with the triple is valid. A syntax inspired by RDF *reification* is introduced to associate the label with a triple within a classical RDF graph. The concept of RDF reification has been introduced by the W3C to add meta-properties to statements represented with RDF [9]. The *temporal* RDF model introduces a vocabulary composed of the properties `temporal`, `instant`, `interval`, `initial` and `final`, and the literal `now` which can be used to describe temporal elements.

This approach is enhanced by the addition of anonymous temporalities in order to represent and to reason with facts for which the temporal component is unknown or incomplete but nevertheless essential [22].

4.1.2. Alternatives to the standard reification

n-ary relations [23] is another W3C approach to describe facts with meta-properties. For including temporal data, the *n-ary* approach consists in adding an intermediate entity which is linked to the value of the property of the initial relation as well as to one or more temporal components.

RDF* and SPARQL* were introduced by [24] and form an alternative to standard reification. A triple nesting model is introduced and serves to define a triple at the subject or object position of another triple. This approach requires an extension of the semantics of RDF and an extension of the SPARQL language. After transformation, an RDF* graph is similar to an RDF graph which uses standard RDF reification.

The use of (*Named graphs*) [25] allows the decomposition of a graph $\mathcal{G}$ into a set of graphs: $\{(\mathcal{G}_1, u_1), (\mathcal{G}_2, u_2), ..., (\mathcal{G}_n, u_n)\}$ in which u_i corresponds to an IRI identifying the graph $\mathcal{G}_i$. Such a graph can be described with meta-properties. The use of named graphs is particularly useful when several facts represented by triples are associated with the same temporal element, but it is not relevant in the case where most triples must be associated with different temporal elements. RDF+ [26] is another RDF extension proposed for the representation of meta-data which is based on the use of named graphs.

4.1.3. Representations based on a perdurantist view

4D fluent [27] constitutes an approach in which some individuals are decomposed in a set of temporal entities. This type of entities is called *perdurant*, as opposed to *endurant*. Relationships between resources are defined by relationships between their temporal parts. A temporal entity can be used for several relationships if the corresponding resource is associated with several facts with the same temporality. From an ontological point of view, this approach may seem more interesting than

[14] The *Exposition internationale d'Électricité* is one of the founding acts in the history of electricity applications. It took place in Paris, at the Palais de l'Industrie, between August 10, 1881 and November 20, 1881 and attracted about 900,000 visitors [20]. Several technological advances were presented, such as Thomas Edison's light bulbs, Werner von Siemens's electric streetcar or Alexander Graham Bell's telephone. The first international congress of electricians was also organized in parallel to the exhibition.

[15] However, only a small number of letters are associated with a precise date of writing.

others because it integrates clear semantics: additional objects are temporal components of perdurant entities.

In [28], the authors propose to simplify the representation based on a perdurantist view. Unlike in the *4D fluent* approach, the original entity is directly interpreted as a temporal part. This is done in order to simplify the transition to a temporal representation by avoiding the need to rewrite the ontology.

The CIDOC Conceptual Reference Model (CRM) is a knowledge representation model suited to represent cultural heritage knowledge.[16] To associate temporal information to facts, this model proposes a solution which is influenced by the perdurantist view. The model is based on the notion of *temporal entities* which corresponds to phenomena (which can be periods, events or states) that happen at a specific contiguous extent in time. A temporal entity is associated to a *time span* which describes the temporal interval or instant associated with this entity. The model includes the definition of abstract temporal extents.

4.1.4. *The* Singleton Property *approach*

The *Singleton Property* approach [30] introduces a new type of property, called *singleton property*, used only once for a particular context. A *singleton property* can be linked to a generic property using the `rdf:singletonPropertyOf` property. Thus, rather than adding meta-data to a triple, a set of triples, or a resource, this method describes the *singleton property* used. Works extend this approach by defining `rdf:singletonPropertyOf` as a sub-property of `rdfs:subPropertyOf` [31]. This has two consequences on the reasoning: an inference engine can now recognize triples which use a *singleton property* and it can also infer a triple described using a *singleton property*.

4.1.5. *Representations using an annotation mechanism*

Other representation formalisms are based on an annotation mechanism. *a*RDF [32] proposes an extension of the RDF language to represent meta-data associated with facts. A set of *a*RDF triples is expressed as $\langle s, p{:}a, v\rangle$, in which *a* corresponds to the time element.

RDFt [33] is another model for annotating triples in order to add temporal components. Contrary to previous approaches, this model makes it possible to represent the temporal validity of a fact by saving the number of modifications made in the database.

[16] Information about the updated version of the model can be accessed online (http://www.cidoc-crm.org/). For a overview of the origin and the methodology related to the creation of the model, see [29].

4.2. *Time ontologies*

The approaches presented in this section are associated with different ontologies for representing temporal elements. The objective of this section is to provide a brief overview of several temporal ontologies in RDFS and OWL. To compare these approaches, it is particularly relevant to look at the integration of Allen's interval algebra, which defines 13 relationships between time intervals: `before` and `after`, `meets` and `met-by`, `overlaps` and `overlapped-by`, `starts` and `started-by`, `finishes` and `finished-by`, and `equal` which is its own inverse. This algebra can be used to formalize facts and for reasoning. Let *A*, *B* and *C* be 3 time intervals and *r*1 and *r*2 be two relationships such that $\langle A\ \texttt{r1}\ B\rangle$ and $\langle B\ \texttt{r2}\ C\rangle$. Depending on the nature of *r*1 and *r*2, it is possible to deduct or restrict the nature of the relationship between *A* and *C*. For example, $\{\langle A\ \texttt{before}\ B\rangle, \langle B\ \texttt{before}\ C\rangle\} \vdash \langle A\ \texttt{before}\ C\rangle$. A composition table is provided for all possible pairs $(\texttt{r1}, \texttt{r2})$.

OWL-Time [34] is the ontology recommended by the W3C to represent temporal elements. It was introduced in 2006 and the current version is compatible with the second version of OWL. This ontology introduces several concepts: temporal entities *Instant* and *Interval*, elements related to temporal systems, elements to represent time positions and durations. The ontology integrates the representation of the 13 relations of Allen's interval algebra.

