Engineering uncertain time for its practical integration in ontologies

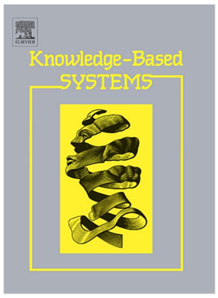
De Albuquerque Siebra, Clauirton; Wac, Katarzyna

How to cite

DE ALBUQUERQUE SIEBRA, Clauirton, WAC, Katarzyna. Engineering uncertain time for its practical integration in ontologies. In: Knowledge-based systems, 2022, p. 109152. doi: 10.1016/j.knosys.2022.109152

This publication URL: https://archive-ouverte.unige.ch/unige:161412
Publication DOI: 10.1016/j.knosys.2022.109152

Journal Pre-proof



Engineering uncertain time for its practical integration in ontologies

Clauirton A. Siebra, Katarzyna Wac

PII: S0950-7051(22)00572-X
DOI: https://doi.org/10.1016/j.knosys.2022.109152
Reference: KNOSYS 109152

To appear in: *Knowledge-Based Systems*

Received date : 2 November 2021
Revised date : 23 May 2022
Accepted date : 24 May 2022

Please cite this article as: C.A. Siebra and K. Wac, Engineering uncertain time for its practical integration in ontologies, *Knowledge-Based Systems* (2022), doi: https://doi.org/10.1016/j.knosys.2022.109152.

Engineering Uncertain Time for its Practical Integration in Ontologies

Clauirton A. Siebra[a,b*], and Katarzyna Wac[a]

[a]Quality of Life Lab, University of Geneva, CH-1227 Switzerland
[b]Applied Artificial Intelligence Lab, Federal University of Paraiba – Joao Pessoa, Brazil

Abstract: Ontologies are commonly used as a strategy for knowledge representation. However, they are still presenting limitations to model domains that require broad forms of temporal reasoning. This study is part of the Onto-mQoL project and was motivated by the real need to extend static ontologies with diverse time concepts, relations and properties, which go beyond the commonly used Allen´s Interval Algebra. Therefore, we use the n-ary relations as the basis for temporal structures, which minimally modify the original ontology, and extend these structures with a generic set of time concepts (moments and intervals), time concept properties (precise and uncertain), time relations (interval-interval, interval-moment, and moment-moment), and time relation properties (qualitative and quantitative). We divided the scientific contribution of this study into three parts. Firstly, we present the ontological temporal model (classes and properties) and how it is integrated into static ontologies. Secondly, we discuss the creation of axioms that give the semantics for precise temporal elements. Finally, as our main contribution, these ideas are extended with axioms for uncertain time. All these elements follow the Ontology Web Language (OWL) standards, so this proposal is still compatible with the main ontology editors and reasoners currently available. A case example demonstrates the use of this approach in the nutrition assessment domain.

Keywords: Ontology design, Rule-based processing, Uncertainty, Temporal Logic

1. Introduction

Ontologies are broadly used to represent knowledge due to their advantages such as consistency, reuse, and easy extensibility. Apart from their advantages, ontologies also present limitations. While temporal aspects are usually present in several domains, modelling information evolving time/events in ontologies is a complex problem because temporal relations are ternary and cannot be directly handled by ontology languages. This problem happens because ontologies use the Description Logics (DL) [1] as their formal basis, which is a decidable fragment of the first-order logic that only uses unary (concepts) and binary (roles) predicates. Therefore, ontologies do not naturally support representations for inputs such as “I slept well from 23:00 to

04:00", "I had a posttraumatic stress disorder after my car accident", and "I had a heart attack in 1999"; while the current temporal extensions that are still compatible with the current ontology resources (e.g., reasoners and DL formalism) are limited in aspects related to uncertain time, the evolution of events, and relations between time concepts [2]. This lack of expressive and standards-compatible temporal representations restricts the range of reasoning processes because several time aspects, which are usually embedded in knowledge domains (e.g., health and education) [3], cannot be employed. Moreover, the integration of uncertain time is particularly an important extension since recent studies indicate the inexistence of approaches devoted to handling uncertain temporal data in ontologies [4].

