Temporal Modelling in Cultural Heritage Knowledge Graphs: Use Cases, Requirements, Evaluation, and Decision Support

Oleksandra Bruns[1]
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Eggenstein-Leopoldshafen, Germany
Karlsruhe Institute of Technology (AIFB), Germany

Jörg Waitelonis
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany
Karlsruhe Institute of Technology (AIFB), Germany

Jeff Z. Pan
The University of Edinburgh, Scotland

Harald Sack
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany
Karlsruhe Institute of Technology (AIFB), Germany

Abstract

Our culture, history and world are in constant motion, continuously shaped by the flow of time, evolving narratives, and shifting relationships. Capturing this temporal complexity within cultural heritage (CH) knowledge graphs is essential for preserving the dynamic nature of human heritage. However, standard RDF predicates fail to effectively model the temporal aspects of cultural data, such as changing facts, evolving relationships, and temporal concepts. Over the past two decades, a variety of RDF-based approaches have been proposed to address this limitation, yet guidance is missing on which method best suits specific CH contexts. This paper presents a systematic evaluation of temporal RDF modelling approaches from a CH perspective. Based on an analysis of real-world CH use cases, core temporal requirements are identified that reflect both modelling expressivity and practical concerns. Six prominent approaches – RDF*, tRDF, Named Graphs, Singleton Property, N-ary Relations, and 4D Fluents – are assessed across these requirements. Our findings reveal that no single solution fits all scenarios, but suitable approaches can be selected based on project-specific priorities. To support practitioners, a decision-support tool is introduced to guide them in selecting the most suitable extension for their specific needs. This work provides practical guidance for CH modelling and contributes to the broader development of temporally aware Linked Data.

2012 ACM Subject Classification Information systems → Information integration; Computing methodologies → Knowledge representation and reasoning; Applied computing → Arts and humanities

Keywords and phrases Temporal Data Representation, RDF Extensions, Cultural Heritage, Knowledge Graphs

Digital Object Identifier 10.4230/TGDK.4.1.2

Category Survey

Supplementary Material
DataPaper: `https://github.com/sashabruns/RDF-Extensions-for-Cultural-Heritage`

Funding *Oleksandra Bruns*: acknowledges support from the Leibniz Association under project number SAW-2020- FIZ KA-4-Transraz.

Received 2024-10-31 **Accepted** 2026-03-09 **Published** 2026-03-31

---

[1] corresponding author

Transactions on Graph Data and Knowledge, Vol. 4, Issue 1, Article No. 2, pp. 2:1–2:46
Transactions on Graph Data and Knowledge
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Our past is not simply a collection of static facts and objects; it is a dynamic, evolving story where the context, relationships, and interpretations change over time. Historical events unfold over specific periods, social networks among persons evolve, cultural objects and terms gain new meanings, and jurisdictional practices and societal norms also change, continually reshaping how humanity understands past events. Preserving and analysing this web of changes provides deeper insight into our culture, society, and the world, revealing a more nuanced understanding of history.

Understanding time has been a central theme in philosophical discussions for centuries and continues to influence practical domains today, including the representation of time in knowledge graphs (KGs). In this context, KGs are graph-based structures designed to represent real-world knowledge, where nodes denote entities of interest and edges define various types of relationships between them [36]. How time is understood and contextualised impacts the methods and approaches used to model temporal data, affecting how evolving information within KGs is captured, reasoned about, and interpreted. For example, in [44], the distinction between the A- and B-theories of time has been introduced. The A-Theory of Time suggests that only the present is real, with the past and future existing merely as conceptual extensions. In contrast, the B-Theory posits that all points in time, past, present, and future, are equally real and exist simultaneously. Thus, when representing temporality within KGs, if aligned with A-Theory, models focus on versioning mechanisms, where the current state of entities and their relationships is actively tracked and updated. While models influenced by B-Theory adopt a snapshot-based representation, where models aim to provide a comprehensive view of all time points, enabling users to query and analyse data across different periods. Complementing this philosophical grounding, recent work on the evolution of open KGs [53] explores how temporal change is modelled and managed in real datasets. The study investigates practical perspectives of different modelling patterns, e.g. versioning and snapshotting, and offers a valuable technical interpretation of these philosophical distinctions in the context of KG evolution.

Similarly, the debate between Endurantism and Perdurantism evolves around the existence of objects over time [33]. Endurantism argues that objects are wholly present at each moment in time. In KGs, such entities are represented as static over time with evolving attributes. On the contrary, Perdurantism suggests that objects are spread across time, existing as a series of temporal parts. This has been formalised in foundational ontologies such as DOLCE [24], which explicitly classifies entities into endurants (e.g., physical objects or people) and perdurants (e.g., events or processes) to model the dynamic nature of reality.

Cultural heritage (CH) KGs are powerful tools for capturing, integrating and uncovering new insights from complex historical data [42, 40, 68, 13, 18]. By incorporating temporal information into these KGs, they can track changes over time, help users understand historical contexts, and uncover trends, leading to more sophisticated reasoning, better decision making, and deeper analysis. For example, understanding when an event took place and its relations to other events can be used to predict future outcomes or to reveal previously unknown details of the past.

Representing temporal context in CH KGs is crucial for several reasons, for example [67, 15, 65]:

- **Better understanding of temporal relationships**: In CH use cases, modelling temporal relationships enables more precise and nuanced historical interpretation. Accurately capturing when events occurred, how they overlap or follow one another, and how relationships evolve (e.g., changes in artistic movements) is essential for understanding narratives, context, and influence.
- **Gaining new insights**: Knowledge stored in CH KGs often comes from sources that are incomplete, fragmented, or disputed. Temporal modelling allows for reasoning over such data to infer possible sequences, detect contradictions, or identify missing links. For instance, when

reasoned over CH data, new connections between events or individuals not explicitly recorded can be detected. This aligns with the needs of historians, archivists, and scholars who work with uncertain or conflicting temporal accounts – for instance, when different historical sources assign different dates or periods to the same event.

- **Enhanced data integration**: CH data comes from diverse and distributed sources – archives, museums, libraries, galleries – and spans centuries. Temporal annotation supports the integration of such heterogeneous data by aligning information across different time periods and evolving terminologies. For instance, an entity may undergo e.g., changes in name, function, or classification over time; or different narratives may emerge around the same historical event with new sources uncovered with the time. Temporal modelling enables consistent representation of such information, ensuring that historical perspectives are (accurately) captured and understood.
- **More accurate provenance**: In CH, temporal (meta)data enables KGs to represent competing or evolving claims, e.g., conflicting dates for a historical event described in various sources. It also supports tracking how interpretations and representations change over time. For example, a stage performance of a classical play such as Antigone might be portrayed differently in the 1930s and the 1950s: under authoritarian regimes, the performance was likely shaped by censorship or propaganda. Temporal modelling captures these shifts, documenting not just what was performed, but how, when, and under which historical circumstances, preserving the meanings that evolve over time.

The Resource Description Framework (RDF) [22] is a widely adopted representation language and exchange format used to encode data on the Web and is a fundamental building block for creating KGs. However, due to its binary triple structure, the standard RDF model cannot natively express context in which a statement holds. Temporality is one type of such contextual information (other examples include location, provenance, certainty, source attribution, perspective etc.). Yet, temporal context comes with specific modelling challenges: it often involves complex structures such as intervals, durations, temporal relations (e.g., before, during), vague or conflicting time references. These aspects are especially prominent in CH applications, where representing temporal knowledge is not just a matter of attaching timestamps, but of capturing historical change, overlapping narratives, evolving interpretations, etc. As a result, various extensions to RDF have been proposed to enable temporal representation. Each of these extensions has its own advantages and limitations, and the choice of the approach depends on the specific requirements of a use case, balancing query simplicity, semantic expressivity, compatibility with existing RDF tools, and scalability. However, despite this variety of available temporal extensions, there remains a critical gap in understanding how well these models address the unique needs of specific application domains – particularly CH, which is characterised by highly heterogeneous and often ambiguous historical data, where facts are uncertain, timelines are incomplete, the meaning of entities evolves over time etc.

While prior efforts have introduced and compared various temporal extensions to RDF [57, 73, 78, 76], they often remain disconnected from concrete applications. By answering these research questions:

- **RQ1.** What RDF-based approaches exist for temporal representation?
- **RQ2.** What temporal requirements arise from CH use cases?
- **RQ3.** How intuitive and usable are different temporal modelling approaches for CH domain experts?
- **RQ4.** How can temporal modelling approaches be evaluated based on CH-specific requirements and technical criteria?
- **RQ5.** How can CH practitioners be supported in selecting a suitable temporal modelling approach?

this work aims to provide a systematic analysis that translates the abstract capabilities of RDF extensions into practical implications for representing time in CH KGs. In particular, it presents a comprehensive research study focused on identifying, structuring, and evaluating RDF-based temporal modelling approaches from a CH-specific perspective. The analysis is grounded in real-world case studies, deriving domain-specific temporal requirements, and developing a decision-support system to guide practitioners. The results aim to inform not only CH projects but also broader domains facing similar challenges in managing temporality in knowledge representation.

The remainder of the paper is structured as follows. Section 2 introduces representative CH use cases and derives a set of temporal modelling requirements. Section 3 reviews related work on temporal modelling in RDF and CH. Section 3.3 presents the RDF-based approaches evaluated in this study. Section 4 details the comparative assessment of these approaches against the identified requirements. Section 5 introduces the decision-support tool for selecting suitable temporal modelling strategies in CH contexts. Finally, Section 6 summarises the findings and outlines directions for future research.

2 Use Cases and Requirements

The modelling of time in CH KGs presents distinctive challenges coming from the nature of historical data and its interpretation. To understand these challenges and the corresponding modelling needs, this section introduces a set of representative CH use cases. Through their analysis, core temporal characteristics and complexities are identified, and a set of domain-specific requirements is derived. These requirements provide a foundation for the subsequent evaluation of temporal modelling for CH KGs.

In [15, 13, 71, 66, 70, 67], initial efforts towards defining challenges and requirements for representing temporal information in CH use cases are conducted[2]. However, these studies examined the use cases independently, each focusing on specific aspects within their respective projects — TRANSRAZ[3], Wiedergutmachung[4], “Subject Related Points of Access within Archivportal-D on Example of the Subject Area Weimar Republic”[5], and Linked Stage Graph[6]. While these studies provide valuable insights, they approach the temporal challenges from isolated perspectives, tailored to project-specific objectives. In this work, these prior findings are systematically consolidated and extended, offering a harmonised view across use cases.

2.1 Case Study: Cultural Heritage

It is acknowledged that the use cases discussed here represent only a fraction of the CH domain, and other subdomains, such as musicology, may pose additional challenges and require different adaptations in the context of temporality. Nonetheless, this structured analysis is a necessary step towards understanding temporal aspects of CH data and providing means for evaluating the existing approaches.

---

[2] The use cases presented in this section were selected based on their origin in distinct, real-world CH research projects that the authors were actively involved in.

[3] `https://www.fiz-karlsruhe.de/en/forschung/transraz`

[4] `https://www.fiz-karlsruhe.de/en/forschung/wiedergutmachung`

[5] `https://www.fiz-karlsruhe.de/en/forschung/archivportal-d-sachthematische-zugaenge`

[6] `https://slod.fiz-karlsruhe.de/`

Use Case 1: Connecting Time Layers of Nuremberg's History

The TRANSRAZ project aims to reconstruct the city of Nuremberg by integrating and linking historical information from different time periods. The project seeks to create a comprehensive and dynamic 3D model of the city, enriched with semantically connected historical data that spans centuries, enabling historians, researchers, and the general public to explore Nuremberg's history interactively, with the ability to navigate through different time periods and understand the city's evolution.

Data and Challenges. The dataset includes a wide variety of historical resources such as address books, biographical articles, research papers, historical maps, city plans, and others. One major challenge lies in modelling how the roles and functions of places or people change over time. For instance, a building served as a bakery in one century and was later converted into a school, even while retaining the same location or name. Another frequent issue is the renaming of streets or places due to political/administrative decisions. Capturing these changes while maintaining continuity in identity requires labelling associated entities with time validity.

Similarly, historical records describe people or institutions in evolving roles or under different names, complicating how such entities are consistently modelled across time. Precise dating is not always available. Sources provide full dates (e.g., "15 July 1892") or vague references (e.g., "during his studies" or "until 1923"), requiring the ability to represent and distinguish between exact timestamps, intervals, and intervals with missing boundaries. Furthermore, the data includes temporally overlapping or sequential events, such as changes in residency, profession, or ownership. This requires capturing not only when things occurred, but also how different events relate to each other chronologically – whether they overlap, follow one another, etc. Finally, much of the data is uncertain, approximate, or even conflicting across sources. For example, different documents provide inconsistent information about a person's occupation or address at a given time. A temporal model must support the representation of multiple interpretations, including metadata about the source and temporal validity of each claim.

Use Case 2: Temporal Alignment in Archival Records

Within the "Wiedergutmachung" project, the focus lies on reconstructing the processes of reparation and social transformation in post-war Germany, particularly after the fall of the Nazi regime. The project examines how restitution and compensation were handled from 1945 to the 1970s, based on a vast array of archival records that document these complex processes. This project is especially crucial as it aims to document the experiences of the victims of Nazi persecution, capturing their personal and family histories as narrated in their reparation claims.

Data and Challenges. The dataset includes extensive archival records spanning from 1945 to the 1970s, with millions of individual case files detailing the experiences and claims of those who suffered under the Nazi regime. These records are spread across various German archives and consist of personal records, organisational information, and detailed accounts of the reparation processes. The primary goal of this project is to reconstruct the timelines of people's lives, significant events, and legal processes by capturing and accurately integrating temporal information from these diverse records. This involves aligning data to reflect changes in personal details, organisational structures, and the progression of reparation processes over time.

Such records contain a large amount of temporal information that needs to be modelled. The primary challenge lies in the need to represent each fact along with its temporal validity and provenance. While resolving conflicting or ambiguous information is outside the scope of the representation model itself, the model must still support the accurate and consistent encoding of changes over time. For example, if an individual is referred to as *"Löwental"* in an archival record from 1948 and as *"Friedrich"* in a document from 1950, a temporal model must be able to express both name variants, associate them with the correct time intervals, and link them to their respective sources. In this way, the temporal evolution of facts, as well as their provenance, can be tracked and maintained within the knowledge graph.

Moreover, reparation processes are inherently temporal and evolve over time. Claims often span multiple legal stages, involve different institutions, and result in varying outcomes depending on case-specific conditions. A claim investigated in 1951 under one administration led to a denial, while the same claim reevaluated in 1954 – after a policy reform – was accepted.

Capturing such processes requires the ability to model sequences of events (e.g., investigation followed by decision and payment), overlaps (e.g., parallel claims by family members), and dependencies between events (e.g., the decision phase relying on the completion of a medical evaluation). In addition, the historical records often contain temporal uncertainty and conflict. Some documents specify exact dates (e.g., the filing of a claim), while others reference vague or conflicting time frames (e.g., "shortly after the war").

Use Case 3: Dynamics in Archival Classification Scheme

The project "Subject Related Points of Access within Archivportal-D on Example of the Subject Area Weimar Republic" focuses on the digitisation and classification of archival records from the Weimar Republic. Unlike the previous use cases, which aim to extract and represent content data, this project is designed to assist archivists in the digitisation and classification process. Temporal aspects are managed through a complex dynamic classification scheme that evolves as new records are digitised and integrated into the archive. For archivists, it is crucial to track the digitisation process and how the classification scheme has developed over time. This allows them to understand the evolution of archival categorisation, ensure consistency, and make informed decisions about organising and accessing historical records as new data is added.

Data and Challenges. The dataset includes over 20,000 digitised archival records, with the digitisation process still ongoing. These records encompass government documents, legal texts, personal correspondences, and other materials from the Weimar Republic era. To collect, store long-term, and make historical records accessible to all, archival resources must be categorised for structured, intuitive search and exploration.

The main challenge is that the classification scheme, which comprises around 900 specialised keywords, 20 categories, and 130 subcategories, is not static. It must be continuously adapted to accommodate new records and additional resources from other archives. This dynamic nature of the classification system requires careful modelling to ensure that it can evolve over time without losing relevance or accuracy. For instance, as new records are added, previously used keywords might be eliminated, or certain documents might be reclassified, reflecting changes in understanding or context. Additionally, it is important for archivists to be able to keep track of all versions of the classification scheme, and allow researchers to analyse the evolution of document classification. This could lead to insights such as which keywords were most often eliminated or which documents were the most reclassified.

While the challenges of managing evolving taxonomies are well-known in archival and library sciences [34], this use case presents a distinct modelling task: capturing classification scheme dynamics as structured, queryable knowledge. Rather than maintaining a single live classification scheme, the aim is to construct a KG that reflects the validity of terms over time, supports time-based retrieval, and enables the analysis of concept evolution within the taxonomy itself. This extends traditional taxonomy management into the temporal KG domain, where the classification system is not only curated but semantically modelled as a historical and dynamic object.

Use Case 4: Linked Stage Graph for Performing Arts

The Baden-Württemberg State Archives[7] in Germany opened up their data to create a Linked Stage Graph, facilitating the exploration and analysis of historical performance data. The initiative aims to provide insights into theatrical performances during a time marked by significant social and political events in Germany, highlighting the context and evolution of stage design, actors, and performances.

Data and Challenges. The dataset includes 7,000 historical black and white photographs and EAD-XML metadata covering the period from the 1890s to the 1940s. The goal is to reveal political and social context of performing arts in Germany, e.g. to research which theatre performances were allowed during challenging times and how certain characters were depicted.

A central challenge in this use case lies in representing how elements of the performing arts evolve over time. Theatrical works are temporal: performances occur on specific dates, while actors, stage designs, and interpretations change across decades. For example, Bertolt Brecht's plays were interpreted very differently in the 1920s compared to their postwar productions. A particular character, like Antigone, was often portrayed as a tragic figure of resistance in one period and as a passive symbol of duty in another, shaped by political censorship and cultural movements. Capturing such evolution requires modelling not just entities like actors or plays, but also their changing roles and meanings over time. Moreover, descriptive terms used in archival metadata often shift. Stage roles are described using different terminology ("lead actor" versus "main character"), and theatre venues are renamed or repurposed. These terminological and functional changes must be represented. Additionally, performances are documented with varying degrees of temporal precision. Some are precisely dated, while others are loosely described as occurring "in the early 1930s" or "around 1940". Capturing temporal relationships between (performance-)related events is essential for situating theatrical life in its full socio-political context. For example, modelling that one actor's portrayal of Antigone ended in 1933 and was succeeded by another can reflect not only the change itself but also historical circumstances such as institutional changes or political pressure. Similarly, a premiere coinciding with the rise of National Socialism can significantly affect how a play was staged and received.

2.2 Requirements

The temporal modelling needs in CH projects are shaped by the way knowledge is produced, interpreted, and preserved over time. Unlike many domains where temporal data can be precise and well structured, CH data are often incomplete, uncertain, and deeply contextual. It must capture not only when entities or events occurred but also how they evolved, how they relate across time, and how they are interpreted differently in varying historical narratives. The complexity of this temporal dimension is further compounded by the diversity of technical infrastructures and data practices across CH institutions.