Time-determined Web Ontology Language (tOWL) [35] is an alternative to OWL-Time corresponding to a temporal extension of the $\mathcal{SROIQ}(\mathcal{D})$ description logic. This extension is divided into 3 layers: the use of concrete domains to represent restrictions using binary predicates, the introduction of concepts (instants, intervals) and relations including the representation of Allen's algebra, and a last layer proposing elements for a persistent view.

HuTO [36] is an ontology formalized in RDFS which allows resources to be annotated by using common language time expressions. 5 types of distinct temporal expressions are considered and can thus be represented: explicit expressions (e.g. December 31, 2020, Winter 2020), deictic expressions (e.g. yesterday, in 10 days), durations (e.g. 1 per hour, 30 days), cyclical expressions (e.g. every two days, every Sunday) as well as mixed expressions (e.g. during one week 3 years ago). Among the 13 relationships derived from Allen's interval algebra, only *before* and *after* are represented.

4.3. Synthesis

Table 1 compares the existing approaches to representation of temporal data within RDF databases. Several criteria are used:

Year: year of publication of the first article presenting the approach.
RDF extension: is an extension of RDF necessary?
Structure: level of structure associated with the representation (triple, quadruple, quintuple, etc.).
Number of tuples: number of tuples necessary to represent a fact by integrating a temporal component.
Semantics: semantics level (RDF, RDFS, OWL, etc.).
Query language: query language associated with the approach.
Ontology: ontology associated with the approach.

5. Implementation and comparison of approaches for the Henri Poincaré corpus

5.1. Methodology

This section details the use of four approaches introduced in the previous section: *Temporal* RDF, *4D fluents*, *Singleton Property* and *n-ary*. Only representation models that do not require major extensions to RDF were retained. For each of the selected approaches, several aspects are studied and presented:

- The representation of a fact using a set of RDF triples;
- A graphical visualization of this representation;
- The writing of two SPARQL queries associated with temporal elements.

The fact to be represented is related to the life of Henri Poincaré:

$\mathcal{F} =$ Henri Poincaré lived in Paris from 1881 to 1912.

In the current database, this fact is represented in an incomplete manner as a binary relation `livesAt(henriPoincaré,Paris)`. Henri Poincaré corresponds to a resource described in the correspondence corpus database,[17] while Paris is defined as a *Geonames* resource.[18] As explained in Section 3, this representation is not adapted and it is necessary to add a temporal component. Each representation approach will be implemented with the OWL-Time ontology (except *Temporal* RDF which introduces a temporal vocabulary). Here are the two requests to be made, expressed here in an informal way:

$\mathcal{Q}_1 =$ Give the place of residence of Henri Poincare in 1900.

$\mathcal{Q}_2 =$ | Give the temporal intervals during which Henri Poincare was president of the Bureau des longitudes.

$\mathcal{Q}_1$ aims at retrieving the Geonames resource corresponding to the place of residence of Henri Poincaré in 1900. $\mathcal{Q}_2$ must return a union of two disjoint intervals because Henri Poincaré was president of the *Bureau des longitudes* from January 1899 to January 1900 and a second time between February 1909 and December 1910. In the resulting SPARQL queries, the term *Bureau des longitudes* is shortened to *bdl*.

5.2. Temporal RDF

To add temporal elements to facts, this method proposes a syntax based on RDF reification. An ontology is also associated with the proposed model. To represent $\mathcal{F}$, an instance of the class `Statement` is created and this instance is associated with the triple ⟨`henriPoincaré livesAt paris`⟩ thanks to the properties `subject`, `predicate` and `object`. This *statement* is defined by a temporal element `t1` associated with a time interval `i1`, described by a start date (1881) and an end date (1912). Figure 2 shows the set of 8 triples required together with a graphical visualization of this representation.

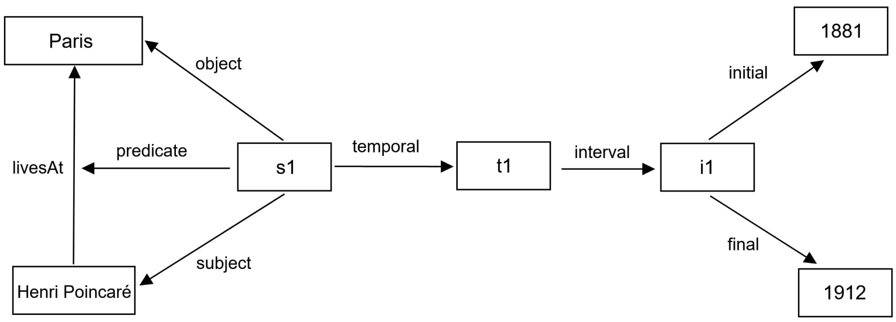
The formulation of SPARQL queries is quite cumbersome because it is necessary to identify the *statement* in order to access the temporal element, which increases the number of triple patterns needed. This issue is expressed in $\mathtt{Q}_1^{\mathrm{TR}}$ and $\mathtt{Q}_2^{\mathrm{TR}}$, formal queries associated with $\mathcal{Q}_1$ and $\mathcal{Q}_2$ (TR stands for *Temporal* RDF). However, one advantage is that the original triple is not altered and it is thus simple to access data without being interested in temporal elements. For example, to find all places of residence of Henri Poincaré which are defined in the database, the query to formulate is not very

[17] http://henripoincare.fr/api/items/843.

[18] https://www.geonames.org/2988507.

Table 1
Comparison of temporal data representation models.

Approach	Year	RDF extension	Structure	Number of tuples	Semantics	Query language	Ontology
Temporal RDF	2005	Yes	Triple	8	Temporal RDF	Model	Model
Named graphs	2005	No	Triple	8	RDFS	τ SPARQL	Any
n-ary	2006	No	Triple	5	OWL	SPARQL	Any
4D fluents	2006	No	Triple	7	OWL	SPARQL	OWL-Time
RDF$^+$	2008	Yes	Quintuple	5	RDF$^+$	ext. SPARQL	Any
*a*RDF	2010	Yes	Triple + Label	N/A	*Annotated Logic*	Algorithm	Any
Singleton property	2012	Yes	Triple	5	RDFS	SPARQL	Any
RDF*	2017	Yes	Triple	7	RDF*	SPARQL*	Any
RDFt	2019	Yes	Triple + Label	8	RDFt	SPARQL[t]	RDFt



(a) Graphical representation of $\mathcal{F}$ with the *Temporal* RDF approach.

```
⟨henriPoincaré livesAt paris⟩ ⟨s1 subject henriPoincaré⟩
⟨s1 predicate livesAt⟩        ⟨s1 object paris⟩
⟨s1 temporal t1⟩              ⟨t1 interval i1⟩
⟨i1 initial 1881⟩             ⟨i1 final 1912⟩
```

(b) Triples representing $\mathcal{F}$ with the *Temporal* RDF approach.