Our study (Onto-mQoL) aims to design a conceptual framework for representation and reasoning about temporal aspects of knowledge, associated with an ontological engineering method to guide the use of this novel framework over the process of converting static ontologies to their temporal versions. Thus, this study will advance the state of the art as follows:

- The majority of the works consider reasoning processes between qualitative time intervals. The present approach will extend this notion to a broad perspective, considering different time concepts (moments and intervals), time concept properties (precise and uncertain), time relations (interval-interval, interval-moment, and moment-moment), and time relation properties (qualitative and quantitative);
- Recent approaches for temporal reasoning employ theories (e.g., Fuzzy Theory Sets and Temporal DLs) that are not compatible with the current Semantic Web standards (e.g., DL formalism and reasoners). The maintenance of this compatibility is an important premise and contribution of this present project;
- The associated engineering method is an innovation proposed in this project since other approaches do not present pragmatic strategies to apply their contributions. This method will indicate, for example, how developers can define a set of DL-based rules to infer new axioms and augment the ontology expressiveness for their domains.

As an example of use and engineering method, we apply this framework to extend a static ontology on nutritional habits, which was based on the analysis of mobile applications and academic studies in this health area. Therefore, the process of extension is detailed, together with examples of queries that we can now answer after this temporal extension. The evaluation in this example is characterised as empiric and based on the ontology's ability to support the generation of answers for competency questions, which are usually specified during the stage of design and represent a list of typical questions that knowledge representations are supposed to offer answers [5].

The remainder of this paper is organised as follows: Section 2 presents a motivational scenario, which emphasises the temporal requirements for knowledge representations. Section 3 summarises the related works, comparing their main features and limitations. Section 4 presents the conceptual framework and how it implements the requirements also specified in such a section. Section 5 addresses the forms of evaluations and their results. Section 6 presents the case example based on the nutritional domain, mainly demonstrating the engineering to transform a static ontology into its temporal-based version. Finally, Section 7 concludes this work with its main remarks and research directions.

2. Motivation

Two main pragmatic aspects motivate this study. Firstly, there are important developments in new pervasive sensors, which can assess a huge and diverse amount of individual's physiological state (e.g., arousal), physical (e.g., sleep), psychological (e.g., affective state), and social functioning (e.g., social relationship) as well as individual's context (e.g., location) data. Thus, ontologies are important to organise and relate these data. Moreover, the assessment of these states and behaviours and their correlation with the individual's health and life quality is mostly continuous and longitudinal, implying several temporal dependencies that must be accurately represented in the data and further modelled. The following sections discuss these two aspects.

2.1 Sensors and Data

The current widespread availability of personalized and miniaturized technological innovations, including mobile devices and applications, referred to as IoT (internet of things) [6, 7], may enable the continuous assessment of daily life behaviours. In our past research, we have identified and analyzed 438 such devices passively assessing human states and behaviours ranging from physical activity via sleep to less common likes - playing golf or baseball [8]. Many efforts have been put into the devices' personalisation, miniaturisation, efficiency and effectiveness over the last years [9, 10].

It has been proven by now that especially repetitive harmful behaviours like smoking, alcohol intake, lack of physical activity, and malnutrition are four top contributors to individual health and life quality in the long term [11]. Additionally, repetitive behaviours like (lack of) physical activity or sleep may result from the individual's health state. Overall, the continuous, accurate, and timely assessment of these different behaviours may facilitate an enhanced understanding of an individual's short-term and long-term health and QoL [12]. Given the current state of the art, it is of utmost importance to model these behaviours, i.e., their timing, duration, frequency and intensity, and standardize the data representation for better knowledge in health maintenance, disease prevention, treatment or rehabilitation purposes [13].

2.2 Time Requirements

Temporal phenomena generate different types of requirements on extensions for ontologies. These extensions, for example, are important for health-related ontologies because time aspects are inherently embedded in the information they describe. Thus, inferences and their refinements cannot be managed without using broad and expressive temporal representations (e.g., a diagnostic criterion for some health issues may be the persistence of some symptoms for a given time). Consider, for example, the following definition: "*insomnia is characterised by complaints about the duration and quality of sleep, difficulty falling asleep, nocturnal awakenings, early awakening, or nonrecuperative sleep. This symptomatology must be present at least three times a week for at least one month, with negative consequences the next day.*". The following reports represent possible answers of two different patients with sleep disorders in their medical appointments:

- I sleep five hours per night [*precise interval*]. I go to bed at 22:00 [*precise moment*] but stay awake for two hours before sleep [*precise quantitative relation between interval and moment*]. I wake up three times during the night [*precise quantitative relation between moments and interval*] and wake up in the morning at 5:00 [*precise moment*]. I always have a headache after waking up [*qualitative relation between moments*]. These bad nights happened every day during the last month [*precise qualitative relation between intervals*].
- I sleep around five hours per night [*uncertain interval*]. I go to bed at early 22:00 [uncertain *moment*] but stay awake for two hours before sleep [*uncertain quantitative relation between interval and moment*]. I wake up three times during the night [*precise qualitative relation between moments and interval*] and wake up in the morning around 5:00 [uncertain *moment*]. I always have a headache that starts around the waking up time [*uncertain qualitative relation between moments*]. I think these bad

nights happened every day during the last month [*uncertain qualitative relation between intervals*].

The first case represents the answer of a patient who is precise in her answers. Thus, the doctor can easily return a diagnosis based on the insomnia definition. However, such a type of answer is unlikely to happen. The second case represents a more common answer, which demonstrates levels of uncertainty about the start or end of events and, consequently, their temporal relations. Knowledge representations, such as ontologies, must express these precise and uncertain facts. Otherwise, they will not be able to support the diagnosis (first case) or the diagnosis probability (second case) of insomnia.

Time requirements become more apparent in representations involving the evolution of concepts, such as bio-health ontologies that describe the development of any biological entity. According to [14], statements made in this type of ontology could include elements developing from others (e.g., embryonic cells developing from other cells), any entity taking place during a specific stage (e.g., organ development taking place in certain trimesters), or an event occurring before, after or during another event (e.g., gestational ages occurring during specific weeks). Therefore, if precise and uncertain time information is needed but cannot be represented for this and several other domains (e.g., representation of physical and chemical phenomena, engineering processes, or academic evolution of students), then knowledge about such domains will be misrepresented, restricting the reasoning process and queries that relate timeline events.

3. Temporal Representations in Ontologies

This section summarises the past approaches that aim at including temporal representations into ontologies. Our analysis considers the conceptual rather than chronological perspective of these studies. The last part of this section compares these approaches regarding their main ideas and limitations.

3.1 Domain-Oriented Approaches

The literature discusses several proposals to use ontologies as resources to represent and reason about temporal aspects of knowledge. For example, Moens and Steedman [15] proposed an ontology for temporal descriptions that relies on the semantics of linguistic categories, such as tense and temporal adverbials, used to define temporal relations in the natural language domain. The work of Artale et al. [16] proposed theoretical extensions for ontology query languages based on temporal logical operators that relate time moments. Thus, rather than an approach for temporal ontology specifications, the authors intend to augment the expressiveness of temporal queries. Pedrinaci et al. [17] presented an ontology for Business Process Analysis (BPA) built upon a time ontology. Thus, it is structured around business elements, such as processes and resources, providing a core terminology for supporting BPA. While these studies show the importance of time representations in ontologies, their approaches consider the particularities of specific domains (e.g., natural language, queries language, BPA). Thus, they are not generic frameworks aimed at extending static ontologies, such as an electronic health record (EHR) ontology, to their temporal versions. Moreover, their approaches focus only on representations of qualitative relations between precise time notions.

3.2 OWL-Time

Differently from the previous approaches, henceforward our focus is on general frameworks that can modify static ontologies, creating their temporal versions. Therefore, OWL-Time [18] is the current standard option to provide a vocabulary for expressing time-related aspects. This ontology provides temporal concepts for facts about topological (ordering) relations among moments and intervals. Moreover, it also includes information about durations and temporal position, such as date-time information. However, apart from the language constructs for the representation of time in ontologies, this standard does not contain mechanisms to represent the evolution of concepts (e.g., events) in time. Moreover, OWL-Time does not propose inference rules to automatically infer new temporal data [4].