To identify the core requirements for temporal modelling in CH, the data and needs of real-world use cases presented in Section 2.1 are analysed. These requirements reflect both i) how well a model can express temporal phenomena and ii) how feasible and usable a model is in practice. While inspired in part by broader literature on knowledge graph evolution (e.g. [53]), our focus remains on the specific constraints and characteristics of CH datasets. The result is a set of requirements that guide the evaluation of temporal modelling approaches for CH use cases.

---

[7] `https://www.landesarchiv-bw.de/de/en/68809`

- **REQ1: Temporal Expressivity** is concerned with the granularity and richness with which temporal semantics can be described. It serves as the foundation for assessing the semantic adequacy of temporal modelling approaches in the CH domain and include:
  - **REQ1.1: Contextual Change** captures the need to model how entities and their descriptions evolve over time. Such changes may involve the functional roles of entities (e.g., a person becomes a director), their classification (e.g., a building's function changes from residential to commercial), naming conventions, or how they are interpreted in different historical contexts. Crucially, the identity of the described entity remains stable, while its representation is adapted across multiple temporal snapshots. The model must therefore support multiple temporally-scoped descriptions of the same entity and preserve the validity period of each[8].
  - **REQ1.2: Temporal statements.** Statements about the world, such as a person's role, a building's function, or an institutional relationship, are rarely static in CH data. This requirement involves the ability to associate such statements with explicit temporal scopes. The modelling approach must allow for the attachment of time intervals (or instants) to individual assertions in a way that distinguishes which statements held true at which times. This enables accurate reconstruction of timelines and state changes over historical periods.
  - **REQ1.3: Temporal Uncertainty and Incompleteness.** Historical records frequently contain vague, approximate, or partially missing temporal information. Dates may be imprecise (e.g., "circa 1900"), incomplete (e.g., "before 1935"), or based on subjective interpretations. A temporal model must provide vocabulary that can represent such uncertainty, without reducing it to artificially precise or misleading constructs.
  - **REQ1.4: Interval Relations.** Beyond isolated timestamps or intervals, CH research often requires understanding the sequence, overlap, or causality of events. This requirement calls for supporting qualitative temporal relations between time intervals or events (e.g., before, meets, overlaps) as well as reasoning over those relations. It enables the reconstruction of event timelines and dependencies, such as legal or historical developments, and supports the integration of events into broader historical narratives.

However, semantic expressivity alone is not sufficient to ensure practical applicability. In real-world CH projects, the deployment of temporal knowledge graphs must also satisfy technical, operational and usability requirements. These additional dimensions reflect recurring concerns raised during the implementation and integration phases of CH projects in which the authors have been involved, especially in discussions with domain partners on performance constraints, tool support, and user interaction. Furthermore, these structural aspects align with established concerns in the Semantic Web literature (see Section 3.4).

- **REQ2: Storage and Scalability**: In CH, where datasets often grow to substantial sizes due to the rich historical details they contain, scalability is essential. Efficient representation of temporal information plays a key role in enabling systems to handle large volumes of data without sacrificing performance. Approaches that produce excessive structural overhead may pose challenges for storage, querying, and reasoning at scale.
- **REQ3: RDF Syntax Extension**: Many temporal models extend the standard RDF syntax to incorporate a temporal dimension, which can influence compatibility with existing RDF frameworks and, e.g. reasoning. This requirement aims to provide an overview to the

---

[8] Contextual change, as used here, is different from semantic shift in ontological terms: it does not imply changes in the meaning of classes or properties themselves, but rather changes in the description of instances under evolving circumstances.

extension, considering that it may directly influence the choice of triple store, reasoner, etc. Approaches that introduce minimal but effective extensions are often preferred, as they maintain compatibility with standard RDF tools and infrastructures, leading to easier integration into existing systems.

- **REQ4: Model Semantics**: This requirement assesses the level of semantic expressiveness required by a modelling approach to effectively represent the information. Specifically, it examines whether the approach relies on basic RDF(S) semantics or if it requires more advanced semantic frameworks, rules or even custom logic to capture additional constraints or specialised concepts that are often introduced in temporal approaches.
- **REQ5: Query Language**: In many CH projects, querying is not just a technical detail but a practical constraint shaped by the institutional tooling in place. CH institutions often rely on existing SPARQL endpoints for accessing and maintaining their data, and these endpoints may not support advanced SPARQL extensions or custom query languages. As a result, the ability to use standard SPARQL becomes not only a matter of usability but also a strict requirement imposed by the technical infrastructure.
- **REQ6: Use of RDFS/OWL Reasoning**: This requirement investigates whether the modelling approach preserves compatibility with OWL/RDFS reasoning, and to what extent temporal constructs can participate in standard inference processes. Evaluation considers whether RDFS- and OWL-based entailment, e.g., class membership, property inheritance, inverse properties, or domain/range inference, can still be performed over temporally enriched data.
- **REQ7: Ease-of-Use**: In the CH domain, users are often domain experts with limited technical expertise, particularly in RDF and querying. Therefore, it is crucial to evaluate the ease-of-use of each approach, ensuring that domain experts can easily understand and use the models without extensive training. While specialised tools and user interfaces can enhance usability by hiding the complexities of direct modelling, this study focuses specifically on the intrinsic ease-of-use of the modelling approaches themselves. We argue that understanding how data is modelled and ensuring its accuracy is crucial for CH domain experts, even if they are not directly responsible for data modelling and querying. Their feedback is essential for validating the effectiveness and correctness of the ontologies and KGs used in the CH domain. Consequently, evaluating ease-of-use helps ensure that the models not only meet technical requirements but also reflect the practical needs and expectations of end-users.

The seven identified requirements highlight the distinct temporal and technical demands of CH projects. Together, they provide a foundation for evaluating existing temporal data models and inform decision support for the CH community in selecting appropriate approaches for temporal data representation. While these requirements are grounded in domain-driven design and real-world constraints derived from iterative collaboration with CH experts, they also align with broader ontology evaluation principles such as clarity, usability, extensibility, and scalability, as outlined in established frameworks [23]. These principles are reflected, for instance, in the attention to ease-of-use, RDF compatibility, and the support for temporal semantics. However, the focus of this work lies in the pragmatic application of temporal modelling within CH projects, where practical solutions often outweigh formal ontology design ideals. Thus, the presented requirements are not intended to exhaustively reflect theoretical completeness but rather to guide realistic decision-making based on specific needs of a CH project. The following section reviews existing work in relation to these requirements.

3 Related Work

To address the challenges and requirements of temporal representation in CHKGs, this section surveys a range of relevant modelling approaches and comparative studies. It begins by reviewing how temporality is currently modelled in existing CHKGs, followed by an overview of how foundational ontologies conceptualise time. The section then explores established temporal data modelling frameworks, with a particular focus on RDF-based extensions. In addition, to lay the groundwork for answering RQ4 (*"How can temporal modelling approaches be evaluated based on CH-specific requirements and technical criteria?"*) Section 3.4 examines comparative studies and surveys.

3.1 Temporal Description Logics

Temporal Description Logics (TDLs) extend classical Description Logics (DLs) by incorporating temporal constructs to represent and reason over knowledge that changes over time. While DLs are widely used for modelling structured knowledge with formal semantics and decidable reasoning, they typically assume that all statements are eternally valid. However, this assumption does not hold for many domains, including CH, where facts are often temporally situated.

Initial work in TDLs, such as combination of DL $\mathcal{FLENR}^-$ with the modal logic $\mathcal{HS}$ [58, 31], laid the foundation for representing temporal relations such as *before*, *meets*, or *overlaps* [2]. Although expressive, such early combinations proved undecidable, prompting later efforts to design fragments that balance expressiveness with decidability and practical reasoning. Surveys such as [6, 7] categorise these into three primary types: point-based, interval-based, and metric TDLs.

Point-based TDLs define time as a sequence of discrete instants, using temporal operators from temporal logics like LTL [52] and CTL [20]. Interval-based TDLs model time through bounded intervals and often leverage decidable subsets of $\mathcal{HS}$. Recent developments also introduce metric TDLs [30, 8], which allow for quantitative temporal constraints (e.g., events occurring within a specific duration). Some TDLs, such as those used in ontology-based systems like *BALOoNSE* [55], demonstrate promising applications in areas like patient care and industry.

Despite their high expressiveness and robust reasoning capabilities, TDLs introduce additional modelling and computational complexity. Reasoning typically requires specialised inference engines beyond standard OWL reasoners and may involve logic programming, automata-theoretic methods, or model checking. Moreover, many fragments are not supported natively in common triple stores or SPARQL-based workflows, requiring alternative tools or custom implementations.

Within the scope of this research, TDLs are not evaluated in depth for two reasons. First, while CH applications certainly benefit from temporal reasoning, most CH infrastructures are based on RDF(S) and OWL and rely on OWL inference provided by common tools. In practice, domain partners in CH projects often lack access to specialised reasoning environments and require solutions that integrate easily with SPARQL endpoints and Linked Data publication pipelines. Second, the requirements analysed in this study originate from concrete use cases (Section 2.2), which prioritise intuitive modelling (REQ7), standard query ability (REQ5), manageable complexity (REQ2), etc. TDL-based systems of today struggle to meet this without significant adaptation. While TDLs provide important theoretical foundations for formalising temporal reasoning, they are not included in the main comparative evaluation of this work due to their limited adoption and practical integration in current CH infrastructures. Nonetheless, they offer valuable insights that may inform future efforts to enhance temporal expressivity and reasoning in RDF-systems.

3.2 Event-based Models for Temporal Representation

One of the most widely adopted paradigms for modelling temporality in CH data is the use of event-based representations. The event-based approach treats events as the main entities that serve to contextualise changes in time. Thus, people, places, objects, and time are linked through event entities, which enables capturing complex temporal relationships via direct attachment of temporal information to the event. This has become a foundational strategy in many upper ontologies as well as CH ontologies and KGs. The following subsections explore how this approach is grounded in foundational ontologies and how it is practically applied in existing CH knowledge graphs.

Time in Foundational Ontologies

Foundational ontologies typically approach temporality through core ontological distinctions that categorise entities based on their persistence and change over time. The *Basic Formal Ontology* (BFO) [5, 51] introduces a fundamental separation between *continuants*, which exist wholly at any time they are present, and *occurrents*, which unfold over time and have temporal parts. Temporality is modelled through participation in occurrents that are explicitly linked to temporal intervals. Similarly, *DOLCE* [24] distinguishes between *endurants* and *perdurants*, where perdurants include events and processes that exist in time. Endurants are then “participate” in perdurants (events), providing a way to model temporality. *GFO* (General Formal Ontology) [35] represents time as a dimension parallel to space, with processes being temporal and possessing temporal parts, and *presentials* defined as entities that exist at specific time boundaries. Concrete individuals thus exist in time, but cannot be described as having temporal parts. The *Unified Foundational Ontology* (UFO) [27] incorporates temporality through three interlinked sub-ontologies: UFO-A for endurants, UFO-B for perdurants (events and situations), and UFO-C for social entities. Temporality is handled most explicitly in UFO-B, where temporality is, similarly to other ontologies, is modelled through structured event participation. UFO-B, however, is completely contextualised in First-Order Logic. Lighter-weight ontologies such as *PROTON* [64] distinguish between *happenings*, *objects*, and *abstractions*, introducing categories for events and situations. More comprehensive upper-level ontologies like *SUMO* [47] and *Cyc* [26] provide extensive temporal vocabularies and formalisations, including support for time intervals, recurring events, and duration constraints. However, they are often defined in higher-order logic or proprietary languages, making integration into RDF-based CH applications more challenging.

Overall, while foundational ontologies provide theoretical tools for conceptualising temporality through entity classification (so-called “objects” vs events) and temporal relations, they lack direct support for the kind of temporal annotation and querying required in RDF-based knowledge graphs. As such, their use in CH often requires additional layers of abstraction.

Time in Existing CH KGs

In context of CH, CIDOC-CRM [16], for instance, is a well-established event-based ontology used to model complex historical and cultural events. Via introducing events and activities, it captures complex temporal relationships between entities, actions, and time periods, and is widely adopted in domains like archaeology and museum documentation. Unlike direct RDF relationships in entity-centric representations, which might struggle to incorporate temporal annotations, event-based models provide a framework where temporal aspects are managed through intermediary events. For instance, in the scenario, `<:play :director :WalterFriedrichPeters>`, an extension to RDF is required to annotate the fact with specific temporal information, such as the year 1938. In contrast, event-based representations allow for representing this information within an event:

```
:play :subjectOf :DirectingEvent .
:DirectingEvent :performedBy :WalterFriedrichPeters .
:DirectingEvent :validIn "1938" .
```

A number of CHKGs build on CIDOC-CRM or similar event-centric paradigms to address temporal modelling needs. For example, the ArCo Knowledge Graph [18], developed for Italian CH data, leverages CIDOC-CRM and introduces specialised modules to support temporal modelling, such as dates of artifact creation, transformations, and changes in ownership. ArCo integrates multiple temporal ontologies and supports both time intervals and approximate dates (as literals), offering a rich but complex modelling strategy that inherits the layered nature of event-based design. Several domain-narrower data models extend CIDOC-CRM to represent the knowledge. For example, the TRANSRAZ data model [14], designed to support historical research on urban transformation in Nuremberg, builds directly on CIDOC-CRM and emphasises the temporal evolution of entities such as buildings and street networks. The Sampo model family [38], adopted across several Finnish CH projects (e.g., BiographySampo, WarSampo), follows a similar pattern by using event-based structures to represent life events, affiliations, and temporal relationships. Alternative to CIDOC-CRM is the Europeana Data Model (EDM) [25] that supports only a lightweight form of temporal annotation using properties such as `dcterms:created` and `edm:hasMet`, and represents events as contextual nodes. EDM provides less semantic depth in comparison to CIDOC-CRM, but prioritises interoperability and integration across a wide variety of heritage institutions. Thus, temporal modelling is often simplified, reflecting the need to harmonise data from heterogeneous sources.

While event-based models are effective and widely used for representing temporality in CH, they introduce additional modelling complexity by requiring events as intermediary entities. This added abstraction can complicate both data creation and querying, especially when only simple temporal annotations are needed. In many CH applications, temporal data primarily concerns approximate periods or dates associated directly with entities, such as persons, roles, or facts, without necessitating the introduction of separate event constructs. Moreover, CH datasets often originate from raw, unstructured sources, e.g. archival documents or spreadsheets, with no preexisting URIs or ontology alignment. In such contexts, restructuring the data to fit an event-based paradigm becomes especially cumbersome. It requires rethinking what counts as a temporal entity (e.g., transforming a fact into an event) and introducing additional layers of modelling just to attach time information. This not only increases the modelling effort, but also risks obscuring the original data structure in ways that are confusing or inaccessible to domain experts.

Given these challenges, this study explores whether alternative approaches, RDF-based extensions, can offer more intuitive ways to represent temporality. By enabling time to be attached directly to facts or relationships without the need for intermediate events, RDF extensions may better support the pragmatic needs of CH practitioners. Consequently, the remainder of this work focuses on RDF extensions as a potentially more lightweight and accessible solution for temporal modelling in CH knowledge graphs.

3.3 Temporal Extensions to RDF

As an alternative to event-based models, RDF reification has been proposed as a way to annotate RDF triples with contextual information[9]. Using the RDF vocabulary (`rdf:Statement`, `rdf:subject`, etc.), a triple can be reified and metadata, e.g. start and end dates, attached. For

---

[9] `https://www.w3.org/TR/rdf-schema/#ch_reificationvocab`

example, to express that theatre performance of *Minna von Barnhelm* was directed by *Hans Tessmer* and by *Wilhelm Speidel* in different time periods, the corresponding triples are reified and annotated with their temporal context, for this:

- two blank nodes of type `rdf:Statement` are introduced.
- the blank nodes are then linked to their subjects (*performances of Minna von Barnhelm*), predicate, e.g. `:directedBy`, and objects (*Hans Tessmer* and *Wilhelm Speidel*) via corresponding properties `rdf:subject, rdf:predicate, rdf:object`.
- the blank nodes are linked to their corresponding time validity (e.g. *1933* or *1939*)

While reification enables annotation of triples with metadata, it introduces significant verbosity and structural overhead, since for every triple `<s, p, o>`, blank nodes and additional triples are introduced (`<_:x, rdf:subject, s>, <_:x, rdf:predicate, p>, <_:x, rdf:object, o>`). Furthermore, RDF reification is a general-purpose mechanism and does not provide built-in semantics for interpreting the metadata it attaches – including temporality.

A variety of approaches have been proposed that extend RDF in different ways to support temporal representation more explicitly and efficiently. The goal of this section is to provide a high-level overview of the most prominent RDF-based temporal modelling approaches. A more detailed analysis of each approach and how it performs with respect to CH requirements is presented in Section 4. The RDF extensions are grouped according to the level at which they introduce temporality – component, triple, or graph level – to illustrate the different strategies available for temporal modelling in RDF.

Component Level Extensions

In component level approaches, temporality is introduced by modifying individual components of RDF triples, such as properties or entities.

- **Singleton Property** is proposed in [45]. The authors argue that in every temporal context, the relationship between two entities is unique. They introduce the concept of "generic property" to represent the general characteristics of relationships and "singleton property" to capture the specific relationship within a particular time context. Thus, a unique property is modelled as an instantiation of a generic property via `rdf:singletonPropertyOf`. The temporal fact has a form of triple `<s, p@t, o>`, where `p` is a generic property and `p@t` is a singleton property that defines a unique temporal context of `p` at time `t`.
- **N-ary Relations** approach introduced in [48] addresses the problem of binary representation of relations in RDF/OWL. The inability to relate more than two individuals or to describe a relation itself is proposed to be solved via objectifying a relation. An objectified relation is a relation converted into an entity that may serve as domain or range of a property. Thus, a temporal fact has a form of a triple `<s, p_1, o>`, where `o` is an objectified relation that can be further described with its time validity `t` – `<o, p_2, t>`.
- **4D Fluents** [75] are based on the perdurantist view, which argues that on a certain scale every entity appears to be different in a different period of time. According to this perspective, each entity is made up of temporal parts that reflect its state at various time points. To convert regular entities into 4D Fluents, class `fl:TemporalPart` and properties `fl:temporalPartOf` and `fl:temporal` are introduced. Consequently, a temporal fact is represented via triple `<s@t, p, o@t>`, where `s@t` and `o@t` are temporal parts of `s` and `o` that are valid at time `t`. In [11] the 4D Fluents ontology is extended via introducing the new class `fl:TimeInterval` with property `fl:isTimeInterval`. Additionally, this extension incorporates Allen's 13 interval relations, such as *before*, *met by*, and *equals*, thereby providing additional semantic depth to the model.