Figure 2. Representation of $\mathcal{F}$ with the *Temporal* RDF approach.

complex and remains the same whether or not there are temporal elements associated with the facts.

$Q_1^{TR} =$

```
SELECT ?loc
WHERE {
    ?s1 subject henriPoincaré .
    ?s1  predicate livesAt .
    ?s1 object ?loc .
    ?s1 temporal ?t1 . ?t1 interval ?i1 .
    ?i1  initial ?d1 . ?i1 final ?d2
    FILTER(?d1 <= 1900 AND ?d2 >= 1900)
}
```

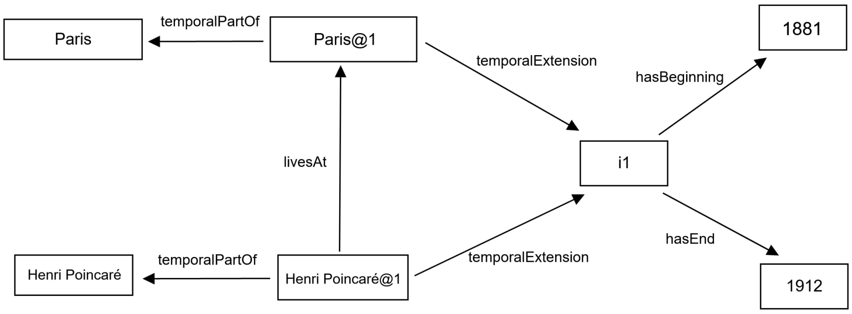
$Q_2^{TR} =$

```
SELECT ?d1 ?d2
WHERE {
    ?s1 subject henriPoincaré .
    ?s1 predicate presidentOf .
    ?s1 object bdl .
    ?s1 temporal ?t1 . ?t1 interval ?i1
    ?i1 initial ?d1 . ?i1 final ?d2
}
```

5.3. 4D fluents

This approach proposes to decompose resources into a set of temporal parts. In order to represent $\mathcal{F}$, it is necessary to add a temporal part to the resources identifying Henri Poincaré and Paris. These temporal parts are both linked to the same temporal element defined as a temporal interval. In total, 7 triples are required to represent $\mathcal{F}$, as shown in figure 3.

The queries Q_1^{4D} and Q_2^{4D} highlight the complexity of SPARQL queries for the *4D fluent* approach. For each of these requests, it is necessary to specify the links between the resources, the temporal parts and the temporal elements.



(a) Graphical representation of $\mathcal{F}$ with the *4D fluents* approach.

```
⟨henriPoincaré@1 temporalPartOf henriPoincaré⟩ ⟨paris@1 temporalPartOf paris⟩
⟨henriPoincaré@1 livesAt paris@1⟩             ⟨henriPoincaré@1 temporalExtension i1⟩
⟨paris@1 temporalExtension i1⟩                ⟨i1 hasBeginning 1881⟩
⟨i1 hasEnd 1912⟩
```

(b) Triples representing $\mathcal{F}$ with the *4D fluents* approach.

Figure 3. Representation of $\mathcal{F}$ with the *4D fluents* approach.

$Q_1^{4D} =$

```
SELECT ?loc
WHERE {
    ?p1 temporalPartOf henriPoincaré .
    ?p2 temporalPartOf ?loc .
    ?p1 livesAt ?p2 .
    ?p1 temporalExtension ?i1 .
    ?p2 temporalExtension ?i1 .
    ?i1  hasBeginning ?d1 .
    ?i1 hasEnd ?d2
    FILTER(?d1 <= 1900 AND ?d2 >= 1900)
}
```

$Q_2^{4D} =$

```
SELECT ?d1 ?d2
WHERE {
    ?p1 temporalPartOf henriPoincaré .
    ?p2 temporalPartOf bdl .
    ?p1 presidentOf ?p2 .
    ?p1 temporalExtension ?i1 .
    ?p2 temporalExtension ?i1 .
    ?i1 hasBeginning ?d1 .
    ?i1 hasEnd ?d2
}
```

5.4. *Singleton property*

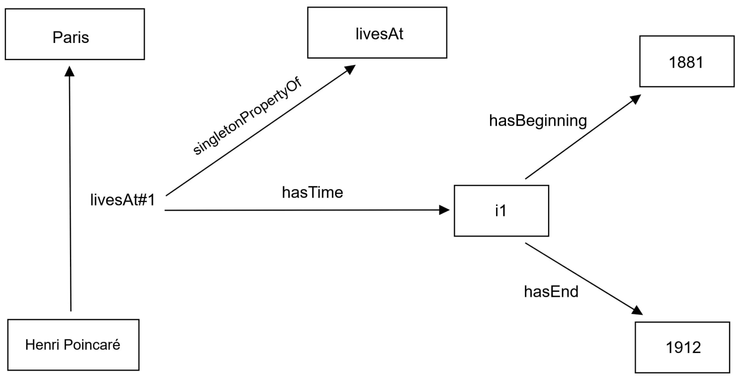
In the case of the *Singleton Property* approach, few triples are required to describe a fact by including a temporal element. For representing $\mathcal{F}$, the `livesAt#1` property is defined as a `singletonPropertyOf` of the `livesAt` property. It is therefore necessary to create a `livesAt#`*i* property each time you wish to use the `livesAt` property with a temporal element. $\mathcal{F}$ is represented by 5 triples, as shown in figure 4. The introduction of *singleton properties* makes the writing of SPARQL queries only slightly more complex. The Q_1^{SP} and Q_2^{SP} queries have a lower number of triple patterns than the triple patterns of the queries of the two previous approaches (5 versus 7 for both $\mathcal{Q}_1$ and $\mathcal{Q}_2$). Moreover, the extension proposed in [31] simplifies the formulation of queries for which the temporal element of a fact is not requested.