3.3 DL Extensions for Precise Time

A strategy to include temporal aspects in ontologies, and ensure support for inference processes, is extending the DL formalism with special operators. For example, Temporal Description Logics (TDLs) [19] offer additional expressive capabilities over non-temporal DLs. The new temporal operators are the temporal qualifier *at*, which specifies the time at which a concept holds, and the existential and universal temporal quantifiers *sometime* and *alltime*. Moreover, the proposal is extended with metric constraints to cover durations, absolute times, and granularities of intervals. Although this approach has the advantage of elegant and well-understood semantics, the existing inference engines (reasoners) and edition tools are not compatible with such DLs extensions. Other similar approaches in this category, which present the same limitations, are Concrete domains [20] and Temporal RDF [21].

3.4 DL Extensions for Precise and Uncertain Time

Approaches that consider uncertain time mostly rely on fuzzy set principles to define temporal relations between uncertain time intervals. The work of Nagypál and Motik [22] follows this direction, introducing a set of auxiliary operators on time intervals and defining fuzzy counterparts of these operators. However, they do not demonstrate the compatibility of these new Fuzzy temporal relations properties. Moreover, their work does not show how uncertain temporal relations are specified. This compatibility problem is also present in similar approaches, such as [23], which require DL extensions.

3.5 DL Implementations for Precise Time

Differently from the approaches in Sections 3.3 and 3.4, some studies avoid introducing additional constructs to the standard language and tools. Thus, they ensure DL compatibility with reasoning engines. Main approaches in this direction are the 4D-Fluents [24] and the N-ary relations [25], which embrace time intervals and associated qualitative relations using Allen´s Interval Algebra [26] as the basis of their reasoning. This algebra captures the 13 possible qualitative relations (before, after, meets, met by, overlaps, overlapped by, starts, started by, during, contains, finishes, finished by, and equals) between two precise intervals. However, they do not offer proper support for managing moments (time points) and qualitative relations between a time interval and a moment, or two moments [4]. Moreover, uncertain time is not part of this algebra.

3.6 DL Implementations for Precise and Uncertain Time

Ghorbel et al. [27] affirm there is no DL compatible approach devoted to handling uncertain temporal data in ontologies. Their study (TimeOnto) tried to cover this lack using a fuzzy-based strategy for representing and reasoning about uncertain time intervals. Differently from the approaches in Section 3.4, TimeOnto does not create new operators but creates temporal object properties that relate ontological concepts. For example, the Allen's relation *Before* may be generalised in three uncertain relations, where $Before_{(1)}$ means *approximately the same time*; $Before_{(2)}$ means *just before* and $Before_{(3)}$ means *long before*. Thus, this approach is focused on the qualification of uncertainty about the temporal distance of events, instead of the uncertainty about the existence of temporal relations. A recent work of the same group [28] proposed an approach for this latter type of uncertainty that relies on probability values for temporal uncertain elements (e.g., beginning and end time point of intervals). However, the definition of these probabilities is not an easy task in some scenarios. After obtaining these probabilities, simple Bayesian networks are used to derive levels of uncertainty for relations.

Thus, these relations do not consider the relative distance between time instances (example in Section 5.3) as part of their semantics. Apart from that, both approaches [27,28] rely on the 4D-Fluents model, which increases the complexity of the ontology because each non-temporal entity must have its related temporal class.

3.7 Comparison of Approaches

Table 1 compares the approaches categories (Sections 3.1 to 3.6) regarding their main ideas and limitations. As one of the results of these limitations, the literature does not present examples of real case applications of these approaches, but only restricted studies based on temporal language queries, such as TOQL and SQWRL, as demonstrated by Bahadorani and Zaeri [29]. Section 4 lists the requirements of our approach aimed at covering these limitations.

Table 1. Comparison among state-of-the-art approaches.

Approach	Main feature	Main limitation
[15,16,17]	Based on features of specific domains	Hard to use in other domains
[18]	Current standard for time representation	Cannot represent properties of objects changing in time
[19,20,21]	Strong background formalism and expressiveness	Standard languages and tools are no longer appropriate
[22,23]	Rely on the Fuzzy Theory Set formalism	Standard languages and tools are no longer appropriate
[24,25]	Follow the principles of DL formalism	Data redundance and focus on precise time
[27]	Covers both precise and uncertain temporal relations	Does not cover uncertainty on existence of temporal relations
[28]	Covers uncertainty on existence of temporal relations	Uncertainty semantics based only on time point probabilities
This study	Well-defined semantics for uncertain relations	Semantics does not cover some uncertain terms (Section 6.5)

4. Conceptual Framework

The aspects discussed in the previous section limit the use of ontologies in more complex domains (e.g., health scenarios). For example, the case presented in Section 2 requires notions of uncertain intervals and moments, which are not currently covered by ontologies. Therefore, this section first lists the requirements that our proposal intends to consider. After that, the following sections details how we implemented each of these requirements. All equations defined in this section were mapped to rules using the Semantic Query-Enhanced Web Rule Language (SQWRL) and tested using the Protégé framework.