- **Valid Time RDF** (*VTRDF*) [72] extends the ideas of 4D fluents, defining valid time resources and relations. Thus, every fact in VTRDF is temporal and represented by a VTRDF triple <s^t, p^t, o^t>, where every entity `s` and `o` is entailed with its existence in time `t`, while `p` establishes a valid time relationship between `s` and `o`. To achieve such representation, the standard RDF vocabulary as well as the standard IRI are extended. A valid time IRI consists of a resource and its valid time that is identified by a delimiter •, e.g. `http://example.org/Henry•[01-01-1990-now]`. Furthermore, literals, data types, blank nodes of standard RDF are all temporalised to their valid time variants. Additionally, the Valid Time SPARQL language for querying VTRDF graphs is designed.

Triple Level Extensions

Triple level extensions incorporate temporality directly into RDF triples, either by modifying them or by attaching additional temporal labels.

- **Temporal RDF** (*tRDF*) [28] and its variants [29, 37, 54] are the extension of standard RDF graphs that allow to label triples with time by means of reification. The standard RDF vocabulary, `rdf:Statement, rdf:subject, rdf:object` and `rdf:predicate`, is used for reification purposes, while temporal semantics is incorporated by introducing temporal vocabulary: properties `trdf:temporal, trdf:initial, trdf:final, trdf:instant, trdf:interval` and literal `NOW`. Thus, a temporal RDF triple has form `<s, p, o>[t]`, where `t` represents the time interval or the time point the triple `<s, p, o>` is valid in. In [29] the problem of uncertain temporality is addressed by allowing anonymous timestamps to label temporal triples. Thus, the temporal label `t` may state the temporality of a fact implicitly without mentioning the exact time point/interval. Additionally, temporal constraints are included, and reasoning over temporal graphs with intervals is entailed and practically tested [37]. Such enhancement of the model contributes to defining temporal relations between triples even if the exact time boundaries of their validity are uncertain or unknown.
- **Annotated RDF** (*aRDF*) [69] is a generic extension to the RDF model that is proposed to support various annotations of RDF triples, such as their provenance or temporal validity. Annotations of an aRDF triple are defined as algebraic structures of domain values, e.g. for temporal domain – time intervals or time points, and employs operators *meet* and *join* from annotated logic [39] to combine the values. The aRDF formalism is represented as `<s, p, o>[a]`, where `a` represents an annotation from a partially ordered set of annotations `A`. aRDF employs annotated logic, which differs from the standard RDF reasoning mechanisms. This divergence requires specialised frameworks for data integration and querying, as the underlying logic for annotations may not directly align with traditional RDF processes.
- **RDF*** [32] extends standard RDF by introducing nesting triples. More specifically, an RDF* fact has a form of a triple `<<s>, p, <o>>`, where `s` and `o` may be directly represented by other triples. The depth of nesting is denoted by `k`, where if `k=0`, the triple is a standard RDF triple; and if `k>0`, the triple is a nested RDF* triple. Turtle* and SPARQL* are the extensions of Turtle and SPARQL that enable writing and querying RDF* graphs. RDF* triples remain backwards compatible and may be transformed into standard RDF triples via unfold, black node assignment and reification functions.

Graph Level Extensions

Graph level extensions introduce temporality by grouping RDF triples into separate graphs based on their temporal dimensions.

- **Named Graphs** are used to group all RDF triples that share the same meta knowledge into separate graphs. This is achieved by adding additional dimension(s) to RDF triples. In [17], the quad format of Named Graph model – `<s, p, o, NamedGraph>` is adopted by W3C. In temporal context, such representation is useful for modelling and querying data from multiple time intervals or time stamps, where events and facts for each time interval can be managed as separate graphs associated with time `t`: `<NamedGraph, p_2, t>`. In contrast to reification, Named Graphs allow for modelling and querying without transformation of original triples.
- **RDF+** [59], the extension of RDF for querying metadata knowledge, proposes a quintuple format by extending the named graph quad with a statement identifier – `<s, p, o, NamedGraph, statementID>`. The identifier is used as the subject of the meta-knowledge assertion, which is an RDF triple.

Most of the mentioned extensions to RDF were not initially designed specifically for temporal contexts but aimed to incorporate meta-knowledge into RDF, thus lacking temporal vocabulary. However, there are several external time ontologies that offer comprehensive temporal semantics and can be used to address various needs for expressive temporal representation [21, 10, 4, 50, 56, 12, 19]. These ontologies, discussed and evaluated in Section 4.1, provide specialised vocabularies and frameworks to incorporate temporal information into RDF.

In summary, various approaches have been developed to extend RDF with temporal capabilities, each offering a unique perspective on integrating temporality into RDF data. However, given this diversity, it becomes increasingly important to understand how these approaches compare in terms of their practical applicability, expressivity, and usability, particularly in the context of specific domain needs such as those in CH. This leads to a crucial research question: **RQ4: How can temporal modelling approaches be systematically evaluated based on CH-specific requirements and technical criteria?**. The following section reviews the work related to this RQ, highlighting existing comparative efforts, their focus areas, and methodological approaches.

3.4 Surveys on Temporal Extensions to RDF

To the best of our knowledge, the first comprehensive survey on the representation of temporal data in RDF is conducted by Rula et al. [57]. This study categorises various approaches to temporal metadata description into three main types: document-centric, sentence-centric, and relationship-centric. It focuses on evaluating the adoption of these approaches within a substantial dataset of RDF quads – quadruples of the form `<s, p, o, c>`, where `c` specifies the contextual information of an RDF triple `<s, p, o>`. The survey employs quantitative methods, including random sampling and manual verification, to assess the prevalence and accuracy of temporal metadata representations. The evaluation is primarily data-centric, concentrating on how frequently and effectively different approaches are implemented.

In [73, 72], a comprehensive analysis is conducted, organising temporal models into explicit and implicit reification-based approaches. For this, a detailed taxonomy based on technical criteria such as RDF, RDFS, or OWL extension availability, requirement of additional objects, and support for instant or interval time values is developed. This evaluation framework provides a nuanced view of how different models handle temporal data.

Later, Zhang et al. [78] address the need for a more detailed review of storage techniques and categorise temporal approaches based on syntax. Their work focuses on models that extend RDF triples to new forms and those that add timestamp information to standard triples. They emphasise the importance of storage efficiency and practical implementation details. A brief discussion of the contextual annotation in RDF is provided in [36]. Their taxonomy includes direct representation, reification approaches, higher-arity models, and annotation models that define context.

Another study discussed in [60] focuses on the integration of meta-knowledge in the Semantic Web, particularly with regard to time, trust, fuzzy logic, and provenance dimensions. It provides a systematic review and experimental evaluation of various modelling approaches, including their performance in terms of graph size, number of statements, redundancy, storage, query length, and response time.

The work by Polleres et al. [53] provides a conceptual framework for analysing the evolution of open knowledge graphs, identifying dimensions such as terminological drift, structural transformation, and data-level changes. This research offers a typology of KG evolution that is valuable for understanding dynamics. Our work complements this research by grounding the evaluation in specific temporal modelling approaches at the RDF level, focusing on how such models can be applied to represent temporality in CH data. The emphasis is placed on evaluating concrete modelling approaches in terms of their applicability to CH case study, rather than categorising types of change abstractly. In this way, the two lines of research address complementary aspects of temporality in KGs: one through high-level evolution frameworks, the other through practical modelling and evaluation of RDF extensions.

To summarise, the current work provides the following contributions that differentiate it from the related works:

- **Cultural Heritage Focus**: Unlike previous works that focus on generic temporal RDF models, this research specifically addresses the representation of temporal data in CH context, which introduces unique requirements, such as evolving historical facts and concepts, timelines, and complex relationships that change over time.
- **Real-World Use Cases**: The research is application-based and integrates insights from four real-world CH use cases, which inform the identification of critical temporal requirements. This practical perspective is missing from other works, which tend to focus on theoretical models or broad taxonomies without engaging with real-world data needs.
- **Modelling Requirements obtained from CH Use Cases**: To evaluate the models, a set of evaluation requirement is developed to address the specific needs of the CH domain.
- **Comprehensive RDF Extension Analysis**: The work goes beyond merely focusing on technical metrics like storage size or binary criteria such as whether OWL reasoning is supported or not when evaluating the extensions. Instead, an in-depth analysis and discussion of RDF extensions is provided, thoroughly evaluating their capability to handle temporal data in CH settings. This includes exploring the potential consequences of adopting each extension, assessing their suitability for different use cases, and evaluating the extent to which these models address the complexities.
- **Decision-support Tool for Practitioners**: The work proposes a practical, scenario-based decision-support system to help practitioners prioritise and select the most suitable RDF extensions based on specific project needs. Other works offer taxonomies and classifications but lack this actionable, decision-making framework, making it harder for users to apply the findings in real-world (CH) projects.

4 Comparative Evaluation and Discussion

In this section, a qualitative evaluation of six widely used temporal modelling approaches is conducted: N-ary Relations, Named Graphs, Singleton Properties, tRDF, RDF*, and 4D Fluents. These approaches are chosen because they have seen the most practical use in real-world applications. The aim is to provide empirical insights and discussions that deepen the understanding of how each approach meets the requirements, the implications of selecting a particular approach, and the inherent strengths and weaknesses associated with each. This analysis lays a comprehensive foundation for developing a decision-support tool (Section 5), offering context for interpreting its design and outcomes.

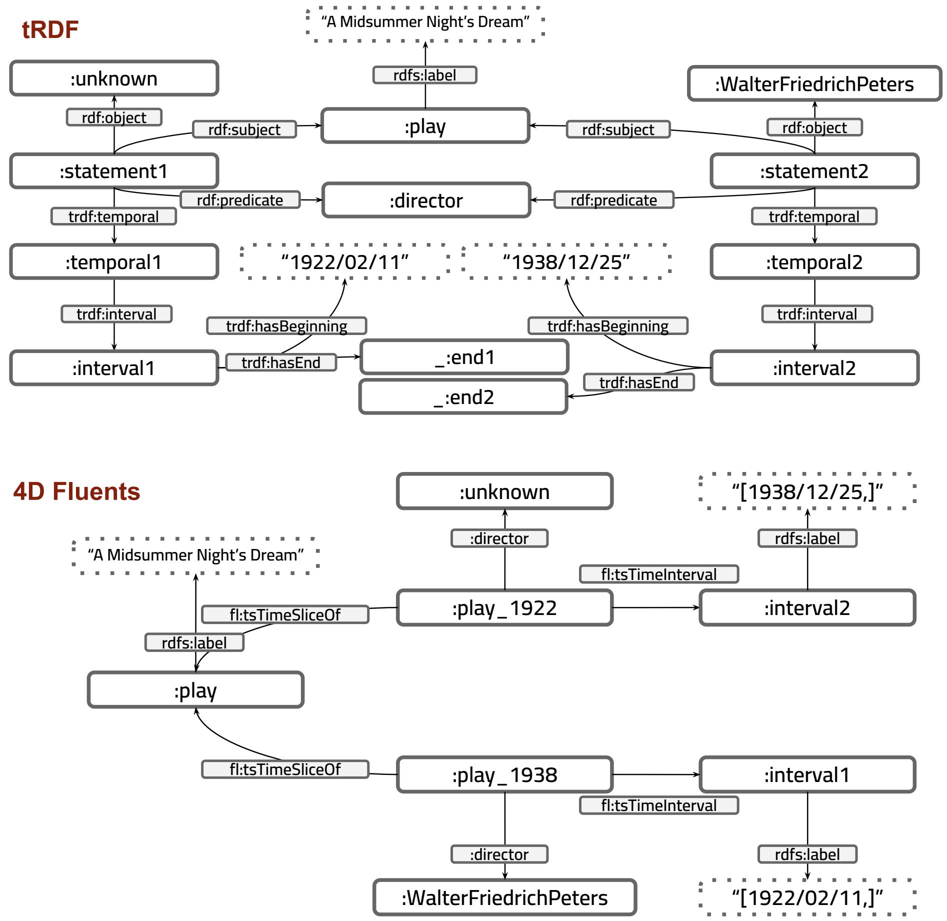


Figure 1 Modelling semantic drift with tRDF and 4D Fluents.

4.1 REQ1: Temporal Expressivity

Temporal expressivity is a core modelling requirement in CH contexts. As discussed in Section 2.2, this requirement is structured into four sub-dimensions – contextual change, temporal statements, temporal uncertainty and incompleteness), and interval relations.

Contextual Change (REQ1.1) is foundational to temporal modelling in CH and is implicitly addressed by all evaluated approaches. It refers to the need to capture how entities and their descriptions evolve over time, e.g. changes in roles, names, functions, or classifications, while maintaining a stable identity. Each model handles this requirement through different structural mechanisms: some reify statements and annotate them with temporal information (e.g., tRDF, RDF*), some transform properties into temporal relations (e.g., N-ary Relations), and some explicitly model temporal slices of entities (e.g., 4D Fluents). These structural strategies enable the representation of contextual change, allowing multiple descriptions of the same entity at different points in time (see Section 3.3 for technical modelling details).

REQ1.1 reflects the need to represent contextual change – that is, the ability to track how the description of an individual entity evolves across time without changing its identity. These temporally-scoped descriptions are crucial in CH, where narratives are rarely static and must

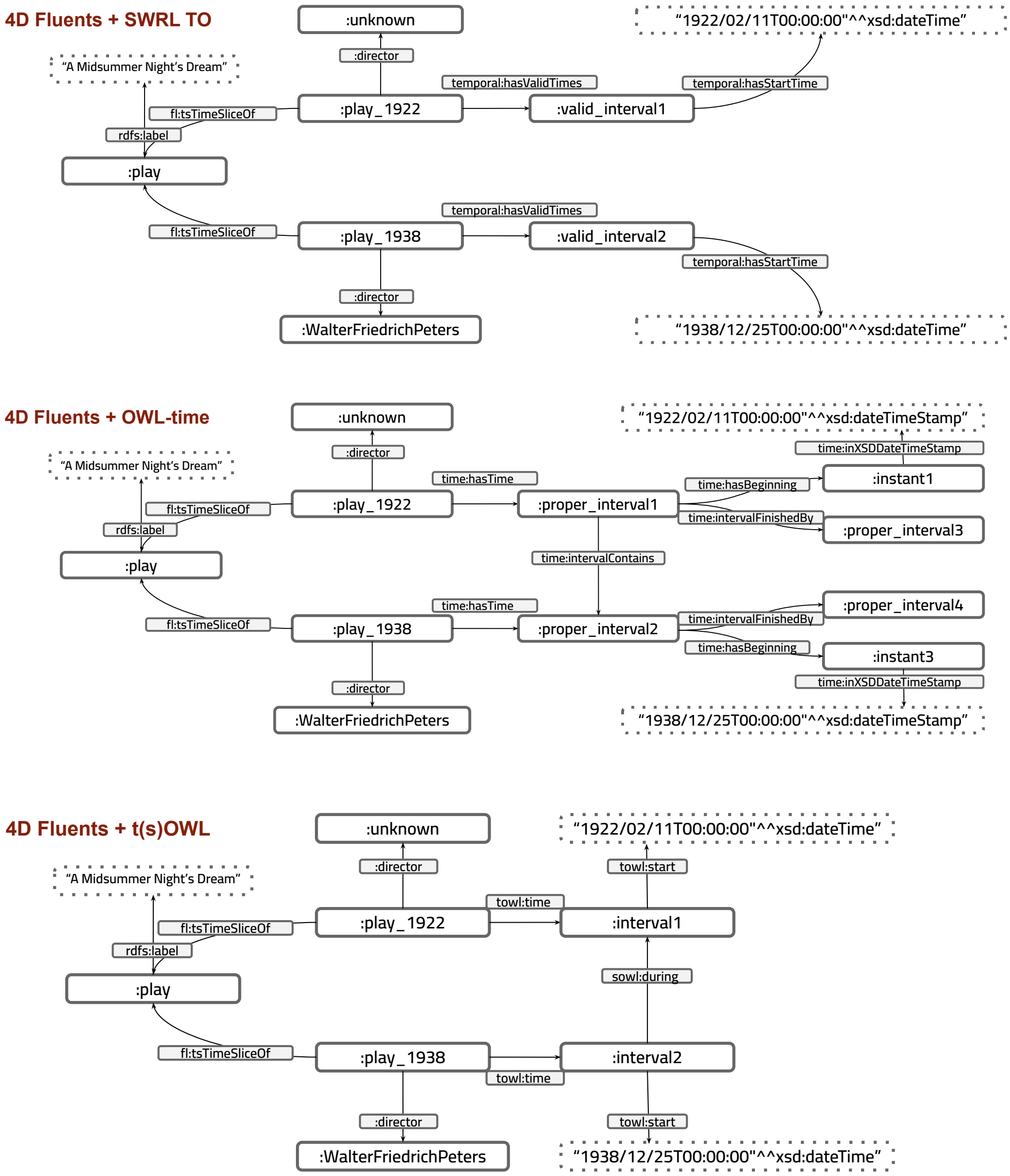


Figure 2 Modelling semantic drift with temporal ontologies – SWRL Temporal Ontology, OWL-Time, and tOWL.

accommodate contested or evolving facts. While REQ1.1 forms the conceptual foundation, the approaches diverge significantly in how they support more specific temporal capabilities, e.g. representing temporal scopes, modelling uncertainty in temporal data, or supporting relations between time intervals. The following discussion addresses each of these dimensions in turn, examining how different models fulfill these CH requirements. To support the discussion, the modelling of a contextual change scenario is visualised:

> *A play titled "A Midsummer Night's Dream" was performed starting from 1922/02/11. On 1938/12/25, a new director was hired, leading to a conceptual change in the storyline of the play, resulting in significant differences during a certain period.*

Figure 1 depicts this scenario using tRDF and 4D Fluents, while Figure 2 shows representations using external temporal ontologies.

REQ1.2: Temporal Statements. Time instants and intervals are crucial temporal concepts in RDF-based temporal modelling. In tRDF, dedicated vocabulary like `trdf:Instant` and `trdf:Interval` is used, and intervals are defined by linking them to their start and end time points via properties like `trdf:hasBeginning` and `trdf:hasEnd`. The entailment of the temporality to a statement succeeds via property `trdf:temporal` which time borders are then further specified via `trdf:instant` or `trdf:interval`. In the 4D fluents model, temporal vocabulary involves introducing two classes: `fl:TimeSlice` to represent slices of entities in a specific time, and `fl:TimeInterval` to represent periods of time. Additionally, `fl:tsTimeSliceOf` property links a time slice to its generic entity, and the `fl:tsTimeInterval` property connects a time slice to a time interval, thus associating temporal details with each time slice.