$Q_1^{SP} =$

```
SELECT ?loc
WHERE {
    henriPoincaré ?p ?loc .
    ?p singletonPropertyOf livesAt .
    ?p hasTime ?i1 .
    ?i1  hasBeginning ?d1 .
    ?i1 hasEnd ?d2
    FILTER(?d1 <= 1900 AND ?d2 >= 1900)
}
```

$Q_2^{SP} =$

```
SELECT ?d1 ?d2
WHERE {
    henriPoincaré ?p bdl .
    ?p singletonPropertyOf presidentOf .
    ?p hasTime ?i1
    ?i1 hasBeginning ?d1 .
    ?i1 hasEnd ?d2
}
```



(a) Graphical representation of $\mathcal{F}$ with the *Singleton Property* approach.

⟨henriPoincaré livesAt#1 paris⟩ ⟨livesAt#1 singletonPropertyOf livesAt⟩
⟨livesAt#1 hasTime i1⟩ ⟨i1 hasBeginning 1881⟩
⟨i1 hasEnd 1912⟩

(b) Triples representing $\mathcal{F}$ with the *Singleton Property* approach.

Figure 4. Representation of $\mathcal{F}$ with the *Singleton Property* approach.

5.5. *n-ary*

This approach represents $\mathcal{F}$ by simulating a ternary relation thanks to the introduction of a blank node. This node is used to link the use of the `livesAt` property to a value and a temporal element. Figure 5 illustrates this representation in which 5 triples are required. As in the case of the *singleton property* approach queries, $Q_1^{n\text{-ary}}$ and $Q_2^{n\text{-ary}}$ have a low number of triple patterns. The *n-ary* approach requires to work with blank nodes, but this mechanism integrates well with SPARQL queries. However, with this approach, the assertion of the original triple is lost, which makes it necessary to add a triple pattern in queries when one does not want to recover the temporal element.

$Q_1^{n\text{-ary}} =$

```
SELECT ?loc
WHERE {
    henriPoincaré livesAt ?node .
    ?node hasValue ?loc .
    ?node hasTime ?i1 .
    ?i1  hasBeginning ?d1 .
    ?i1 hasEnd ?d2
    FILTER(?d1 <= 1900 AND ?d2 >= 1900)
}
```

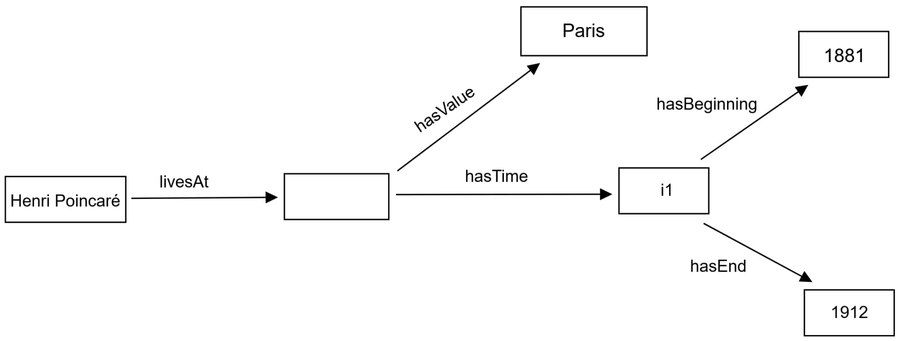
$Q_2^{n\text{-ary}} =$

```
SELECT ?d1 ?d2
WHERE {
    henriPoincaré presidentOf ?node .
    ?node hasValue bdl .
    ?node hasTime ?i1 .
    ?i1 hasBeginning ?d1 .
    ?i1 hasEnd ?d2
}
```

5.6. *Conclusion and selection of the approach*

The implementation of these different approaches highlights their pros and cons for concrete cases of editing and information retrieval. For the Henri Poincaré corpus, it has finally been decided to opt for the model based on *n*-ary relations. This approach seems to be one of the most intuitive and has been recommended by the W3C. No extension of the semantics of RDF is necessary and the number of additional triples introduced by the representation remains low compared to other models. Moreover, the proposed structure makes it possible to add additional elements to describe the facts (degree of certainty, author, commentary, etc.).

The following section introduces a contribution related to the exploitation of temporal data for Semantic Web databases. This work may be particularly relevant to the study of historical corpora. The adaptation of a transformation rules mechanism is proposed to improve information retrieval in temporal RDF graphs.



(a) Graphical representation of $\mathcal{F}$ with the *n-ary* approach.

⟨henriPoincaré livesAt ?node⟩	⟨?node hasValue paris⟩
⟨?node hasTime i1⟩	⟨i1 hasBeginning 1881⟩
⟨i1 hasEnd 1912⟩	

(b) Triples representing $\mathcal{F}$ with the *n-ary* approach.

Figure 5. Representation of $\mathcal{F}$ with the *n-ary* approach.

6. Temporal knowledge querying and reasoning using a transformation rule mechanism

This section presents a form-based interface used to generate SPARQL queries for the interrogation of the Henri Poincaré correspondence corpus. This user interface is combined with the application of a transformation rule mechanism which allows to assist the querying of temporal knowledge. This tool, named SQTRL [37], allows the definition and application of SPARQL query transformation rules. Another tool, named RTRL and which is inspired by SQTRL, is introduced to represent entailment rules for temporal data.

6.1. *A* SPARQL *query generation interface*

SPARQL is an expressive query language which is the recommendation when it comes to query RDF data. But writing SPARQL queries requires a good understanding of language capabilities and syntax which is not well-suited to most historians of science and people that are not used to Semantic Web technologies. To address this issue, a Web interface has been developed for the querying of the Henri Poincaré correspondence corpus database. This interface consists in a form containing a set of inputs which can be edited by users. The set of inputs is generated according to the type of the targeted resource. If one needs to find some information about a correspondent, he/she should select the *person* class and will be proposed inputs corresponding to the properties used to describe instances of this class (e.g. first and family names, birth date, place of residence). The filled inputs are used to generate a SPARQL query to be executed over the corpus graph. This interface is suitable for all users as it does not require any specific knowledge about SPARQL syntax or about the ontology.