4.1 Requirements

Our work considers the following requirements to approach the limitations discussed in Section 3:

- R1: precise and uncertain time notions (interval and moments) must be classes rather than data properties. This requirement allows relating time elements (e.g., event-a *before* event-b) using object properties, and also associate data properties to these elements (e.g., time unit and zone);

- R2: the semantics of time elements must represent uncertain time-related terms, such as "about" (e.g., my stomach-ache started about mid-day) and "by" (e.g., Her Alzheimer´s symptoms started by 2009");
- R3: support for specifying precise qualitative and quantitative temporal relations between any two precise time notions (moment-moment, interval-interval, moment-interval, and interval-moment);
- R4: automatic inference of temporal relations between events, regardless of the type of time notion (moment or interval) when these events occur;
- R5: support for specifying uncertain qualitative and quantitative temporal relations between any two uncertain time notions;
- R6: support for a flexible definition of levels of uncertainty for relations. For example, the relation *before* can have 3 (e.g., almost surely, strong surely, weak surely etc.) or more uncertainty levels;
- R7: the semantics of uncertain relations must also consider the mix of precise and uncertain time notions.

These requirements were mostly implemented in an incremental way since the implementation of some requirements create the basis for other requirements. The following schema illustrates the overall picture of this process.

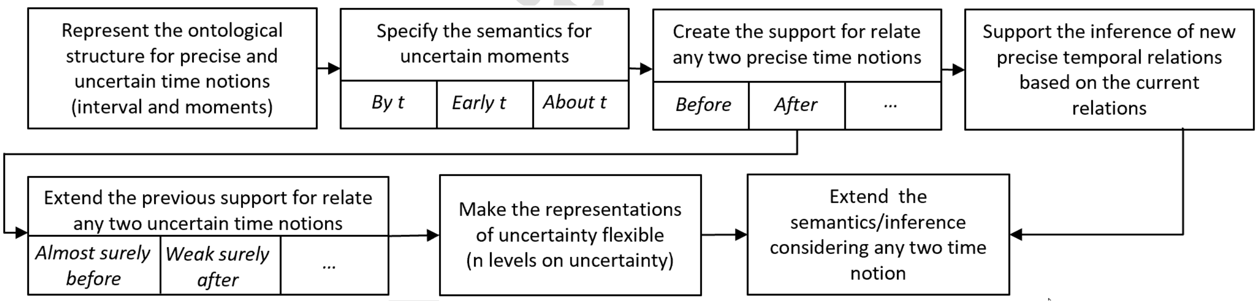


Fig. 1. Implementation flow of the requirements.

4.2 Designing Time elements as Classes (R1)

Our work considers the N-ary relations (Fig. 2) [25] as the basis for the ontological structure. The N-ary approach implies minimal data redundancy when compared with other methods (e.g., 4D-Fluents). However, this redundancy still exists regarding, for example, the inverse and symmetric properties. Temporal elements are now considered as explicit events (e.g., *EatingEvent*) and represented as classes rather than data properties. Meanwhiles, individuals of such classes correspond to individuals of the relation. For example, the *takes* relation is no longer a relation having as object an individual of the class *Meal* and a subject of the class *Patient* as they are now related to the new class *EatingEvent*. Then, all events have a time interval, with start and end moments, and such events happen during this interval. This structure is partially in accordance with the requirement R1 since the temporal elements are explicitly represented as classes rather than data properties. However, we need to extend this structure so it could support the concept of uncertain time and also be generic for both intervals and moments.