The SWRL Temporal Ontology [50] is a lightweight approach that can be layered on existing OWL languages to add a temporal dimension to a standard model. The core class of the ontology is `temporal:Fact`, it models a temporal individual that is connected to time periods when it is held to be true via the property `temporal:hasValidTimes`. The range of the property is the class `temporal:ValidTime`, which has sub-classes `temporal:ValidInstant` and `temporal:ValidInterval`. The borders of the interval are modelled via connecting intervals to the time points of their beginnings and ends via properties `temporal:hasStartTime` and `temporal:hasFinishTime`. The instants are linked to temporal data using properties like `temporal:hasTime`, which connects an instant to a precise time value (e.g., a date or timestamp). Additionally, to address the granularities of intervals and points, e.g. days and minutes, class `temporal:Granularity` is introduced.

tOWL (Temporal OWL) [21] is an extension of OWL designed to incorporate temporal reasoning within ontologies. tOWL offers a specialised framework for temporal modelling with classes like `towl:Instant` and `towl:Interval` for representing time instants and intervals. Instants of tOWL are represented using XML Schema `dateTime`, capturing a specific point in time. Intervals are modelled with properties `towl:start` and `towl:end`, representing the beginning and end of a time span. These properties rely on the concrete domain based on rational numbers to ensure the start is always strictly before the end, enforcing proper interval definitions. Similarly to 4D Fluents, `towl:TimeSlice` describes temporal aspects of entities, associating them with a period using `towl:time`.

OWL-Time provides[10] the most expressive and granular ontology for time instants and intervals. It starts with the class `time:TemporalEntity` and its two primary sub-classes: `time:Interval` and `time:Instant`. The vocabulary includes properties `time:hasBeginning` and `time:hasEnd`

---

[10] `https://www.w3.org/TR/owl-time/`

to describe temporal limits of an interval and to link it to the entity of type `time:Instant`. Additionally, intervals can be linked to `time:Duration` via `time:hasTemporalDuration` to represent an extent of an interval, expressed as a scaled value or a nominal value. Instants can be specified with properties like `time:inXSDDateTimeStamp, time:inXSDgYear, time:inTimePosition, time:inDateTime` to provide various ways of indicating their temporal position.

REQ1.3: Temporal Uncertainty and Incompleteness. In [41], the SWRL Fuzzy Temporal Ontology (SWRL-FTO), extension of the SWRL temporal ontology is introduced to address modelling and reasoning with imprecise temporal expressions. Thus, the `fuzzytemporal:FuzzyTime` class represents imprecise or uncertain time values associated with fuzzy temporal propositions. It has two main sub-classes: `fuzzytemporal:FuzzyTimeInstant` and `FuzzyTimePeriod` that represent an event spanning an imprecise time instant and interval with uncertain start and end times accordingly. The `fuzzytemporal:FuzzyModifiers` class includes terms like “about” or “around” that describe the fuzziness of temporal information, helping to quantify this uncertainty. Although the approach introduces a promising concept for handling imprecise temporal expressions through the SWRL-FTO ontology, further evaluation and empirical evidence, such as a concrete OWL implementation or experimental results, are needed to fully demonstrate its practical effectiveness.

A recent contribution to temporal modelling is the Onto-mQoL framework [61], which offers a solution for representing both precise and uncertain temporal information. Onto-mQoL builds on the n-ary relation approach. To capture both precision and uncertainty, the framework introduces specialised temporal classes. For precise temporal values, it defines `PreciseMoment` and `PreciseInterval`. In contrast, for modelling vagueness and uncertainty in temporal expressions, it defines classes `UncertainMoment` and `UncertainInterval`. An uncertain moment includes a reference point, a degree of imprecision (`hasUncertainRange`, typed as a duration), and a functional modifier (`hasUncertainFunction`, e.g., “about”, “before”, “early”). Similarly, an uncertain interval allows both a precise moment and an uncertain moment at its temporal boundaries. Custom SWRL rules are defined to infer temporal relationships such as `almostSurelyBefore`, accounting for the fuzziness inherent in uncertain intervals. To support such expressive logic, Onto-mQoL relies on SQWRL [49]. Overall, Onto-mQoL presents a robust and OWL DL–compliant approach for modelling both precise and uncertain temporal aspects. However, due to its high expressivity and reliance on SWRL-based reasoning, formulating queries becomes complex and less accessible, particularly in comparison to standard SPARQL workflows. Future work may investigate its practical applicability in CH contexts, including the learning curve for non-technical users and opportunities for optimising rule management and querying efficiency.

Alternatively, temporal uncertainty can be represented in KGs either textually through plain literals, offering a straightforward but less formalised solution, or through translating them into qualitative intervals, which provide a more structured approach but still lack precise definitions. However, for more advanced representations (and reasoning) about uncertain dates, normalisation techniques are necessary. While this work does not cover these techniques, tools like HeidelTime [63, 62] specialise in detecting and normalising temporal expressions in text. HeidelTime employs pattern matching to identify various temporal references and can be extended to enhance its coverage and accuracy, thus offering a more robust solution for managing temporal uncertainty.

REQ1.4: Interval Relations. For many use cases, it is crucial not only to represent the temporal boundaries of facts, but also to express how time intervals relate to one another, e.g. whether one interval precedes, follows, overlaps another. Such temporal relations become even more important in the presence of temporal uncertainty or incompleteness (see REQ1.3). In [37] a formalism known as c-temporal graphs, which extends traditional temporal RDF by integrating constraints and interval algebra is introduced to model relations between intervals with imprecise or unknown boundaries. sOWL [10] extends tOWL by supporting qualitative temporal relations between intervals, such as `sowl:during`. This property allows for the representation of how one interval can occur during another, without needing to define their concrete start and end points. SWRL Temporal Ontology also integrates functions based on Allen's interval algebra, providing operators like `temporal:before, temporal:overlaps, temporal:contains`. These functions enable reasoning about the relationships between intervals based on qualitative aspects, regardless of whether the exact start and end points are specified. OWL-Time is primarily designed for representing and reasoning about temporal entities with explicitly defined start and end points, and provides mechanisms for detailing temporal ordering and relationships with 15 interval relations [3] such as `time:intervalBefore, time:intervalAfter, and time:intervalMeets`. The authors acknowledge that temporally incomplete intervals and their relations can be modelled within OWL-Time, but restrict themselves from addressing reasoning about such intervals, stating that additional methods or extensions may be necessary beyond what is provided by OWL-Time.

Overall, the approaches reviewed in this section reflect a wide spectrum of solutions for achieving temporal expressivity in CHKGs. All models aim to support contextual change, but they differ in their structural strategies and in their expressive power and complexity. Native support for time instants and intervals is uneven: it is built-in in tRDF, 4D Fluents, tOWL, and OWL-Time, while RDF* and Named Graphs leave the temporal domain to be defined externally. Only a subset of models, e.g. SWRL Temporal Ontology, SWRL-FTO, and Onto-mQoL, provide mechanisms to address temporal uncertainty and incompleteness, though with varying degrees of maturity and practical implementability. Similarly, modelling qualitative interval relations is not natively supported by most models and typically requires external ontologies (e.g., sOWL, OWL-Time) or rule-based extensions based on Allen's interval algebra. More expressive frameworks such as Onto-mQoL introduce robust capabilities for handling vague or uncertain temporal expressions but rely on custom SWRL rules and SQWRL querying, which can be complex to implement and query in practice.

Although no single approach fully satisfies all four dimensions of REQ1, several offer strong partial support depending on the modelling priorities (e.g., clarity vs. expressiveness) and infrastructure requirements (e.g., tool compatibility, reasoning support) of the use case. This highlights the importance of individual use case-aware model selection.

4.2 REQ2: Storage and Scalability

To assess this requirement, the number of RDF triples required by each approach in a representative modelling scenario is compared. While not a standalone measure of scalability, triple count provides a practical and widely adopted proxy for estimating modelling overhead. To assess the number of triples utilised by each temporal modelling approach, an example from use case 1 (see Section 2) has been used to illustrate a terminological change of context over time. Specifically, the example models the renaming of a street:

Until 1809, the street was called "Roßmarkt," but after 1809, it was renamed to "Adlerstraße".

This example is depicted in Figure 3 and Figure 4, and serves as a basis for evaluating how different RDF temporal extensions handle this simple yet common scenario[11]. For the purpose of this evaluation, only the ABox is considered when counting triples, and the definition of classes and class memberships is excluded to simplify the analysis. The focus is placed on the number of triples required to model the temporal aspects of the change, thereby providing a clear comparison of how each approach handles temporal data in terms of scalability and efficiency.

---

[11] All examples are available at: `https://github.com/sashabruns/RDF-Extensions-for-Cultural-Heritage/blob/main/CR1_number_of_triples.md`

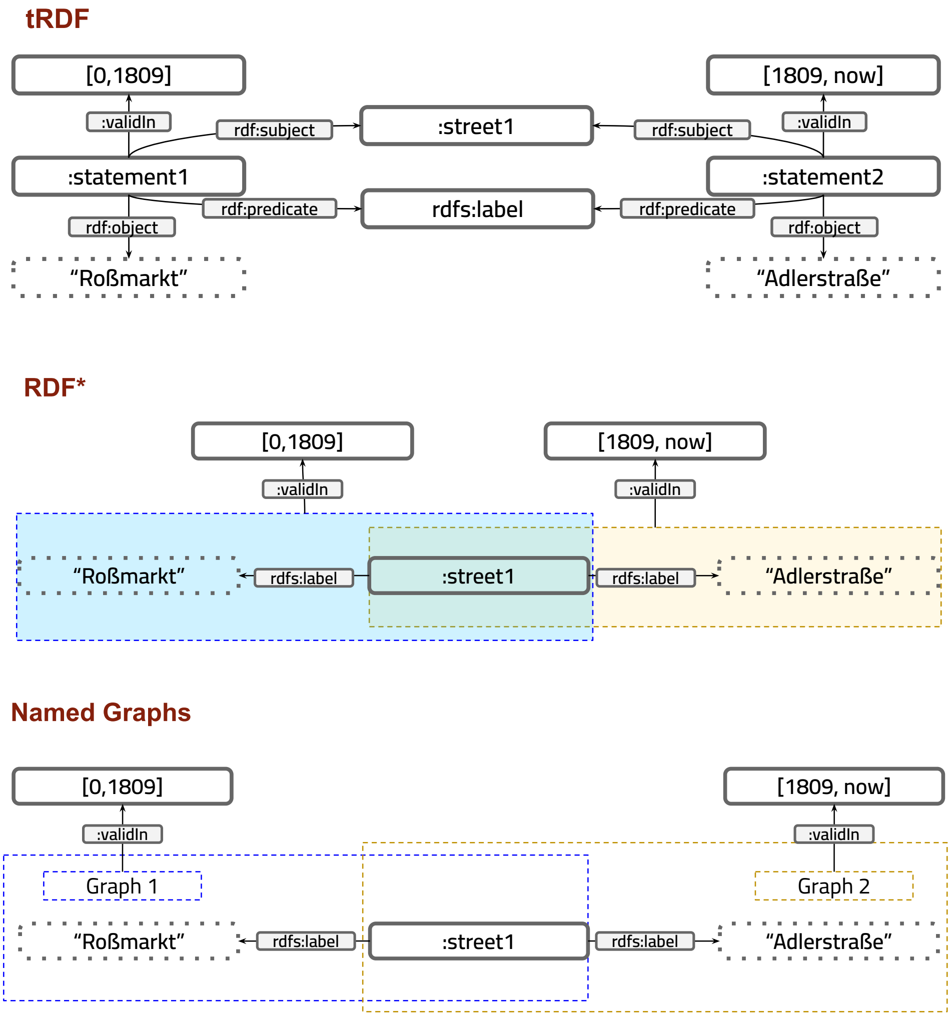


Figure 3 The running example of the triple and graph level extensions to RDF.

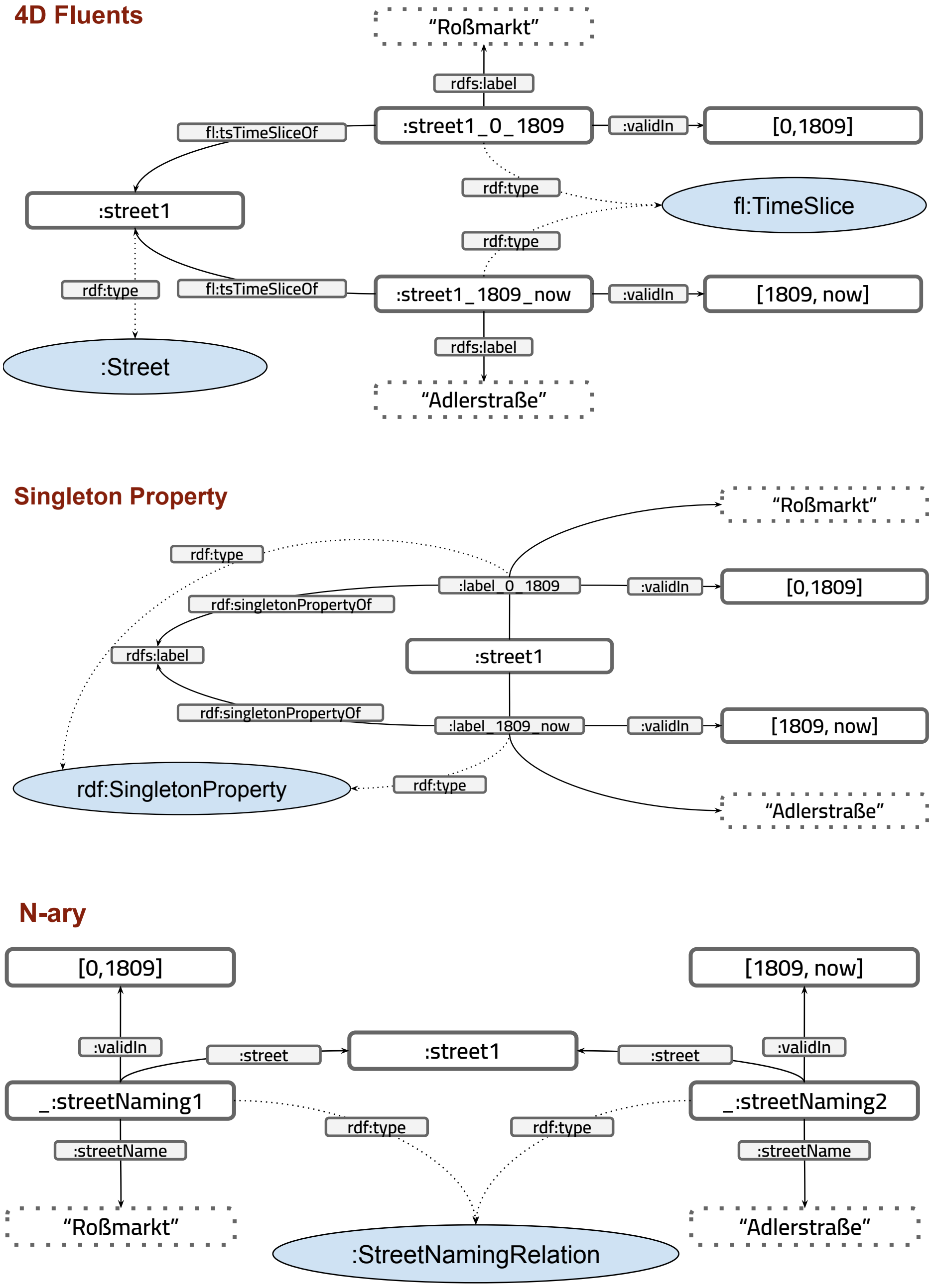


Figure 4 The running example of the component level extensions to RDF.

In the tRDF approach, each temporal statement is explicitly modelled using a combination of triples. Therefore, each statement about the street's label over time results in multiple triples, leading to a high count of triples – 8. RDF* uses reification to embed temporal validity directly within the triple structures. By integrating the temporal information with reified triples, and using RDF* syntax, RDF* achieves a more compact representation – 4 triples. Named Graphs represent temporal information by asserting RDF triples in different named graphs, where each named graph corresponds to a temporal snapshot. Triples that are valid within the same time interval are grouped under the same named graph, while additional RDF triples describe the temporal validity of these graphs. In the example, this results in two triples asserted in named graphs and two triples describing their temporal validity. The 4D Fluents approach involves creating separate entities for each time slice of the street, with associated labels and validity periods. Each time slice is modelled as an independent entity, resulting in multiple triples for both the temporal statement and the association with the original entity (6 triples). In the Singleton Properties approach, additional unique predicates are used to represent each temporal change (6 triples). The N-ary Relations approach models temporal changes by objectifying the relations between entities. In the example, separate entities to link the street with its names and validity periods are created that results in 6 triples.

Examining the representation of only one street provides a vision how each approach behaves but may not fully visualise the possible scalability challenges and efficiency differences between approaches when the amount of data increases. With a single street, the information is relatively unique and limited, leading to similar results across approaches. However, when additional information is introduced – such as name changes for four streets simultaneously – the differences between the approaches become more pronounced. Thus, in tRDF, each street and its associated changes require multiple triples, with each temporal statement needing its own set of triples. For the example involving four streets, it results in 32 triples in total. RDF* requires 16 triples (8 statements, each with associated temporal validity) to model four streets. Named Graphs group triples that are valid in the same time period under a single named graph, thereby consolidating the information. In this example, 8 triples are asserted in named graphs, and 2 additional triples describe the temporal validity of the two graphs. 4D Fluents, Singleton Properties and N-ary relations involve creating distinct entities for a temporal change. With four streets, the total number of triples is 24.

The analysis is visualised on the example where the object of the triples is a literal value (street names). To explore the impact of using entities instead of literals, further scenarios were considered, where the triples involve entities, such as when annotating temporal facts, e.g. `≪:person1 :livesIn :street1> validIn ’1908’>`, `≪:person1 :livesIn :street2> validIn ’1910’>`. The results for most approaches remained consistent with those observed for literals. However, in the case of 4D Fluents, using entities rather than literals results in an increased number of required triples. This is because both entities, in the subject and object positions, need to be associated with their respective time slices, further complicating the model and requiring additional triples. This extension underscores the added complexity and potential scalability issues that arise when modelling temporal changes with entities instead of literals. Singleton Properties would also see an increase in the number of triples needed if the object is an entity. Similar to 4D Fluents, each temporal change would require creating a new unique predicate, resulting in additional triples.

While counting number of RDF triples is a convenient and quantifiable proxy for modelling complexity, it should not be viewed in isolation or as a definitive measure of model performance. The evaluation in this section is based on small, illustrative examples, which do not capture the full variability of real-world CH datasets. However, triple count has been shown to correlate with broader performance measures such as data loading time and repository size. For instance, the Ontotext study[12] benchmarks various RDF modelling approaches on a subset of Wikidata, showing that RDF*, which requires fewer triples, significantly outperforms traditional reification and N-ary relations in terms of both repository size and load time.

4.3 REQ3: RDF Syntax Extension

Introducing temporal context to standard RDF often requires extending or adapting its syntax. While this is a technical consideration, it remains relevant to CH contexts where RDF-based systems are often pre-defined and tool usage may be constrained. Many CH infrastructures rely on fixed tools, and deviations from standard RDF syntax can limit compatibility and practical applicability. REQ3 therefore examines whether a temporal model introduces such syntax extensions and what concrete syntaxes (e.g., TTL, TriG, TTL*) it requires.

tRDF does not alter the fundamental RDF triple structure and can be implemented using standard RDF constructs. This makes it compatible with existing RDF tools and serialisations such as Turtle or RDF/XML. RDF* extends the RDF data model by allowing triples to be embedded within other triples, which simplifies metadata annotations, including temporal ones. Although it leads to a more streamlined representation, RDF* requires specialised syntax (TTL*) and querying tools (e.g., SPARQL*), which may not be available or permitted in institutional CH settings. While RDF* can be translated into standard reification, its full usability depends on support for these extended tools. In Named Graphs, each graph represents a distinct context within a dataset. For this, triples asserted in named graphs are commonly represented together with a graph identifier, a representation often referred to as a quad. This mechanism is well supported in most RDF frameworks and standardised in RDF and SPARQL. However, to fully utilise named graphs, dataset serialisations such as TriG are typically required, which introduce additional syntactic complexity compared to simpler graph-level formats such as Turtle.