Imagine a historian looking for information about Henri Poincaré's places of residence. Two situations are considered: either the historian wishes to retrieve all places of residence without temporal information, or he/she wants to list the pairs (place, temporal interval). The proposed interface allows users to choose a mode of query generation, depending on whether or not he/she is interested in temporal elements. An excerpt of this form-based interface is presented in Figure 9.

6.2. *Taking into account the n-ary representation*

The main challenge related to the query generation is that the system must consider the specific *n*-ary representation used to associated a temporal extent to several facts of the corpus. Indeed, with this representation, the property value and the temporal extent must be accessed by writing a non-usual triple pattern form in the SPARQL query body. A solution to this problem is to use the form-based mode to generate an incomplete query, which is then compiled by a system into the expected query which takes into account the *n*-ary representation. The final query presented to the user is the one to be executed over the corpus graph. One way to implement such a system is to use SQTRL to define syntactic transformation rules.

6.2.1. SPARQL *Query Transformation Rule Language*

SQTRL is a tool, associated with a language, that has been introduced to allow flexible querying with the SPARQL language. This mechanism works by defining transformation rules that can be applied to generate new queries from an initial query and an RDF database. Rules can be application-independent (generalization, specialization, etc.) or application-dependent (exchange of the sender and recipient of the letter, replacement of the topic of the letter by a related topic, etc.). An SQTRL rule r is defined by a set of fields:

- `name`(r): a string identifying the rule;
- `context`(r): an RDF triple pattern to search for in the database;
- `left`(r): an RDF triple pattern to be matched with the initial query;
- `right`(r): an RDF triple pattern which replaces `left`(r);
- `cost`(r): a positive number representing the transformation cost;
- `explanation`(r): a text describing the rule in an informal way and which may contain variables from `context`, `left` and `right`.

Let $\mathcal{G}$ be an RDF graph, $\mathtt{Q}$ be a SPARQL query, and r be an SQTRL rule. r can be applied to $\mathtt{Q}$ if `context`(r) can be bound with $\mathcal{G}$ and `left`(r) can be bound with the body of $\mathtt{Q}$. The application of r to $\mathtt{Q}$ generates $\mathtt{Q}'$ by substituting `left`(r) by `right`(r). A cost is associated with each of the rules to favor the application of certain rules. For most SQTRL rules in this article, the cost is null, meaning that the application of this rule does not introduce any imprecision.

The rules are represented using an XML syntax. Figure 6 shows an example of a rule which is used to replace an instance (`?o`) by a variable (`?x`). This variable is of the type of one of the classes to which the instance belongs (`?C`).

The definition of transformation rules should consist in a joint work between a knowledge engineer and an application domain expert. Once a rule has been defined, it can be reused without any modification in several tools and systems.

```
<rule name="Generalize instance in object position">
  <context>?o a ?C</context>
  <left>?s ?p ?o</left>
  <right>?s ?p ?x . ?x a ?C</right>
  <cost>2.0</cost>
  <explanation>
    Generalize ?o in any instance of ?C.
  </explanation>
</rule>
```

Figure 6. An example of SQTRL rule in XML syntax.

6.2.2. *Simplifying query generation by applying transformation rules*

For the purpose of this application, the transformation rules r_1 and r_2 are introduced (Figure 7). r_1 is used to retrieve the value of a property and r_2 is used to access the value with the associated temporal element. In both cases, the form-based interface does not need to be adapted to consider the structure associated with the *n*-ary representation. The interface generates a query of the form `SELECT ?o WHERE {?s ?p ?o}` and the SQTRL system generates a query containing the full triple pattern allowing access to the value and, if necessary, to the temporal element.

In our example, an historian was interested in finding information about Henri Poincaré places of residence. By completing the appropriate fields, he/she can use the interface to first generate the query $\mathtt{Q}_h$. In accordance with the selected mode, one of the two transformation rules applies to transform the query and to be able to retrieve the value and, if necessary, the temporal extent related to each statement. The application of r_1 generates the query $\mathtt{Q}_{h_1}$ and the application of r_2 generates the query $\mathtt{Q}_{h_2}$.

$\mathtt{Q}_h =$
```
SELECT ?v
WHERE {
    ?x firstName "Henri" .
    ?x familyName "Poincaré" .
    ?x livesAt ?v .
}
```

$\mathtt{Q}_{h_1} =$
```
SELECT ?v
WHERE {
    ?x firstName "Henri" .
    ?x familyName "Poincaré" .
    ?x livesAt ?node .
    ?node hasValue ?v
}
```

```
<rule name="n-ary value">
  <context>
    ?x ?p ?y . ?y :hasValue ?v
  </context>
  <left>?s ?p ?o</left>
    ?s ?p ?node . ?node :hasValue ?o
  </right>
  <cost>0</cost>
  <explanation>
    Adapt the triple pattern to the n-ary struc-
    ture to retrieve the value associated to ?p.
  </explanation>
</rule>
```

(a) The transformation rule r_1.

```
<rule name="n-ary value plus time>
  <context>
    ?x ?p ?y . ?y :hasValue ?v . ?y :hasTime ?t .
    ?t :hasBeginning ?b . ?t :hasEnd ?e
  </context>
  <left>?s ?p ?o</left>
  <right>
    ?s ?p ?node . ?node :hasValue ?o . ?node :hasTime ?i .
    ?i :hasBeginning?d1 . ?i :hasEnd ?d2
  </right>
  <cost>0</cost>
  <explanation>
    Adapt the triple pattern to the n-ary structure to re-
    -trieve the value and temporal element associated to ?p.
  </explanation>
</rule>
```

(b) The transformation rule r_2.

Figure 7. Transformation rules r_1 and r_2.