Singleton Property extends RDF by creating unique predicates to express context, including time. Though it adheres to the RDF triple model, this approach increases the number of distinct properties significantly. Without specialised support to generate, maintain, and interpret these properties, the model can become error-prone and hard to manage, especially in long-lived CH projects. 4D Fluents introduce time slices as additional entities, linking them to base entities and time intervals. This modelling strategy is compatible with RDF but structurally complex, requiring careful construction of time-related entities and relations to ensure consistency. N-ary Relations transform simple binary assertions into more complex structures involving intermediate nodes (reified events or states). This approach does not require syntax extensions and is fully RDF-compliant, but adds modelling overhead by increasing the graph's depth and indirection.

Overall, the models vary significantly in the extent to which they modify or extend the standard RDF syntax. tRDF and Named Graphs introduce no or only minor changes to the RDF structure and are considered highly interoperable and easily adoptable. RDF* requires extended syntaxes for full usability, however, it can be easily translated into standard RDF. Singleton Property maintains RDF compliance but introduces a large number of unique predicates, complicating maintenance and tool support. 4D Fluents requires explicit modelling of time slices and temporal linkage structures, adding a significant representational layer. N-ary Relations remain syntactically within RDF, but their reliance on intermediate reified nodes and objectified relations create deep graph structures.

---

[12] `https://www.ontotext.com/blog/rdf-reification-wikidata/`

4.4 REQ4: Model Semantics

This requirement considers whether an approach can be expressed using standard RDF(S) semantics or whether it relies on more advanced semantic frameworks. This is relevant in CH contexts, where the modelling of temporal change often involves complex identity, classification, and constraint structures. The degree of semantic support influences both the expressiveness and the practical feasibility of a given approach.

When incorporating temporal information into RDF, the approaches vary in how much semantic richness they require. Some approaches, e.g. tRDF and RDF* stay within RDF(S) and aim for lightweight integration. For instance, RDF* uses embedded triples to capture temporal metadata while preserving compatibility with the RDF(S) model. These approaches are advantageous in settings with limited semantic infrastructure or tool constraints. 4D Fluents, N-ary Relations, and Named Graphs can be implemented using RDF(S), but are typically used with OWL to capture more complex semantics. For example, 4D Fluents introduce temporal slices of entities, which require properties like fl:tsTimeSliceOf and class axioms to maintain coherence between temporal parts parts and the main static entity. Such semantics are difficult to enforce purely in RDF(S) without OWL constructs. Similarly, N-ary Relations rely on intermediate nodes to connect entities and their temporal qualifiers. RDF(S) can represent this structure syntactically, but cannot express constraints like functional or cardinality restrictions on these connections. Named Graphs offer a standardised RDF extension for contextualising triples, but RDF(S) provides no native means of reasoning across graphs or expressing constraints between them. Semantic interpretation of graph boundaries or relations between graphs must be handled externally or with custom logic. Singleton Property introduces a new predicate for each contextualised statement, and requires an explicit semantic mapping (via rdf:singletonPropertyOf) to relate each singleton property to its generic property. This mapping must be explicitly interpreted to preserve entailment.

For the purposes of evaluation, this requirement is assessed in isolation, focusing solely on the semantic expressiveness required to support temporal modelling. From this perspective, an approach that achieves temporal entailment with simpler semantics (e.g., RDF(S) rather than OWL or rule-based logic) is considered advantageous in terms of conceptual and technical overhead. However, this does not imply that more expressive approaches are weaker overall. In practice, the semantics requirement must be balanced with usability, performance, and institutional tooling constraints.

4.5 REQ5: Query Language

In this section, the aim is not to analyse how different SPARQL engines handle RDF data in practice (for this see, e.g. [1]) but to provide overview of how querying is achieved when using different RDF extensions and approaches. By understanding the querying mechanisms associated with each approach, their usability and compatibility with existing semantic web technologies can be better evaluated. While REQ5 and REQ3 are closely related, they capture two distinct dimensions of technical compatibility. REQ3 addresses whether an approach adheres to standard RDF syntax, while REQ5 focuses on whether it remains compatible with native SPARQL querying workflows.

Approaches such as N-ary Relations and tRDF replace standard relations by introducing new concepts – `rdf:Statement` for tRDF and `:Relation` for N-ary Relations – and associating these with temporal labels. This additional layer of abstraction complicates querying because it requires working with these new entities or statements that encapsulate triples, rather than querying the triples directly. As a result, the model's complexity increases, making the querying process less intuitive. Named Graphs, on the other hand, use standard static predicates but involve creating a number of graphs, each associated with unique timestamps. Managing these numerous graphs can lead to graph overload, making the querying process cumbersome. The need to handle many individual graphs can result in performance issues and increased difficulty in formulating queries.

Despite these complexities, these approaches can be queried using native SPARQL without requiring extensions beyond those already supported for standard RDF. In these cases, temporal information is either embedded directly within triples using RDF reification (as in tRDF) or encapsulated within named graphs, which SPARQL naturally supports. Thus, these approaches score high in terms of compatibility with native SPARQL due to their straightforward representation of temporal data.

RDF* necessitates an extension of standard SPARQL, SPARQL*, to handle nested triples effectively. Although more and more query engines today provide support of SPARQL* (e.g. GraphDB[13], Apache Jena[14]), it is crucial for users choosing the RDF* approach for modelling their temporal data to be aware of the querying capabilities of the specific engine they are planning to use. This awareness ensures that the chosen engine can fully leverage RDF*'s enhanced expressiveness without compromising performance or functionality in querying.

Singleton Property, on the other hand, does not require a SPARQL extension. Instead, it can be integrated within the existing SPARQL framework, allowing for the direct association of (temporal) metadata with RDF triples without modifying the query language itself. However, the use of Singleton Property introduces its own complexities in querying. When dealing with datasets that incorporate Singleton Properties, several considerations come into play [46]. In particular, the nature of the dataset significantly influences the querying process. For datasets containing only Singleton Property triples, standard SPARQL queries may produce empty results due to the absence of traditional RDF triples. To address this, it is essential to ensure that 1) the query engine supports RDFS inference to compute the RDFS closure, and 2) that the schema includes the specific axiom `<rdf:singletonPropertyOf rdfs:subPropertyOf rdfs:subPropertyOf>`. On the other hand, when a dataset comprises both RDF triples and Singleton Property triples, querying becomes more straightforward, but can introduce redundancy. Including inferred RDF triples derived from Singleton Properties can enhance query performance; however, it may also lead to inconsistencies if the data triples are removed while the Singleton Property triples remain. This highlights a trade-off between achieving efficient query performance and maintaining dataset consistency. Understanding these dynamics is crucial for effectively managing and querying datasets that utilise Singleton Property approach, ensuring a balance between query efficiency and data integrity.

Similarly, 4D Fluents involve more intricate representations where entities are often associated with time slices or involve multiple relationships that go beyond simple triple patterns. In such cases, while native SPARQL can still be used, it often requires custom solutions or more complex query patterns to effectively retrieve the desired information. For instance, querying relationships like `<:John@1908 :wife :Mary@1908>` requires searching through instances of `:TemporalSlice` instead of `:Person`, complicating the query structure. Additionally, relationships must often be explicitly modelled to account for their temporal dimensions, introducing redundancy and further complicating queries.

In contrast, some approaches go even further, requiring entirely new query languages to manage the complexities introduced by their specific temporal representations. For example, Annotated RDF or Extended 4D Fluents may require dedicated languages like AnQL [43] or TOQL [9], respectively, because the standard SPARQL framework cannot easily express the nuanced semantic

---

[13] `https://graphdb.ontotext.com/documentation/10.7/rdf-sparql-star.html`

[14] `https://jena.apache.org/documentation/RDF*/`

relationships or nested temporal structures these models introduce. The need for these specialised languages arises from the fact that these approaches fundamentally change how data is structured and how temporal information is linked to RDF entities, making standard SPARQL insufficient for querying without significant extensions.

4.6 REQ6: Use of RDFS/OWL Reasoning

Incorporating temporal context into RDF models often requires extending the basic RDF framework, which can have significant implications for OWL and RDF(S) reasoning. Reasoning relies on well-defined logical constructs, such as class hierarchies, property relationships, and inverse properties, to infer new knowledge from existing data. However, when temporal dimensions or other complex structures are introduced, these logical constructs can become less effective or even incompatible.

4D Fluents maintain strong RDF(S)/OWL reasoning capabilities by integrating temporal dimensions within the OWL framework itself. This approach represents entities and their temporal parts as distinct classes preserving standard RDF triple structure and exploiting original properties. Thus, OWL property constructs like inverse properties and transitivity are fully supported. However, on the other hand, when changing the original class belonging, reasoning over class hierarchies becomes more complicated. For example, given the RDF data:

```
:Director rdfs:subClassOf :Person .
:director rdfs:range :Director .
:play@1938 :director :WalterFriedrichPeters@1938 .
:WalterFriedrichPeters@1938 :timeSliceOf :WalterFriedrichPeters .
```

A reasoner will infer that `:WalterFriedrichPeters@1938` is a `:Director` and, by subclass logic, a `:Person`. However, the reasoning engine does not automatically infer that `:WalterFriedrichPeters` (the generic entity) is a `:Director` or `:Person` just because its time slice `:WalterFriedrichPeters@1938` is. To enable these inferences to the generic entity, additional rules or custom reasoning mechanisms are needed beyond standard OWL.

Named Graphs do not change the structure of RDF triples but organise triples into separate graphs within an RDF dataset. This allows standard OWL reasoning to be applied within individual named graphs. However, practical support depends on the triple store's ability to perform reasoning and query processing over named graphs. In particular, querying patterns that extensively rely on the graph component may lead to performance degradation, as the efficiency of querying within named graphs varies significantly across implementations.

Due to reification in tRDF, the original triple's properties are encapsulated within the `rdf:Statement` instance, making them opaque to standard RDF(S)/OWL reasoning. For example, the triple `<:play :director :WalterFriedrichPeters>` is reified as `<:statement1 rdf:predicate :director .>`. This encapsulation turns the original properties into objects of new triples, which disrupts direct reasoning. As a result, defining `<:director owl:inverseOf :directed>` does not provide any inference results because OWL cannot infer across the boundary created by reification. Furthermore, reification itself does not imply the existence of the original triple, nor does the original triple imply the existence of its reified version, significantly limiting the ability to reason about the original triple's metadata or related relationships. Similarly, in the N-ary Relation approach, the property `:director` is transformed into an entity rather than a direct property, e.g. `:Directing`, and the related information is linked to the new entity, e.g. `<:Directing :directorEntity :WalterFriedrichPeters>`. This transformation results in the loss of the direct connection between `:play` and `:WalterFriedrichPeters`, as this relationship is now mediated through the new entity `:Directing`. Consequently, enabling reasoning in this context would require more sophisticated reasoning rules and potentially custom constructs to interpret these indirect relationships effectively.

In the Singleton Property approach, a unique instance of the property is created for each specific relationship to represent temporal information. For example:

```
:Director rdfs:subClassOf :Person .
:director rdfs:range :Director ;
          owl:inverseOf :directed .
:director_1938 rdf:singletonPropertyOf :director ;
               :validIn "1938" .
:play :director_1938 :WalterFriedrichPeters .
```

The property `:director_1938` is a singleton property of `:director`, meaning it is treated as a unique instance associated with the year 1938. However, because it is not explicitly defined as a sub-property of `:director`, the `owl:inverseOf :directed` relationship does not automatically apply. This means that the triple `<:WalterFriedrichPeters :directed :play>` will not be inferred. Similarly, the range of `:director_1938` does not automatically lead to the inference that `:WalterFriedrichPeters` is a `:Director` or a `:Person` because the sub-class relationship depends on recognising the singleton as a sub-property of the generic property. To enable these kinds of standard OWL inferences requires several custom solutions to enable efficient reasoning, e.g. defining inverse properties for each singleton, explicitly defining each singleton property as a sub-property of the original property, defining custom reasoning rules, e.g. `<rdf:singletonPropertyOf rdfs:subPropertyOf rdfs:subPropertyOf .>`, etc. However, each additional rule or definition significantly increases the number of triples in the dataset. This not only complicates the RDF model but also leads to performance concerns, as reasoning engines must process a larger and more intricate set of rules and triples. As a result, reasoning with singleton properties in an RDF context becomes cumbersome and less efficient.

RDF* introduces a mechanism for annotating triples with (temporal) information, which results in a significant challenge in RDF(S)/OWL reasoning due to RDF*'s referential opacity. In RDF*, there is a distinction between a quoted triple and an unquoted triple, as shown in the following example:

```
:Director rdfs:subClassOf :Person .
:director rdfs:range :Director ;
          owl:inverseOf :directed .
«:play :director :WalterFriedrichPeters» :validIn "1938" .
```

Here, the quoted triple `«:play :director :WalterFriedrichPeters»` represents a specific statement and is associated with the year 1938. However, because of RDF*'s referential opacity, RDF(S)/OWL reasoning does not treat this quoted triple the same as the unquoted triple `<:play :director :WalterFriedrichPeters>`. This means that the inverse property relationship or a class belonging from the range definition of `:director` does not apply to the quoted triple, and the inference, e.g. `«:WalterFriedrichPeters :directed :play»` is not generated. This limitation arises because OWL reasoners do not inherently recognise a quoted triple as denoting the same relationship as its unquoted counterpart. This distinction exists for a reason: the quoted triple is only valid within its specified context, meaning the relationship it describes is context-dependent. As a result, it cannot be inferred that the inverse property necessarily applies to the unquoted version, as this relationship may not hold universally outside the given context. To address this for specific contexts, RDF* introduces the concept of Transparency Enabling Properties

(TEPs[15]), which can be used to selectively enable referential transparency for specific properties. For example, stating `<:validIn rdf:type RDF*:TransparencyEnablingProperty>` will ensure that a reasoner considers the quoted triple as equivalent to the unquoted triple, thus allowing the inverse relationship and other inferences to apply. However, this selective transparency only applies within the context of the RDF* graph where the TEP is defined, which can become cumbersome when working with multiple RDF* graphs. While TEPs provide a mechanism to address the limitations of RDF*'s referential opacity, their use requires careful consideration by users in determining which properties should be defined as transparent. Additionally, users must ensure that reasoning engines are correctly configured to interpret these properties, adding another layer of complexity to the reasoning process. This complexity is particularly seen in cases where RDF* data from various sources is integrated or queried together, as inconsistencies in TEP definitions across graphs could lead to undecidability in reasoning results.

Overall, evaluating various approaches against this criterion aims to reveal their effects on standard reasoning. The discussion shows that all approaches require some form of adaptation to achieve effective reasoning. While certain solutions, such as TEPs in RDF* or Named Graph reasoning mechanisms, are developed directly by the authors, others necessitate user-created custom rules. Therefore, when selecting an approach, it is essential to consider not only the built-in reasoning capabilities but also the potential need for user-defined modifications to ensure optimal performance.

4.7 REQ7: Ease-of-Use

In CH, knowledge graph users are typically domain experts rather than computer scientists or ontology engineers. These users may lack extensive technical training in data modelling but are nonetheless expected to integrate, interconnect, and retrieve complex historical data through semantic technologies. Therefore, evaluating the *ease of use* of temporal modelling approaches is crucial for ensuring broad adoption and meaningful participation in semantic knowledge engineering across CH communities.

To assess this requirement, a targeted usability survey is conducted. This survey is designed as a study to capture the intuitions and practical preferences of CH professionals. Its goal is to complement technical assessment with user-centred insights, particularly regarding which models are perceived as most intuitive and accessible. This aligns with the broader objective of our work, in particular to support informed, context-sensitive modelling decisions through decision-support mechanisms tailored to CH practice.

- **Survey Procedure.** The survey took approximately 10 minutes to complete. Participants first answered demographic questions about their educational and domain background, as well as their experience with data modelling or knowledge engineering. The core part of the survey involved a set of pairwise modelling comparisons. For each comparison, participants were asked to select which they found easier to understand, more intuitive, or generally better suited for their work. They were encouraged to briefly justify their preferences, providing qualitative insight into their choices.
- **Survey Setup.** To ensure consistency and comparability across participants, the survey used a fixed example: a simple natural language sentence describing a historical change "Until the year 1809, the street was called 'Roßmarkt'. In 1809, it was officially renamed to 'Adlerstraße'."[16]

---

[15] `https://www.w3.org/2021/12/RDF*.html#RDF*-vocabulary`

[16] This example is adapted from data used in the TRANSRAZ project. The primary historical sources in this project are the annual address books (*Adreßbücher*) of the city of Nuremberg. Changes in street names over time, such as this one, represent a typical temporal modelling scenario encountered in the data.

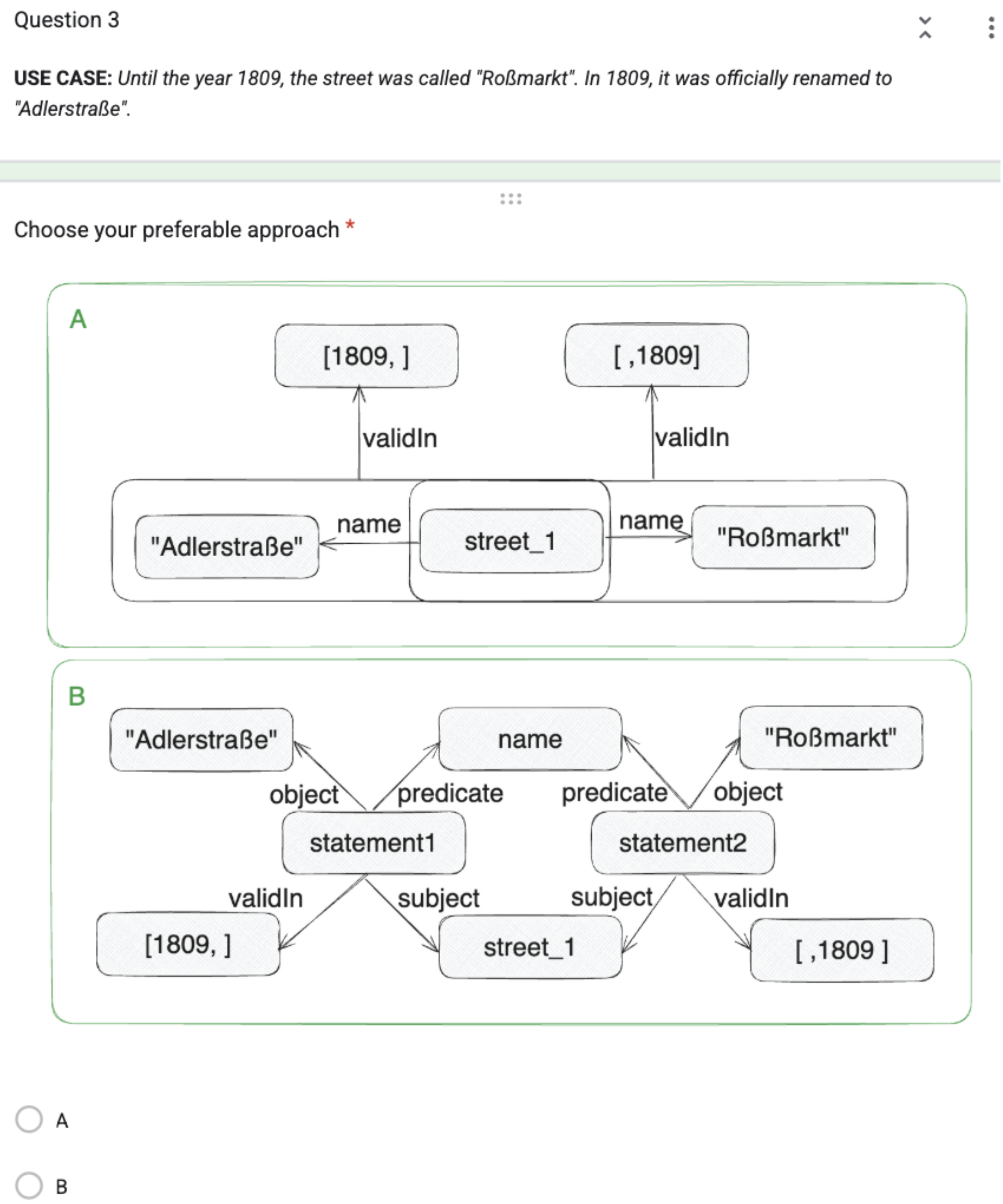
This sentence was then represented using different modelling approaches (e.g., 4D Fluents, RDF*, tRDF, etc.), which were shown two at a time for pairwise comparison, see Figure 5. Each representation was schematically visualised and participants were asked to evaluate which version they found easier to understand or more intuitive. The same example was used throughout the survey to isolate the modelling approach as the only varying factor and prevent differences in content from influencing the results. Participants compared each approach against every other, resulting in a full pairwise comparison across all methods. To reduce learning effects and avoid biased results due to repetition or familiarity, the ordering of the pairs was randomised as much as possible. It was specifically ensured that no representation was shown in two consecutive questions, and that each approach appeared roughly equally in both the top and bottom positions across different comparisons.

- **Participants.** The survey was completed by 22 participants with diverse backgrounds. This mix enabled us to explore how ease-of-use perceptions vary across CH professionals and technically trained participants. Key statistics are as follows[17]:
  - **Domain Background:** 58% cultural heritage experts (History, Digital Humanities, Archaeology, Numismatics, Musicology, Theology, etc.), 38% computer scientists working on CH data and technologies (Semantic Web, ontologies, data mining, etc.).
  - **Educational Background:** 54.5% hold a PhD, 41.4% a Master's degree.
  - **Modelling Experience:** 81.8% reported prior experience with data modelling or knowledge engineering.
- **Results.** As shown in Figure 6, 4D Fluents emerged as the overall favorite, praised for its intuitive and clear structure. RDF* also received strong support, particularly for its syntactic clarity and minimal complexity. Named Graphs were well received among cultural experts, likely due to their conceptual similarity to archival structures. Meanwhile, tRDF and N-ary Relations were appreciated by some participants for their explicitness and flexibility, though these approaches were generally seen as more complex. The direct comparison between 4D Fluents and RDF* yielded nearly even results, with RDF* slightly ahead (12 vs. 10 votes), suggesting that both are strong candidates from a usability standpoint.

Overall, the survey contributes valuable insights into how different temporal modelling approaches are perceived by CH experts and technical experts. While not intended as a formal empirical evaluation, the survey captures practitioner perspectives that are often overlooked in technical assessments. This user-centred viewpoint complements the project requirements and supports the broader goal of aligning technical choices with the practical needs and preferences of CH experts. However, several limitations must be acknowledged. Firstly, the survey focuses on simplified visualisations of RDF representations. The decision to show schematic visualisations rather than raw RDF syntax (e.g., Turtle or RDF/XML) was made to make the survey accessible to participants regardless of their technical background. While this lowers the barrier to participation, it also abstracts away some of the actual complexity of the models, possibly affecting how effort or clarity is perceived. Participants may judge models more on layout or visual simplicity than on how intuitive the representations are. Secondly, the representations are derived from a single example, which introduces a risk of learning effects over time. While pair order and positioning were randomised and balanced to reduce this, the possibility of biases remains. Thirdly, the survey measures perceived ease-of-use rather than actual modelling performance; it does not involve task-based validation such as modelling data or querying a knowledge graph.

---

[17] All survey statistics and results can be accessed here: `https://github.com/sashabruns/RDF-Extensions-for-Cultural-Heritage/blob/main/CR7_survey_statistics.ipynb`



Question 3

USE CASE: *Until the year 1809, the street was called "Roßmarkt". In 1809, it was officially renamed to "Adlerstraße".*

Choose your preferable approach *

○ A

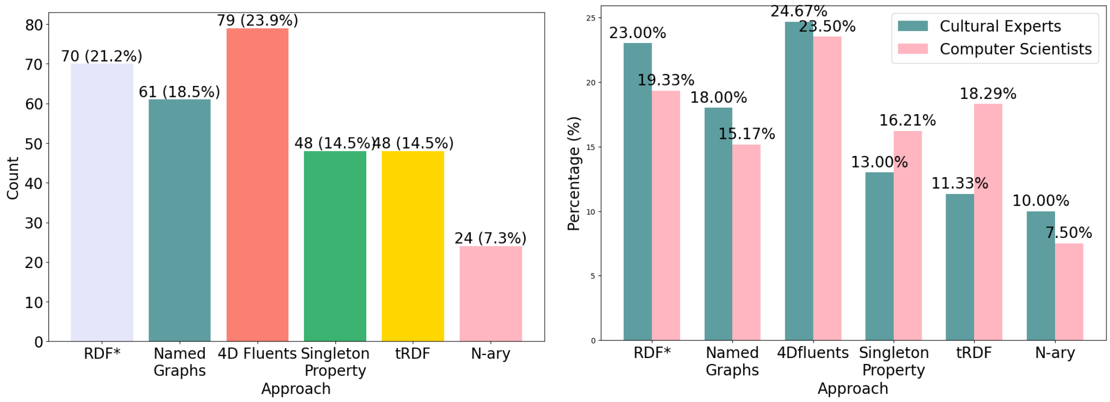
○ B

Figure 5 Example survey question used to evaluate ease-of-use.

Future studies could expand this preliminary analysis by incorporating task-based evaluations, larger and more diverse samples, and scenarios that engage with full modelling pipelines.

4.8 Limitations and Future Work

This section provides a structured analysis of temporal modelling approaches for RDF-based CH KGs, evaluated against concrete requirements derived from real-world CH use cases. By investigating and discussing requirements one by one, model comparison is achieved in a systematic and use-case-aware manner, moving beyond purely formal or purely empirical evaluation strategies.



(a) Overall preferences of the participants.

(b) Results based on the domain background.

Figure 6 Survey results evaluating the ease-of-use of each modelling approach.

There are several limitations of the study that must be acknowledged. First, while the evaluation captures a broad range of temporal modelling approaches, it is not exhaustive. The focus lies on representative RDF extensions, however, more event-based and foundational models may offer alternative perspectives that deserve future inclusion. Second, the evaluation is based on schematic examples than large-scale deployment or empirical benchmarking. While this enables comparison, it does not fully capture the implementation costs, scalability issues, or other impacts on modelling and querying tasks. Third, the ease-of-use survey with CH practitioners complements technical evaluations, however the survey scope is limited to pairwise comparisons on a single example. Broader user studies are beneficial to validate these findings at scale.

Future work will address these limitations by expanding the comparison to include other temporal data models. Additionally, in future work, the current analysis will be complemented with a more extensive empirical evaluation using representative datasets from CH domains. While the present study offers a structured comparison, modelling scenarios, and expert feedback, it focuses primarily on qualitative aspects. Future research will extend this by applying the evaluated approaches to larger and more diverse datasets, measuring practical outcomes such as scalability, query complexity and reasoning performance. Such an empirical layer will enhance the generalizability of the findings and offer deeper validation of how different temporal models perform under real-world conditions.

5 Decision Support for Selecting Temporal Modelling Approaches

The discussion of temporal modelling approaches, presented in Section 4, demonstrates how each approach responds to a specific requirement, but it still lacks clear guidelines on which approach to choose. Every extension has its own set of strengths and weaknesses, making it difficult to give a recommendation, as they cannot be easily reduced to a straightforward or binary answer when assessing their suitability for specific requirements. To provide a more nuanced comparison, this section introduces a structured decision support mechanism. This comprehensive tool aims at providing systematic guidelines for practitioners in evaluating approaches by prioritising the requirement based on their specific project needs and diverse scenarios, and navigating the decision-making process for temporal data modelling. The following subsections present: i) the scoring methodology used to quantify criteria, ii) three illustrative scenarios grounded in real-world CH project use cases, and iii) a simple ranking mechanism that helps identify the most fitting approach based on project-specific priorities.

5.1 Design and Methodology

Recognising that the choice of a temporal modelling technique is highly use case dependent, the decision-support aims to provide a nuanced analysis rather than a simplistic binary assessment. By defining specific metrics for each requirement, the tool enables a detailed and systematic comparison of how well different approaches meet project/data requirements. This methodology not only emphasises the relative advantages and limitations of each approach but also supports informed decision-making by aligning the evaluation process with the unique needs of each use case.

This methodology involves i) assigning raw scores to approaches per requirement, ii) applying normalisation to enable comparability, iii) incorporating weights to reflect project-specific priorities, and iv) computing similarity to an ideal solution.

- **Scoring and Scales.** Each modelling approach is assigned a raw score per requirement using requirement-specific metrics. These metrics are not uniform across all requirements: for example, *REQ2: Storage and Scalability* is based on empirical thresholds of triple counts, while *REQ1: Temporal Expressivity* and *REQ3: RDF Syntax Extension* use more nuanced qualitative scales. This variability reflects the fundamentally different natures of the requirements: some (e.g., scalability) are quantifiable and bounded, while others (e.g., expressivity) require expert judgment across a conceptual spectrum. The full set of scoring scales is presented in Table 1.
- **Normalisation.** To enable cross-requirement comparison, all raw scores (see Table 2) are min-max normalised to a common scale between 0 and 1, where 0 represents the best achievable performance. For example, a raw score of 1 is normalised to 0, and higher raw values are scaled proportionally based on the range defined for that requirement.
- **Weighting.** Users can assign weights to each requirement to reflect their relative importance for a given use case. The weights sum to 1.0 and allow the decision-support system to tailor recommendations to specific project contexts. The weighted normalised scores are then combined for each approach.
- **Distance to Ideal.** Finally, the Euclidean distance between each approach's weighted score vector and the ideal vector `[0.0, 0.0, ..., 0.0]` is calculated. The approach with the smallest distance is considered the best fit for that scenario.

This methodology yields a transparent and flexible mechanism for assessing temporal modelling approaches based on both domain-specific requirements and project-specific constraints.

This decision-support system is designed to support both technical implementers and CH practitioners in making informed, transparent, and requirement-driven decisions when modelling temporality in knowledge graphs. Its value lies not only in ranking options, but in making the trade-offs between different modelling priorities explicit.

5.2 Proof-of-Concept: Scenarios and Applications

This section presents a prototype decision-support tool developed to demonstrate how the proposed decision-support tool can aid in selecting a temporal RDF modelling approach based on project-specific priorities. The tool allows users to assign custom weights to the identified requirements (see Section 2.2) and visualises the alignment of each approach with a perfect score across those requirements. The goal of this prototype is not to prescribe definitive modelling choices, but to illustrate the potential of a structured, transparent, and comparative evaluation mechanism tailored to CH needs. The tool and the example scenarios are available online[18].

---

[18] `https://github.com/sashabruns/RDF-Extensions-for-Cultural-Heritage/blob/main/framework.ipynb`

Table 1 Requirements, scoring bases, and raw score scales used in the decision-support system.

Requirement	Scoring Basis	Raw Score Scale				
		1 (best)	2	3	4	5 (worst)
REQ1: Temporal Expressivity	Support for REQ1.1–REQ1.4	Full semantics	Intervals + relations	Intervals only	Limited support	No support
REQ2: Storage and Scalability	Triple count for representing facts	Minimal increase	Moderate increase	High increase	–	–
REQ3: RDF Syntax Extension	Degree of RDF syntax modification	None / minimal	Moderate (compatible)	Significant (custom tooling)	Complex (new layers)	RDF overhaul
REQ4: Model Semantics	Semantic expressiveness required	RDF(S)	OWL	Custom / rule-based	–	–
REQ5: Query Language	Compatibility with SPARQL	Native SPARQL	SPARQL + minor adaptation	Requires SPARQL extensions	Requires custom query	–
REQ6: RDFS/OWL Reasoning Support	OWL/RDFS reasoning capability	Full support	Moderate support	Limited support	No support	–
REQ7: Ease-of-Use	Expert judgement in CH context	Top-rated	2nd most usable	3rd in preference	4th in preference	Least usable

Table 2 Raw scores of temporal modelling approaches per requirement (lower is better).

Requirement	Temporal Modelling Approach					
	tRDF	4D Fluents	RDF*	Named Graphs	N-ary Relations	Singleton Property
REQ1: Temporal Expressivity	2	1	4	4	4	4
REQ2: Storage and Scalability	3	3	1	1	2	2
REQ3: RDF Syntax Extension	1	5	3	2	5	4
REQ4: Model Semantics	1	2	1	2	2	3
REQ5: Query Language	1	2	3	1	1	2
REQ6: RDFS/OWL Reasoning Support	2	2	2	1	3	3
REQ7: Ease-of-Use	4	1	2	3	5	4

To demonstrate its functionality, three fictional scenarios inspired by recurring patterns and decision-making challenges are introduced. These scenarios are not directly extracted from specific project documentation but were shaped through iterative discussions with CH practitioners and technical partners during past collaborations. They reflect recurring tensions between expressivity, reasoning, technical constraints, and usability that shape real-world modelling decisions. While inspired by such discussions, these scenarios serve only as illustrative examples, designed to showcase the flexibility of the tool rather than document exact real-world processes. The assignment of weights in each scenario reflects typical trade-offs encountered in the field rather than normative recommendations.

Figure 7 visualises the results. Each chart shows how well each approach performs against a normalised "perfect" score, with the Euclidean distance used to rank the most suitable approach for each set of requirements.

Scenario 1: Large-Scale Data Integration for Historical Archives

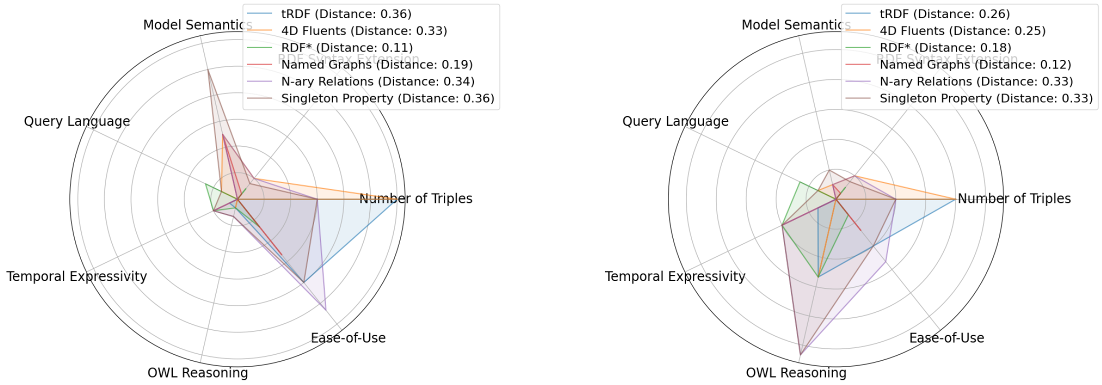
- **Context:** Integration of vast archival datasets, focusing on temporal annotation and federated querying of historical events.
- **Project Requirements:**
  - *Scalability:* Efficient handling of high triple volumes is critical.
  - *Model Semantics:* Lightweight semantics are preferred for performance reasons.
  - *Ease-of-Use:* Accessibility for a diverse set of researchers is key.
  - *Query Language:* SPARQL compatibility ensures interoperability.
  - *Temporal Expressivity:* Sufficient to support timeline construction.
- **Criteria Weighting:**
  - Storage and Scalability: 0.3
  - Model Semantics: 0.25
  - Ease-of-Use: 0.2
  - Query Language: 0.1
  - Temporal Expressivity: 0.05
  - OWL/RDF(S) Reasoning: 0.05
  - RDF Syntax Extension: 0.05
- **Preferred Approach:** RDF* (see Figure 7a).

Scenario 2: Integration and Reliability Checking for Historical Data of Nuremberg

- **Context:** Aggregating and validating data from diverse CH sources related to the historical evolution of a city.
- **Project Requirements:**
  - *OWL/RDF(S) Reasoning:* Needed to support inference and consistency checks.
  - *Scalability:* Must handle large-scale data ingestion and reasoning.
  - *Temporal Expressivity:* Important for tracing entity changes over time.
- **Criteria Weighting:**
  - OWL/RDF(S) Reasoning: 0.4
  - Storage and Scalability: 0.2
  - Temporal Expressivity: 0.1
  - Query Language: 0.1
  - Ease-of-Use: 0.1
  - Model Semantics: 0.05
  - RDF Syntax Extension: 0.05
- **Preferred Approach:** Named Graphs (see Figure 7b).

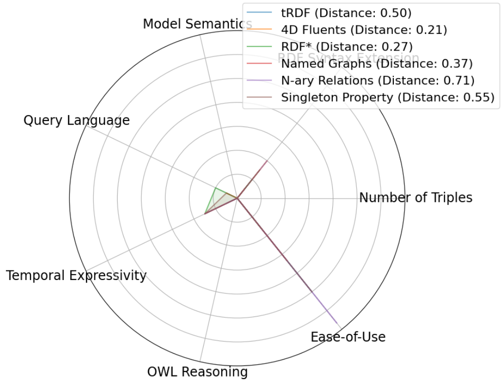
Scenario 3: User-Friendly Temporal Data Management for Digital Humanities

- **Context:** Supporting archivists in managing evolving classification schemes with minimal technical complexity.
- **Project Requirements:**
  - *Ease-of-Use:* Primary users are non-technical domain experts.
  - *RDF Syntax:* Adherence to familiar, standard RDF is important.
  - *Query Language:* Native SPARQL must be supported.
  - *Temporal Expressivity:* Needed to annotate changing labels and roles.
- **Criteria Weighting:**
  - Ease-of-Use: 0.5
  - RDF Syntax Extension: 0.2
  - Query Language: 0.15
  - Temporal Expressivity: 0.15
  - Model Semantics: 0.00
  - Storage and Scalability: 0.00
  - OWL/RDF(S) Reasoning: 0.00
- **Preferred Approach:** 4D Fluents (see Figure 7c).