$\mathcal{Q}_{h_2}$ =

```
SELECT ?v ?d1 ?d2
WHERE {
    ?x firstName "Henri" .
    ?x familyName "Poincaré" .
    ?x livesAt ?node .
    ?node hasValue ?v .
    ?node hasTime ?i1 .
    ?i1 hasBeginning ?d1 .
    ?i1 hasEnd ?d2
}
```

6.3. *Reasoning with temporal data*

The choice of OWL-Time ontology, by integrating Allen's interval algebra, is used to represent relations between different temporal elements associated with facts from the Henri Poincaré corpus. In some cases, temporal elements are missing because the available sources do not allow for the estimation of these elements. It also happens that some known information is not yet represented in the knowledge base. As a result, when a user queries the corpus using the SPARQL language (through the form-based interface), several resources may not be returned by the system when they could provide possible answers to a research problem. One idea to overcome this issue is to set up customized inference rules that would rely on ontology knowledge. To this end, a new transformation rule language, named RTRL (RDF Transformation Rule Language), is introduced to create entailment rules. Let $\mathcal{G}_{ex}$ be the following example of RDF graph:

$\mathcal{G}_{ex}$ = {⟨letter11 sentBy henriPoincaré⟩,
⟨letter11 sentTo göstaMittagLeffler⟩,
⟨letter11 hasWritingDate t1⟩,
⟨letter22 sentBy göstaMittagLeffler⟩,
⟨letter22 sentTo henriPoincaré⟩,
⟨letter22 hasWritingDate t2⟩,
⟨t2 hasTime 1883⟩,
⟨letter11 hasReply letter22⟩}

Imagine a historian doing a search on this graph using the form-based interface and formulating the following query:

$\mathcal{Q}_{1885}$ = Give letters exchanged between Henri Poincaré and Gösta Mittag-Leffler before 1885.

The system must return `letter22` because its date of writing is known and corresponds to the constraint defined in $\mathcal{Q}_{1885}$. On the other hand, `letter11` is not returned because its date of writing is unknown to the system. However, the database contains an important statement: `letter22` replies to `letter11`. It is therefore possible to deduce that the date of writing of `letter11` is earlier than that of `letter22`, and therefore earlier than 1885. One way to automate this reasoning is to make the relationship between the two dates of writing explicit. For this, it is necessary to add to the database the triple ⟨`t1 before t2`⟩.

To make this type of reasoning automatic, a new type of rule is introduced for temporal RDF graph entailment. For the running example, the $r_{hasReply}$ rule (Figure 8) is used to add relationships between the temporal elements related to dates on which letters in the correspondence were written. This new type of rules introduces the `new` field which replaces the `left` and `right` fields. This field corresponds to the RDF triples to be added to the knowledge base.

This entailment mechanism is useful for specifying relationships between temporal elements. It is based on the definition of RTRL rules whose applications allow

```
<rule name="Has reply">
  <context>
    ?l1 hasWritingDate ?t1 . ?l2 hasWritingDate ?t2 .
    ?l1 hasReply ?l2
  </context>
  <new>?t1 before ?t2</new>
  <explanation>
    Add relation between temporal elements
    related to letters writing dates.
  </explanation>
</rule>
```

Figure 8. An example of RTRL rule.

the addition of triples to RDF bases. It is noteworthy that this language can also express the RDFS inference rules such as $\frac{\langle x \text{ a } C\rangle \ \langle C \text{ subc } D\rangle}{\langle x \text{ a } D\rangle}$. The implemented system prevents the application of rules that would lead to the addition of triples already existing in the knowledge base. This section presented an example related to the dates on which letters were written, but other rules have been defined for the Henri Poincaré correspondence corpus. For example, one rule has the purpose of restraining the possible date of writing of a scientific article based on the dates on which letters mentioning it were written. Another rule aims at specifying the time intervals during which Henri Poincaré and others taught jointly within the same institution.

7. Conclusion

Representing historical corpus knowledge requires the integration of temporal elements. For corpora indexed using Semantic Web technologies, this can be difficult because the RDF model is based on binary relations. Different models have been proposed to include temporal data, some requiring an extension of standard Semantic Web models. For the exploitation of the corpus related to the correspondence and works of Henri Poincaré, it was decided to use *n*-ary representations. This type of representation meets the needs stated in the introduction of this article: it describes a fact with temporal elements (instants or intervals); it does not require an extension to RDF and it allows the use of the SPARQL query language. Moreover, by associating this representation with elements of the OWL-Time ontology, W3C standard, including the representation of Allen's interval algebra, it is possible to carry out reasoning on corpus temporal data.

As part of this work, a SPARQL query transformation mechanism is used to simplify the generation of queries involving *n*-ary representations. Associated with a form-based interface, this system allows users to ignore SPARQL syntax and simply retrieve a fact with or without its temporal element. An extension of the SQTRL tool also makes it possible to entail temporal RDF graphs with custom rules that exploit the semantics of ontology properties. The idea is to make explicit knowledge that can be released by the application of inferences with the use of properties of the OWL-Time ontology.

The work carried out for the integration and the exploitation of temporal knowledge has made possible a more accurate representation of facts related to the Henri Poincaré correspondence corpus. For instance, it is now possible to represent that a individual has held different roles within various learned societies by specifying the associated time frames. The form-based interface could be used to retrieve temporal information associated with members positions in various institutions. A historian interested in the evolution of an institution may also use the form-based interface to find letters exchanged with a correspondent while he/she was a member of the institution.

One of the advantages of choosing the *n*-ary model is that this structure can also be reused to describe a fact using additional elements. In particular, it would be relevant to associate a point of view to some of the facts in the knowledge base. Incorporating this element of subjectivity could help to represent knowledge more accurately. Another future work concerns the representation of uncertain knowledge. In history, it is often complex to define the date of an event. In the context of the Henri Poincaré corpus, this may be the date of a union, a nomination, the date on which a letter was written, etc. A historian must make choices to guarantee the coherence of narratives, but it would be relevant to associate with the representation of a fact an uncertain time that would account for the information as perceived by the historian. Work has been carried out to integrate fuzziness during the representation of temporal knowledge. In particular, an OWL-Time extension serves to describe uncertain time intervals using the properties `fuzzyHasBegining` and `fuzzyHasEnd` [38].

Acknowledgements

This work was supported partly by the French PIA project “Lorraine Université d'Excellence”, reference ANR-15-IDEX-04-LUE.

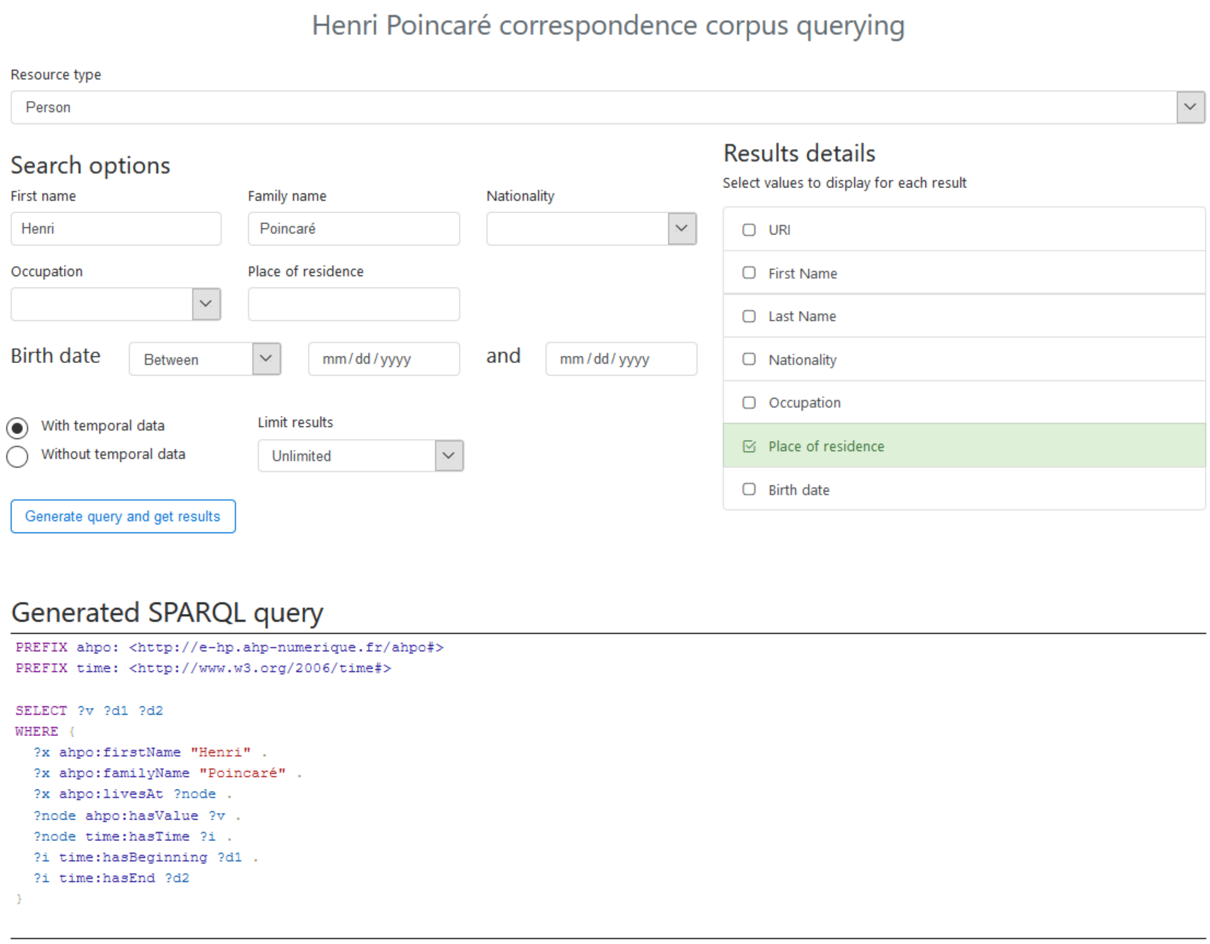


Figure 9. Form-based interface used to generate SPARQL queries (note the use of radio buttons to choose to generate a query giving — or not — temporal data).

References

[1] H. Poincaré, *La Science et l'Hypothèse*, Flammarion, 1902.

[2] H. Poincaré, *La Valeur de la Science*, Flammarion, 1905.

[3] H. Poincaré, *Science et Méthode*, Flammarion, 1908.

[4] L. Rollet (ed.), *La correspondance de jeunesse d'Henri Poincaré : les années de formation. De l'École polytechnique à l'École des Mines (1873-1878)*, Publications of the Henri Poincaré Archives, Birkhäuser, 2017. doi:10.1007/978-3-319-55959-9.

[5] P. Nabonnand (ed.), *La correspondance entre Henri Poincaré et Gösta Mittag-Leffler*, Birkhäuser, Basel, 1998.

[6] C. Boulaire and R. Carabelli, Du digital naive au bricoleur numérique : les images et le logiciel Omeka, in: *Expérimenter les humanités numériques. Des outils individuels aux projets collectifs*, É. Cavalié, F. Clavert, O. Legendre and D. Martin, eds, Les Presses de l'Université de Montréal, Montréal, Québec, 2017, pp. 81–103, Chapter 7.

[7] N. Larrousse and J. Marchand, A Techno-Human Mesh for Humanities in France: Dealing with preservation complexity, in: *DH 2019*, Pierazzo, Elena and Ciotti, Fabio, ed., Utrecht, Netherlands, 2019. https://hal.archives-ouvertes.fr/hal-02153016.

[8] O. Bruneau, N. Lasolle, J. Lieber, E. Nauer, S. Pavlova and L. Rollet, Applying and Developing Semantic Web Technologies for Exploiting a Corpus in History of Science: the Case Study of the Henri Poincaré Correspondence, *Semantic Web – Interoperability, Usability, Applicability* **12**(2) (2021), 359–378. https://hal.univ-lorraine.fr/hal-02879820.

[9] F. Manola, E. Miller, B. McBride, G. Schreiber and Y. Raimond, RDF 1.1 Primer, W3C Recommendation, 2014, Last consultation: March 2021. https://www.w3.org/TR/rdf11-primer/.

[10] G. Carothers and E. Prud'hommeaux, RDF 1.1 Turtle, 2014, Last consultation: March 2021. http://www.w3.org/TR/2014/REC-turtle-20140225/.

[11] D. Brickley and R.V. Guha, RDF Schema 1.1, W3C recommendation, 2014, Last consultation: March 2021. https://www.w3.org/TR/rdf-schema/.