(a) Scenario 1: Archival data integration.

(b) Scenario 2: City-wide integration and validation.



(c) Scenario 3: Archivist-friendly modelling.

Figure 7 Scenario-specific ranking of modelling approaches based on requirement weights and normalised distance from the ideal score.

Discussion

The comparative evaluation of temporal modelling approaches across three different cultural heritage scenarios provides a more nuanced understanding of their suitability under varying priorities. The decision-support prototype quantifies how closely each approach aligns with the requirements defined for a specific use case and offers a transparent rationale behind each recommendation.

In **Scenario 1**, where scalability, lightweight semantics, and ease-of-use are emphasised, RDF* is selected as the preferred solution. Its compact modelling and minimal syntax extension match the needs of high-volume archival integration. Named Graphs follow closely, with similar strengths but slightly higher semantic complexity. tRDF and Singleton Property scored the highest distances, mainly due to their larger triple count and more complex RDF syntax, which make them less suitable for large-scale deployments with non-technical users. In **Scenario 2**, which requires reasoning capabilities and temporal expressivity for validating integrated data, Named Graphs turned out to be the top choice. Its native compatibility with OWL reasoning and straightforward contextualisation align well with the scenario's focus on consistency and logical inference. Singleton Property and N-ary Relations rank lower, as their reasoning support is limited or dependent on custom semantics, which reduces their effectiveness in tasks involving inference over complex data relationships. In **Scenario 3**, which prioritises usability and accessibility for non-technical users such as archivists, 4D Fluents scores the lowest distance and is identified as the preferred approach. Despite its triple verbosity, it offers intuitive temporal modelling, straightforward RDF syntax, and high usability in describing classification schemes over time. RDF* performs nearly as well due to its compact and intuitive syntax. In contrast, N-ary Relations and Singleton Property exhibit higher distances, reflecting the cognitive overhead associated with their more complex or less standard modelling constructs, which may hinder adoption in scenarios requiring minimal technical knowledge.

This analysis shows how weighting domain-specific requirements enables tailored recommendations. While RDF* and Named Graphs often appear in top positions, their suitability is not universal but context-sensitive. Similarly, approaches that score lower in the discussed scenarios are not universally less valid but may impose trade-offs that are mismatched with the scenario-specific constraints. By making these trade-offs explicit, the tool supports transparent, criteria-driven decision-making for temporal representation in CH projects.

Overall, the prototype illustrates how decision-support tools can assist in navigating the complex landscape of temporal RDF modelling. The Euclidean distance from the ideal solution offers a quantifiable and interpretable signal about the relative suitability of different approaches under competing priorities. This approach does not replace human judgment or detailed project-specific modelling design, but it enables a more structured comparison of alternatives, which remains underexplored in CH-oriented RDF practice.

5.3 Limitations and Future Work

While the decision support mechanism presented in this paper offers a structured way to align temporal modelling approaches with project-specific requirements, several limitations remain and thus inform directions for future work.

First, although the current implementation is intentionally lightweight and easy to apply, it requires users to assign numerical weights to requirements based on their relevance to the project context. This step can be challenging, particularly for users without prior experience in such decision-making processes. To address this, alternative ways of prioritising requirements are to be explored. One direction is to replace or complement numerical weighting with a ranking-based approach, where users simply rank requirements in order of importance. This method could reduce cognitive load and encourage more intuitive engagement, especially for non-technical stakeholders. Second, the evaluation currently relies on a fixed scoring scheme, which is derived from conceptual and practical analysis of the approaches but has not yet been empirically validated. In future work, we intend to refine this scoring through systematic benchmarking and testing, capturing performance characteristics such as reasoning behaviour, scalability, and actual ease-of-use across CH projects. Third, since some temporal approaches are closely scored, slight changes in requirement priorities can shift the final recommendation. We therefore plan to incorporate sensitivity analysis into the tool to highlight how robust a recommendation is to small variations in input priorities. This would help users better understand trade-offs and the comparative strengths of alternative approaches. Lastly, the prototype is currently implemented as a Jupyter notebook, which assumes some technical familiarity. We are planning to develop a more accessible graphical user interface and evaluate the tool in real-world CH infrastructure planning settings, with the goal of supporting broader adoption and iterative refinement based on user feedback.

6 Summary and Conclusions

This works presents a systematic evaluation of RDF-based approaches for representing temporal information in Cultural Heritage Knowledge Graphs. Temporal representation in CH is especially challenging due to the need to model evolving entities, uncertain or incomplete historical data, and richly contextualised timelines. To structure this investigation, six RQs were formulated, which guided both the theoretical foundation and evaluation of the study.

RQ1: What RDF-based approaches exist for temporal representation? was addressed by investigating the current practices for representing temporal information in RDF, while key RDF extensions such as RDF*, tRDF, Named Graphs, N-ary Relations, 4D Fluents, and Singleton Property were identified. In addition to the main focus of the research – RDF extensions – alternative approaches for representing time in RDF, such as event-based models and temporal description logics, were briefly explored. These methods differ from RDF extensions in their strategies for handling temporal information, whether through focusing on event-driven representations, or applying logic-based temporal reasoning. This allowed a more comprehensive view of the methodologies available for managing temporality.

Suitability of the approaches for CH KGs was a central concern throughout this study. While many temporal modelling approaches originate from general-purpose use cases, their application in CH remained open. For this, answering **RQ2: What temporal requirements arise from Cultural Heritage use cases?**, in Section 2, temporal modelling needs that are specific to CH contexts were identified. Based on four CH use cases, data and their project requirements, seven requirements were presented. These include: *REQ1:* Temporal Expressivity – to capture contextual changes over time and to represent time statements, uncertainty, and interval relations; *REQ2:* Storage and Scalability – to ensure that models can handle large CH datasets efficiently; *REQ3:* RDF syntax extension – to investigate which model and how they extend RDF syntax to entail temporality, and how this influences interoperability and data integration; *REQ4:* Model semantic – to define the level of logic and inference needed for the extension; *REQ5:* Query language – to evaluate support for expressive and practical querying, especially with SPARQL; *REQ6:* Use of OWL reasoning – to observe how extending RDF with temporal constraint influence the standard reasoning mechanisms; and *REQ7:* Ease-of-Use – to assess accessibility for domain experts with limited technical background. Together, these criteria provide a structured lens through which different modelling approaches can be assessed in terms of their applicability and usability in CH contexts.

To complement the technical evaluation and answer **RQ3: How intuitive and usable are different temporal modelling approaches for CH domain experts?**, a small-scale usability survey (Section 4.7) was conducted among CH and technical experts. The study focused on perceived intuitiveness using a controlled example and schematic visualisations. While 4D Fluents and RDF* emerged as the most preferred approaches overall, N-ary Relations were perceived as less intuitive and scored lower in terms of ease-of-use.

Section 4 was dedicated to answer **RQ4: How can temporal modelling approaches be evaluated based on CH-specific requirements and technical criteria?**. It presented a detailed comparison across seven identified REQs. The evaluation showed that no temporal modelling approach fully meets all CH requirements, but each offers specific strengths depending on project needs. RDF* and Named Graphs perform well in terms of scalability and integration into existing infrastructures (REQ2), with RDF* offering strong ease-of-use and readability (REQ7), while Named Graphs also provide solid support for query formulation and reasoning (REQ5, REQ6). 4D Fluents stands out for its rich temporal expressivity (REQ1) and was found to be highly intuitive by CH experts (REQ7), but it introduces structural overhead that negatively impacts RDF compatibility and infrastructure alignment (REQ3). The Nary Relations approach is particularly well suited for modelling complex events and states, the direct attachment of temporal information to objectified relations enables easy querying (REQ5), aligning with OWL reasoning (REQ4). It scales well (REQ2), yet falls short in representing temporal information (REQ1), extends RDF syntax (REQ3), and is not intuitive for users (REQ7). Singleton Property supports contextual metadata and integrates well with infrastructure (REQ5), but performs poorly in modelling semantic change (REQ1), introduces high overhead (REQ2), and lacks usability and readability for users (REQ7). Finally, tRDF scores well in RDF compatibility and OWL alignment (REQ3, REQ4), and moderately in supporting query formulation (REQ5), but it does not scale as well in large infrastructures (REQ2) and is less intuitive for domain experts (REQ7).

These results underscore that “suitability” in CH contexts is not only a question of expressivity but of trade-offs: between representational power and model simplicity, between formal reasoning and system compatibility, and between technical potential and practical implementation. Our findings suggest that CH projects must evaluate temporal models not just by their semantic capabilities, but also by their alignment with the project requirements and conditions, e.g. institutional tooling. In this sense, answering **RQ5: How can CH practitioners be supported in selecting a suitable temporal modelling approach?**, the decision-support tool was introduced in Section 5. Based on the outcomes of the requirement-based evaluation, the tool aims to support experts in choosing an appropriate approach based on specific project needs. It enables users to prioritise different requirements and select approaches accordingly. The decision matrix offers a structured path through complex trade-offs and is intended to serve as a practical tool for project planning.

While this work offers a structured, requirement-driven comparison of temporal RDF models, several limitations must be acknowledged. The evaluation was conducted on a limited set of example scenarios and does not include large-scale performance metrics or full CH datasets. Furthermore, the usability study was not a formal empirical validation but rather an exploratory survey intended to highlight practitioner perspectives. Future studies could extend this work by performing task-based evaluations (e.g., modelling, querying, or annotation tasks) across diverse domains. Another direction lies in expanding the range of modelling strategies considered. Event-centric models were not covered in depth here but are highly relevant for CH and deserve systematic inclusion in future evaluations. Likewise, the interplay between temporal and spatial dimensions in CH datasets presents a rich area for future exploration.

Additionally, the compatibility and interoperability of RDF extensions will be a focus of future work. The aim is to explore how effectively different RDF extensions can be aligned and evaluate the effort required to translate between various temporal modelling approaches. This is particularly important for use cases that rely on diverse extensions and require integration to enable simpler data access, discovery, and querying. Achieving such interoperability will ensure more cohesive, flexible, and efficient temporal representations within CH KGs, improving their usability across diverse applications.

Overall, this research has the potential to significantly improve how humanity preserves, accesses, and interprets the temporal dimensions of cultural heritage. It offers new means for temporal representations, and, thus, new means to engage with history and shared past, not merely as static moments but as a rich, unfolding story over time.

Resource Availability Statement

All supplementary resources are openly available on GitHub[19], including the visualisations presented in the paper, example queries supporting the discussion in Section 4, statistical analyses and insights from the user study in Section 4.7, and the implementation of the framework described in Section 5.

Declaration of AI Usage

AI-based tools by OpenAI GPT-4–class models[20] were used to support the writing process, in particular, for the linguistic refinement of the text, including spelling, grammar, punctuation, and stylistic refinement. All scientific content, research ideas and methodology are the work of the authors.

References

1 Waqas Ali, Muhammad Saleem, Bin Yao, Aidan Hogan, and Axel-Cyrille Ngonga Ngomo. A survey of RDF stores & SPARQL engines for querying knowledge graphs. *VLDB J.*, 31(3):1–26, 2022. `doi:10.1007/s00778-021-00711-3`.

2 James F. Allen. An interval-based representation of temporal knowledge. In Patrick J. Hayes, editor, *Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI '81, Vancouver, BC, Canada, August 24-28, 1981*, volume 81, pages 221–226. International Joint Conferences on Artificial Intelligence Organization (IJCAI), William Kaufmann, 1981. URL: `http://ijcai.org/Proceedings/81-1/Papers/045.pdf`.

3 James F. Allen and George Ferguson. Actions and events in interval temporal logic. *J. Log. Comput.*, 4(5):531–579, 1994. `doi:10.1093/logcom/4.5.531`.

4 Eleftherios Anagnostopoulos, Sotiris Batsakis, and Euripides G. M. Petrakis. CHRONOS: A reasoning engine for qualitative temporal information in OWL. In Junzo Watada, Lakhmi C. Jain, Robert J. Howlett, Naoto Mukai, and Koichi Asakura, editors, *17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems, KES 2013, Kitakyushu, Japan, 9-11 September 2013*, volume 22 of *Procedia Computer Science*, pages 70–77. Elsevier, 2013. `doi:10.1016/j.procs.2013.09.082`.

5 Robert Arp, Barry Smith, and Andrew D Spear. *Building ontologies with Basic Formal Ontology.* MIT Press, 2015. `doi:10.7551/mitpress/9780262527811.001.0001`.

6 Alessandro Artale and Enrico Franconi. A survey of temporal extensions of description logics. *Ann. Math. Artif. Intell.*, 30(1-4):171–210, 2000. `doi:10.1023/A:1016636131405`.

7 Alessandro Artale and Enrico Franconi. Temporal description logics. In Michael Fisher, Dov M. Gabbay, and Lluís Vila, editors, *Handbook of Temporal Reasoning in Artificial Intelligence*, volume 1 of *Foundations of Artificial Intelligence*, pages 375–388. Elsevier, 2005. `doi:10.1016/S1574-6526(05)80014-8`.

8 Franz Baader, Stefan Borgwardt, Patrick Koopmann, Ana Ozaki, and Veronika Thost. Metric temporal description logics with interval-rigid

---

[19] `https://github.com/sashabruns/RDF-Extensions-for-Cultural-Heritage`

[20] OpenAI. (2024). ChatGPT-4o [Large language model]. `https://chat.openai.com`

names. *ACM Trans. Comput. Log.*, 21(4):30:1–30:46, 2020. `doi:10.1145/3399443`.

9 Evdoxios Baratis, Euripides G. M. Petrakis, Sotiris Batsakis, Nikolaos Maris, and Nikos Papadakis. TOQL: temporal ontology querying language. In Nikos Mamoulis, Thomas Seidl, Torben Bach Pedersen, Kristian Torp, and Ira Assent, editors, *Advances in Spatial and Temporal Databases, 11th International Symposium, SSTD 2009, Aalborg, Denmark, July 8-10, 2009, Proceedings*, volume 5644 of *Lecture Notes in Computer Science*, pages 338–354. Springer, Springer, 2009. `doi:10.1007/978-3-642-02982-0_22`.

10 Sotiris Batsakis and Euripides G. M. Petrakis. SOWL: A framework for handling spatio-temporal information in OWL 2.0. In Nick Bassiliades, Guido Governatori, and Adrian Paschke, editors, *Rule-Based Reasoning, Programming, and Applications - 5th International Symposium, RuleML 2011 - Europe, Barcelona, Spain, July 19-21, 2011. Proceedings*, volume 6826 of *Lecture Notes in Computer Science*, pages 242–249. Springer, Springer, 2011. `doi:10.1007/978-3-642-22546-8_19`.

11 Sotiris Batsakis and Euripides GM Petrakis. Representing temporal knowledge in the semantic web: The extended 4d fluents approach. In *Combinations of intelligent methods and applications*, pages 55–69. Springer, 2011. `doi:10.1007/978-3-642-19618-8_4`.

12 Elena Beisswanger, Stefan Schulz, Holger Stenzhorn, and Udo Hahn. Biotop: An upper domain ontology for the life sciencesa description of its current structure, contents and interfaces to OBO ontologies. *Appl. Ontology*, 3(4):205–212, 2008. `doi:10.3233/AO-2008-0057`.

13 Oleksandra Bruns, Tabea Tietz, Mehdi Ben Chaabane, Manuel Portz, Felix Xiong, and Harald Sack. The nuremberg address knowledge graph. In Ruben Verborgh, Anastasia Dimou, Aidan Hogan, Claudia d'Amato, Ilaria Tiddi, Arne Bröring, Simon Maier, Femke Ongenae, Riccardo Tommasini, and Mehwish Alam, editors, *The Semantic Web: ESWC 2021 Satellite Events - Virtual Event, June 6-10, 2021, Revised Selected Papers*, volume 12739 of *Lecture Notes in Computer Science*, pages 115–119. Springer, Springer, 2021. `doi:10.1007/978-3-030-80418-3_21`.

14 Oleksandra Bruns, Tabea Tietz, Sandra Göller, and Harald Sack. TRANSRAZ data model: Towards a geosocial representation of historical cities. In Maribel Acosta, Silvio Peroni, Sahar Vahdati, Anna Lisa Gentile, Tassilo Pellegrini, and Jan-Christoph Kalo, editors, *Knowledge Graphs: Semantics, Machine Learning, and Languages - Proceedings of the 19th International Conference on Semantic Systems, 20-22 September 2023, Leipzig, Germany*, volume 56 of *Studies on the Semantic Web*, pages 161–176. IOS Press, 2023. `doi:10.3233/SSW230012`.

15 Oleksandra Bruns, Tabea Tietz, Mahsa Vafaie, Danilo Dessì, Harald Sack, et al. Towards a representation of temporal data in archival records: Use cases and Requirements. In *CEUR Workshop Proc.*, volume 3019, pages 128–134. CEUR-WS, 2021. `doi:10.5445/IR/1000141526`.

16 George Bruseker, Nicola Carboni, and Anaïs Guillem. Cultural heritage data management: the role of formal ontology and cidoc crm. *Heritage and archaeology in the digital age: acquisition, curation, and dissemination of spatial cultural heritage data*, pages 93–131, 2017. `doi:10.1007/978-3-319-65370-9_6`.

17 Gavin Carothers and Lex Machina, Inc. RDF 1.1 N-Quads. A line-based syntax for RDF datasets. W3C Recommendation, 25 February 2014, 2014. Accessed: 2024-09-16. URL: `https://www.w3.org/TR/n-quads/`.

18 Valentina Anita Carriero, Aldo Gangemi, Maria Letizia Mancinelli, Ludovica Marinucci, Andrea Giovanni Nuzzolese, Valentina Presutti, and Chiara Veninata. Arco: The italian cultural heritage knowledge graph. In Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtech Svátek, Isabel F. Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon, editors, *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, volume 11779 of *Lecture Notes in Computer Science*, pages 36–52. Springer, Springer, 2019. `doi:10.1007/978-3-030-30796-7_3`.

19 Simon Cox, Chris Little, Jerry R. Hobbs, and Feng Pan. Time ontology in owl. https://www.w3.org/TR/2022/CRD-owl-time-20221115/, 2022. Accessed: 2024-09-16. `doi:10.62973/16-071r3`.

20 E. Allen Emerson and Edmund M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Sci. Comput. Program.*, 2(3):241–266, 1982. `doi:10.1016/0167-6423(83)90017-5`.

21 Flavius Frasincar, Viorel Milea, and Uzay Kaymak. towl: Integrating time in OWL. In Roberto De Virgilio, Fausto Giunchiglia, and Letizia Tanca, editors, *Semantic Web Information Management - A Model-Based Perspective*, pages 225–246. Springer, 2009. `doi:10.1007/978-3-642-04329-1_11`.