[12] P. Hitzler, M. Krötzsch, B. Parsia, P.F. Patel-Schneider, S. Rudolph et al., OWL 2 Web Ontology Language Primer, W3C recommendation, Last consultation: March 2021.

[13] E. Prud'hommeaux, SPARQL Query Language for RDF, W3C Recommendation, 2008, Last consultation: March 2021. http://www.w3.org/TR/rdf-sparql-query/.

[14] A. Prost, *Douze Leçons sur l'histoire*, Points Histoire, Éditions du Seuil, 1996.

[15] C.D. Conner, *A People's History of Science: Miners, Midwives, and Low Mechanicks*, Nation books, 2005.

[16] H. Zinn, *A People's History of the United States*, Harper & Row, 1980.

[17] R. Koselleck, *The Practice of Conceptual History: Timing History, Spacing Concepts*, Cultural Memory in the Present, Stanford University Press, 2002.

[18] A. Gilroy and W.M. Verhoeven, *Epistolary histories: letters, fiction, culture*, University of Virginia Press, 2000.

[19] M. Schiavon and L. Rollet, *Pour une histoire du Bureau des longitudes (1795-1932)*, Presses universitaires de Nancy-Éditions universitaires de Lorraine, 2017.

[20] P.A. Carré, Expositions et modernité : Electricité et communication dans les expositions parisiennes de 1867 à 1900, *Romantisme* **19**(65) (1989), 37–48.

[21] C. Gutierrez, C. Hurtado and A. Vaisman, Temporal RDF, in: *European Semantic Web Conference*, Vol. 3532, A. Gómez-Pérez and J. Euzenat, eds, Springer, 2005, pp. 93–107.

[22] C. Gutierrez, C.A. Hurtado and A. Vaisman, Introducing time into RDF, *IEEE Transactions on Knowledge and Data Engineering* **19**(2) (2006), 207–218.

[23] N. Noy, A. Rector, P. Hayes and C. Welty, Defining n-ary Relations on the Semantic Web, Vol. 12, 2006, Last consultation: March 2021. https://www.w3.org/TR/swbp-n-aryRelations/.

[24] O. Hartig, Foundations of RDF* and SPARQL*:(An alternative approach to statement-level metadata in RDF), in: *AMW 2017 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, Montevideo, Uruguay*, Vol. 1912, J. Reutter and D. Srivastava, eds, 2017.

[25] J.J. Carroll, C. Bizer, P. Hayes and P. Stickler, Named Graphs, Provenance and Trust, in: *Proceedings of the 14th International Conference on World Wide Web*, A. Ellis and T. Hagino, eds, WWW '05, Association for Computing Machinery, New York, New York, USA, 2005, pp. 613–622–.

[26] B. Schueler, S. Sizov, S. Staab and D.T. Tran, Querying for Meta Knowledge, in: *Proceedings of the 17th International Conference on World Wide Web*, J. Huai, R. Chen, H. Hon and Y. Liu, eds, WWW '08, Association for Computing Machinery, New York, USA, 2008, pp. 625–634–.

[27] C. Welty and R. Fikes, A Reusable Ontology for Fluents in OWL, in: *Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*, Vol. 150, Bennett, Brandon and Fellbaum, Christiane, ed., IOS Press, 2006, p. 226.

[28] H.-U. Krieger, B. Kiefer and T. Declerck, A Framework for Temporal Representation and Reasoning in Business Intelligence Applications, in: *AAAI Spring Symposium: AI Meets Business Rules and Process Management*, K. Hinkelmann, ed., 2008, pp. 59–70.

[29] M. Doerr, The CIDOC conceptual reference module: an ontological approach to semantic interoperability of metadata, *AI magazine* **24**(3) (2003). doi:10.1609/aimag.v24i3.1720.

[30] V. Nguyen, O. Bodenreider and A. Sheth, Don't like RDF Reification? Making Statements about Statements Using Singleton Property, in: *Proceedings of the 23rd International Conference on World Wide Web*, C.-W. Chung, ed., WWW '14, Association for Computing Machinery, New York, USA, 2014.

[31] V. Nguyen, O. Bodenreider, K. Thirunarayan, G. Fu, E. Bolton, N. Queralt-Rosinach, L.I. Furlong, M. Dumontier and A.P. Sheth, On Reasoning with RDF Statements about Statements using Singleton Property Triples, *Computer Research Repository* **abs/1509.04513** (2015). http://arxiv.org/abs/1509.04513.

[32] O. Udrea, D.R. Recupero and V.S. Subrahmanian, Annotated RDF, *ACM Transactions on Computational Logic (TOCL)* **11**(2) (2010), 1–41.

[33] F. Zhang, K. Wang, Z. Li and J. Cheng, Temporal Data Representation and Querying Based on RDF, *IEEE Access* **7** (2019), 85000–85023.

[34] S. Cox, C. Little, J.R. Hobbs and F. Pan, Time Ontology in OWL, 2020, Last consultation: March 2021. https://www.w3.org/TR/owl-time/.

[35] V. Milea, F. Frasincar and U. Kaymak, tOWL: a temporal web ontology language, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **42**(1) (2011), 268–281.

[36] P.F. Diallo, O. Corby, I. Mirbel, M. Lo and S.M. Ndiaye, HuTO: une Ontologie Temporelle Narrative pour les Applications du Web Sémantique, in: *26es Journées francophones d'Ingénierie des Connaissances*, Rennes, France, 2015.

[37] O. Bruneau, E. Gaillard, N. Lasolle, J. Lieber, E. Nauer and J. Reynaud, A SPARQL Query Transformation Rule Language — Application to Retrieval and Adaptation in Case-Based Reasoning, in: *Case-Based Reasoning Research and Development. ICCBR 2017*, D. Aha and J. Lieber, eds, Lecture Notes in Computer Science, Springer, Cham, 2017, pp. 76–91.

[38] F. Ghorbel, F. Hamdi, E. Métais, N. Ellouze and F. Gargouri, A Fuzzy-Based Approach for Representing and Reasoning on Imprecise Time Intervals in Fuzzy-OWL 2 Ontology, in: *Natural Language Processing and Information Systems*, M. Silberztein, F. Atigui, E. Kornyshova, E. Métais and F. Meziane, eds, Springer, Cham, 2018, pp. 167–178.