22 Fabien Gandon, Guus Schreiber, and Dave Beckett. Rdf 1.1 xml syntax. W3C Recommendation, 2014. Accessed: 2024-10-31. URL: `https://www.w3.org/TR/rdf-syntax-grammar/`.

23 Aldo Gangemi, Carola Catenacci, Massimiliano Ciaramita, and Jos Lehmann. Modelling ontology evaluation and validation. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings*, volume 4011 of *Lecture Notes in Computer Science*, pages 140–154. Springer, Springer, 2006. `doi:10.1007/11762256_13`.

24 Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with DOLCE. In Asunción Gómez-Pérez and V. Richard Benjamins, editors, *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Siguenza, Spain, October 1-4, 2002, Proceedings*, volume 2473 of *Lecture Notes in Computer Science*, pages

166–181. Springer, Springer, 2002. `doi:10.1007/3-540-45810-7_18`.

25 Stefan Gradmann. Objects, process, context in time and space: what we can do with the europeana data model (edm)-and what we cannot do. In *Brown University, Date: 2012/03/15-2012/03/15, Location: Providence*, 2012. `doi:10.1021/acsomega.7b00274.s002`.

26 Ramanathan V. Guha and Douglas B. Lenat. CYC: A midterm report. *AI Mag.*, 11(3):32–59, 1990. `doi:10.1609/aimag.v11i3.842`.

27 Giancarlo Guizzardi, Gerd Wagner, João Paulo Andrade Almeida, and Renata S. S. Guizzardi. Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story. *Appl. Ontology*, 10(3-4):259–271, 2015. `doi:10.3233/AO-150157`.

28 Claudio Gutierrez, Carlos A. Hurtado, and Alejandro A. Vaisman. Temporal RDF. In Asunción Gómez-Pérez and Jérôme Euzenat, editors, *The Semantic Web: Research and Applications, Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29 - June 1, 2005, Proceedings*, volume 3532 of *Lecture Notes in Computer Science*, pages 93–107. Springer, Springer, 2005. `doi:10.1007/11431053_7`.

29 Claudio Gutierrez, Carlos A. Hurtado, and Alejandro A. Vaisman. Introducing time into RDF. *IEEE Trans. Knowl. Data Eng.*, 19(2):207–218, 2007. `doi:10.1109/TKDE.2007.34`.

30 Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Ana Ozaki. On metric temporal description logics. In Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen, editors, *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 837–845. IOS Press, 2016. `doi:10.3233/978-1-61499-672-9-837`.

31 Joseph Y. Halpern and Yoav Shoham. A propositional modal logic of time intervals. *J. ACM*, 38(4):935–962, 1991. `doi:10.1145/115234.115351`.

32 Olaf Hartig. Foundations of rdf⋆ and sparql⋆ (an alternative approach to statement-level metadata in RDF). In Juan L. Reutter and Divesh Srivastava, editors, *Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, Montevideo, Uruguay, June 7-9, 2017*, volume 1912 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017. URL: `https://ceur-ws.org/Vol-1912/paper12.pdf`.

33 Sally Haslanger. Persistence through time. *The Oxford Handbook of Metaphysics*, pages 315–354, 2003. `doi:10.1093/oxfordhb/9780199284221.003.0012`.

34 Muhammad Rosyihan Hendrawan, Azman Mat Isa, and Ahmad Zam Hariro Samsudin. Evolving taxonomies: Fifty years of taxonomy development in library and information science. *Journal of Librarianship and Information Science*, page 09610006251334372, 2025. `doi:10.1177/09610006251334372`.

35 Heinrich Herre. *General Formal Ontology (GFO): A Foundational Ontology for Conceptual Modelling*, pages 297–345. Springer Netherlands, Dordrecht, 2010. `doi:10.1007/978-90-481-8847-5_14`.

36 Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Comput. Surv.*, 54(4):71:1–71:37, 2022. `doi:10.1145/3447772`.

37 Carlos A. Hurtado and Alejandro A. Vaisman. Reasoning with temporal constraints in RDF. In José Júlio Alferes, James Bailey, Wolfgang May, and Uta Schwertel, editors, *Principles and Practice of Semantic Web Reasoning, 4th International Workshop, PPSWR 2006, Budva, Montenegro, June 10-11, 2006, Revised Selected Papers*, volume 4187 of *Lecture Notes in Computer Science*, pages 164–178. Springer, Springer, 2006. `doi:10.1007/11853107_12`.

38 Eero Hyvönen. Digital humanities on the semantic web: Sampo model and portal series. *Semantic Web*, 14(4):729–744, 2023. `doi:10.3233/SW-223034`.

39 Michael Kifer and V. S. Subrahmanian. Theory of generalized annotated logic programming and its applications. *J. Log. Program.*, 12(3&4):335–367, 1992. `doi:10.1016/0743-1066(92)90007-P`.

40 Mikko Koho, Esko Ikkala, Petri Leskinen, Minna Tamper, Jouni Tuominen, and Eero Hyvönen. Warsampo knowledge graph: Finland in the second world war as linked open data. *Semantic Web*, 12(2):265–278, 2021. `doi:10.3233/SW-200392`.

41 Abba Lawan and Abdur Rakib. FT-SWRL: A fuzzy-temporal extension of semantic web rule language. *CoRR*, abs/1911.12399, 2019. `doi:10.48550/arXiv.1911.12399`.

42 Pasquale Lisena, Daniel Schwabe, Marieke van Erp, Raphaël Troncy, William Tullett, Inger Leemans, Lizzie Marx, and Sofia Colette Ehrich. Capturing the semantics of smell: The odeuropa data model for olfactory heritage information. In Paul Groth, Maria-Esther Vidal, Fabian M. Suchanek, Pedro A. Szekely, Pavan Kapanipathi, Catia Pesquita, Hala Skaf-Molli, and Minna Tamper, editors, *The Semantic Web - 19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, May 29 - June 2, 2022, Proceedings*, volume 13261 of *Lecture Notes in Computer Science*, pages 387–405. Springer, Springer, 2022. `doi:10.1007/978-3-031-06981-9_23`.

43 Nuno Lopes, Axel Polleres, Umberto Straccia, and Antoine Zimmermann. Anql: Sparqling up annotated RDFS. In Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Z. Pan, Ian Horrocks, and Birte Glimm, editors, *The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part I*, volume 6496 of *Lecture Notes in Computer Science*, pages 518–533. Springer, Springer, 2010. `doi:10.1007/978-3-642-17746-0_33`.

44 J Ellis McTaggart. The unreality of time. *Mind*, pages 457–474, 1908. `doi:10.1093/mind/xvii.4.457`.

45 Vinh Nguyen, Olivier Bodenreider, and Amit P. Sheth. Don't like RDF reification?: making statements about statements using singleton property. In Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel, editors, *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 759–770. ACM, 2014. `doi:10.1145/2566486.2567973`.

46 Vinh Nguyen, Olivier Bodenreider, Krishnaprasad Thirunarayan, Gang Fu, Evan Bolton, Núria Queralt Rosinach, Laura Inés Furlong, Michel Dumontier, and Amit P. Sheth. On reasoning with RDF statements about statements using singleton property triples. *CoRR*, abs/1509.04513, 2015. `doi:10.48550/arXiv.1509.04513`.

47 Ian Niles and Adam Pease. Towards a standard upper ontology. In *2nd International Conference on Formal Ontology in Information Systems, FOIS 2001, Ogunquit, Maine, USA, October 17-19, 2001, Proceedings*, pages 2–9. ACM, 2001. `doi:10.1145/505168.505170`.

48 Natasha Noy, Alan Rector, Pat Hayes, and Chris Welty. Defining n-ary relations on the semantic web. *W3C working group note*, 12(4), 2006. Accessed: 2024-10-31. URL: `https://www.w3.org/TR/swbp-n-aryRelations/`.

49 Martin J. O'Connor and Amar K. Das. SQWRL: A query language for OWL. In Rinke Hoekstra and Peter F. Patel-Schneider, editors, *Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2009), Chantilly, VA, United States, October 23-24, 2009*, volume 529 of *CEUR Workshop Proceedings*, pages 1–8. CEUR-WS.org, 2009. URL: `https://ceur-ws.org/Vol-529/owled2009_submission_42.pdf`.

50 Martin J. O'Connor and Amar K. Das. A method for representing and querying temporal information in OWL. In Ana L. N. Fred, Joaquim Filipe, and Hugo Gamboa, editors, *Biomedical Engineering Systems and Technologies - Third International Joint Conference, BIOSTEC 2010, Valencia, Spain, January 20-23, 2010, Revised Selected Papers*, volume 127 of *Communications in Computer and Information Science*, pages 97–110. Springer, Springer, 2010. `doi:10.1007/978-3-642-18472-7_8`.

51 J. Neil Otte, John Beverley, and Alan Ruttenberg. BFO: basic formal ontology. *Appl. Ontology*, 17(1):17–43, 2022. `doi:10.3233/AO-220262`.

52 Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 46–57. IEEE, IEEE Computer Society, 1977. `doi:10.1109/SFCS.1977.32`.

53 Axel Polleres, Romana Pernisch, Angela Bonifati, Daniele Dell'Aglio, Daniil Dobriy, Stefania Dumbrava, Lorena Etcheverry, Nicolas Ferranti, Katja Hose, Ernesto Jiménez-Ruiz, Matteo Lissandrini, Ansgar Scherp, Riccardo Tommasini, and Johannes Wachs. How does knowledge evolve in open knowledge graphs? *TGDK*, 1(1):11:1–11:59, 2023. `doi:10.4230/TGDK.1.1.11`.

54 Andrea Pugliese, Octavian Udrea, and V.S. Subrahmanian. Scaling RDF with time. In Jinpeng Huai, Robin Chen, Hsiao-Wuen Hon, Yunhao Liu, Wei-Ying Ma, Andrew Tomkins, and Xiaodong Zhang, editors, *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, pages 605–614. ACM, 2008. `doi:10.1145/1367497.1367579`.

55 Katerina El Raheb, Theofilos Mailis, Vladislav Ryzhikov, Nicolas Papapetrou, and Yannis E. Ioannidis. Balonse: Temporal aspects of dance movement and its ontological representation. In Eva Blomqvist, Diana Maynard, Aldo Gangemi, Rinke Hoekstra, Pascal Hitzler, and Olaf Hartig, editors, *The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part II*, volume 10250 of *Lecture Notes in Computer Science*, pages 49–64. Springer, 2017. `doi:10.1007/978-3-319-58451-5_4`.

56 Yves Raimond and Samer Abdallah. The timeline ontology. `http://motools.sourceforge.net/timeline/timeline.html`, 2007. Accessed: 2024-09-16.

57 Anisa Rula, Matteo Palmonari, Andreas Harth, Steffen Stadtmüller, and Andrea Maurino. On the diversity and availability of temporal information in linked open data. In Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I*, volume 7649 of *Lecture Notes in Computer Science*, pages 492–507. Springer, Springer, 2012. `doi:10.1007/978-3-642-35176-1_31`.

58 Albrecht Schmiedel. Temporal terminological logic. In Howard E. Shrobe, Thomas G. Dietterich, and William R. Swartout, editors, *Proceedings of the 8th National Conference on Artificial Intelligence. Boston, Massachusetts, USA, July 29 - August 3, 1990, 2 Volumes*, pages 640–645. AAAI Press/The MIT Press, AAAI Press / The MIT Press, 1990. URL: `http://www.aaai.org/Library/AAAI/1990/aaai90-096.php`.

59 Bernhard Schueler, Sergej Sizov, Steffen Staab, and Duc Thanh Tran. Querying for meta knowledge. In Jinpeng Huai, Robin Chen, Hsiao-Wuen Hon, Yunhao Liu, Wei-Ying Ma, Andrew Tomkins, and Xiaodong Zhang, editors, *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, pages 625–634. ACM, 2008. `doi:10.1145/1367497.1367582`.

60 Sangeeta Sen, Mariana Curado Malta, Biswanath Dutta, and Animesh Dutta. State-of-the-art approaches for meta-knowledge assertion in the web of data. *IETE Technical Review*, 38(6):672–709, 2021. `doi:10.1080/02564602.2020.1819891`.

61 Claiurton A. Siebra and Katarzyna Wac. Engineering uncertain time for its practical integration in ontologies. *Knowl. Based Syst.*, 251:109152, 2022. `doi:10.1016/j.knosys.2022.109152`.

62 Jannik Strötgen, Thomas Bögel, Julian Zell, Ayser Armiti, Tran Van Canh, and Michael Gertz. Extending heideltime for temporal expressions referring to historic dates. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 2390–2397. European Language Resources Association (ELRA), 2014. URL: `http://www.lrec-conf.org/proceedings/lrec2014/summaries/849.html`.

63 Jannik Strötgen and Michael Gertz. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In Katrin Erk and Carlo Strapparava, editors, *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010*, pages 321–324. The Association for Computer Linguistics, 2010. URL: `https://aclanthology.org/S10-1071/`.

64 Ivan Terziev, Atanas Kiryakov, and Dimitar Manov. D. 1.8. 1 base upper-level ontology (bulo) guidance. *Deliverable of EU-IST Project IST*, 2005.

65 Tabea Tietz, Mehwish Alam, Harald Sack, and Marieke van Erp. Challenges of knowledge graph evolution from an NLP perspective. In Alessandro Adamou, Enrico Daga, and Albert Meroño-Peñuela, editors, *Proceedings of the Third Workshop on Humanities in the Semantic Web (WHiSe 2020) co-located with 15th Extended Semantic Web Conference (ESWC 2020), Heraklion, Greece, June 2, 2020 (online)*, volume 2695 of *CEUR Workshop Proceedings*, pages 71–76. CEUR-WS.org, 2020. URL: `https://ceur-ws.org/Vol-2695/paper8.pdf`.

66 Tabea Tietz, Oleksandra Bruns, Sandra Göller, Matthias Razum, Danilo Dessì, and Harald Sack. Knowledge graph enabled curation and exploration of nuremberg's city heritage. In Adrian Paschke, Georg Rehm, Jamal Al Qundus, Clemens Neudecker, and Lydia Pintscher, editors, *Proceedings of the Conference on Digital Curation Technologies (Qurator 2021), Berlin, Germany, February 8th - to - 12th, 2021*, volume 2836 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2021. `doi:10.5445/IR/1000131502`.

67 Tabea Tietz, Oleksandra Bruns, and Harald Sack. A data model for linked stage graph and the historical performing arts domain (short paper). In Antonis Bikakis, Roberta Ferrario, Stéphane Jean, Béatrice Markhoff, Alessandro Mosca, and Marianna Nicolosi Asmundo, editors, *Proceedings of the International Workshop on Semantic Web and Ontology Design for Cultural Heritage co-located with the International Semantic Web Conference 2023 (ISWC 2023), Athens, Greece, November 7, 2023*, volume 3540 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2023. `doi:10.5445/IR/1000169249`.

68 Tabea Tietz, Jörg Waitelonis, Kanran Zhou, Paul Felgentreff, Nils Meyer, Andreas Weber, and Harald Sack. Linked stage graph. In Mehwish Alam, Ricardo Usbeck, Tassilo Pellegrini, Harald Sack, and York Sure-Vetter, editors, *Proceedings of the Posters and Demo Track of the 15th International Conference on Semantic Systems co-located with 15th International Conference on Semantic Systems (SEMANTiCS 2019), Karlsruhe, Germany, September 9th - to - 12th, 2019*, volume 2451 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019. `doi:10.5445/IR/1000100183`.

69 Octavian Udrea, Diego Reforgiato Recupero, and V. S. Subrahmanian. Annotated RDF. *ACM Trans. Comput. Log.*, 11(2):10:1–10:41, 2010. `doi:10.1145/1656242.1656245`.

70 Mahsa Vafaie, Oleksandra Bruns, Danilo Dessì, Nastasja Pilz, and Harald Sack. Modelling archival hierarchies in practice: Key aspects and lessons learned. In Yasunobu Sumikawa, Ryohei Ikejiri, Antoine Doucet, Eva Pfanzelter, Mohammed Hasanuzzaman, Gaël Dias, Ian Milligan, and Adam Jatowt, editors, *Proceedings of the 6th International Workshop on Computational History (HistoInformatics 2021) co-located with ACM/IEEE Joint Conference on Digital Libraries 2021 (JCDL 2021), Online event, September 30-October 1, 2021*, volume 2981 of *CEUR Workshop Proceedings*, page 6. Aachen, Germany: RWTH Aachen, CEUR-WS.org, 2021. `doi:10.5445/IR/1000139652`.

71 Oleksandra Vsesviatska, Tabea Tietz, Fabian Hoppe, Mirjam Sprau, Nils Meyer, Danilo Dessì, and Harald Sack. Ardo: an ontology to describe the dynamics of multimedia archival records. In Chih-Cheng Hung, Jiman Hong, Alessio Bechini, and Eunjee Song, editors, *SAC '21: The 36th ACM/SIGAPP Symposium on Applied Computing, Virtual Event, Republic of Korea, March 22-26, 2021*, pages 1855–1863. ACM, 2021. `doi:10.1145/3412841.3442057`.

72 Hsien-Tseng Wang. *Valid Time RDF*. PhD thesis, City University of New York, USA, 2020. URL: `https://academicworks.cuny.edu/gc_etds/4037`.

73 Hsien-Tseng Wang and Abdullah Uz Tansel. Temporal extensions to RDF. *J. Web Eng.*, 18(1-3):125–168, 2019. `doi:10.13052/jwe1540-9589.18134`.

74 Robert Weida and Diane Litman. Subsumption and recognition of heterogeneous constraint networks. In *Proceedings of the Tenth Conference on Artificial Intelligence for Applications*, pages 381–388. IEEE, 1994. `doi:10.1109/caia.1994.323650`.

75 Christopher A. Welty and Richard Fikes. A reusable ontology for fluents in OWL. In Brandon Bennett and Christiane Fellbaum, editors, *Formal Ontology in Information Systems, Proceedings of the Fourth International Conference, FOIS 2006, Baltimore, Maryland, USA, November 9-11, 2006*, volume 150 of *Frontiers in Artificial Intelligence and Applications*, pages 226–236. ACM / IOS Press, 2006. `doi:10.5555/1566079.1566106`.

76 Di Wu, Hsien-Tseng Wang, and Abdullah Uz Tansel. A survey for managing temporal data in RDF. *Inf. Syst.*, 122:102368, 2024. `doi:10.1016/j.is.2024.102368`.

77 Fouad Zablith, Grigoris Antoniou, Mathieu d'Aquin, Giorgos Flouris, Haridimos Kondylakis, Enrico Motta, Dimitris Plexousakis, and Marta Sabou. Ontology evolution: a process-centric survey. *Knowl. Eng. Rev.*, 30(1):45–75, 2015. `doi:10.1017/S0269888913000349`.

78 Fu Zhang, Zhiyin Li, Dunhong Peng, and Jingwei Cheng. RDF for temporal data management - a survey. *Earth Sci. Informatics*, 14(2):563–599, 2021. `doi:10.1007/s12145-021-00574-w`.