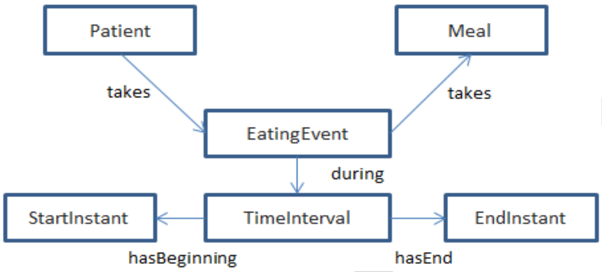


Fig. 2. Example of the N-ary relations approach

Therefore, in our novel approach (Fig. 3), a generic event is related (*occurs* object property) to a temporal notion (*TimeNotion* class), which can be an interval or moment (subclasses of *TimeNotion*). Intervals are associated with two moments using the *hasBeginning* and *hasEnd* object properties, while any moment used in this representation has a precise or uncertain value.

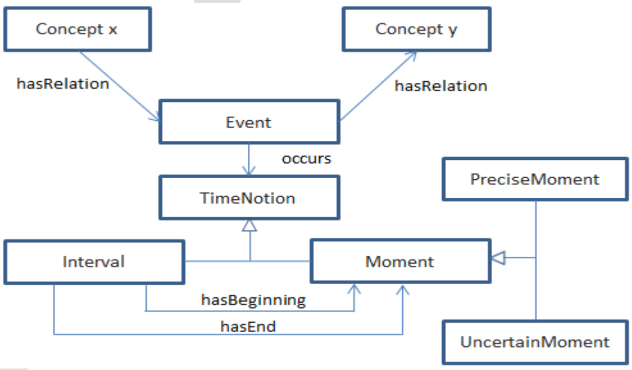


Fig. 3. Extensions of the N-ary relations.

4.3 Semantics for Uncertain Time Concepts (R2)

OWL-Time is the default representation of time in ontologies and it already has concepts (e.g., hour, daytime, and days of the week) to represent precise moments. However, it does not represent uncertain time-related terms as described in the requirement R2. Consider now some examples of uncertain moments defined by the following sentences that have 20:00 as their time reference μ:

- My dog ate by 20:00 - this moment is uncertain in the sense that it could mean, for example, values from 17:00 to 20:00 with an increasing probability;
- My dog ate early 20:00 - this moment is uncertain in the sense that it could mean, for example, values from 20:00 to 23:00 with a decreasing probability;
- My dog ate about 20:00 – this moment is uncertain in the sense that it considers the two previous situations together. In other words, for example, values from 17:00 to 23:00.

The terms "increasing" and "decreasing" indicate that the moments have associated probability values. Our approach represents these values using the membership functions illustrated in the following graphs (Fig. 4), where UR_b and UR_e respectively mean the beginning and end of an uncertain temporal range.

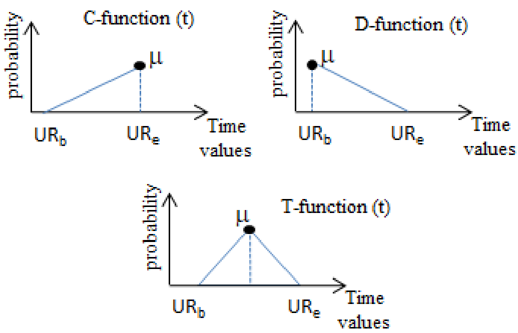


Fig. 4. Graphical representation of membership functions, which indicate the probability of a moment occurring.

Membership functions provide the basis to represent any uncertain moment using three properties: uncertain function type ('C'rescente, 'D'ecrescent, and 'T'riangular), referential moment (μ), and uncertain range (ϕ). Therefore, the following relations (1) – (3) are given for each membership function:

$$\textit{C-function: } \mu = UR_e \textit{ and } \phi = UR_e - UR_b \qquad (1)$$

$$\textit{D-function: } \mu = UR_b \textit{ and } \phi = UR_e - UR_b \qquad (2)$$

$$\textit{T-function: } \mu = (UR_e + UR_b)/2 \textit{ and } \phi = (UR_e - UR_b)/2 \qquad (3)$$

This definition for uncertain moments has implications for the concept of interval since intervals are associated with two (start and end) moments, which can be precise or uncertain. Thus, we define uncertain intervals as any interval that has at least one of its two limits defined as an uncertain moment. Moreover, uncertain moments create intervals that have subintervals with different semantics. For example, consider a sentence such as "She started the diet on about 15/Jan (μ_1) and finished it on about 25/Jan (μ_2)". In this case, both intervals are uncertain moments defined by means of the T-function (Fig. 5).

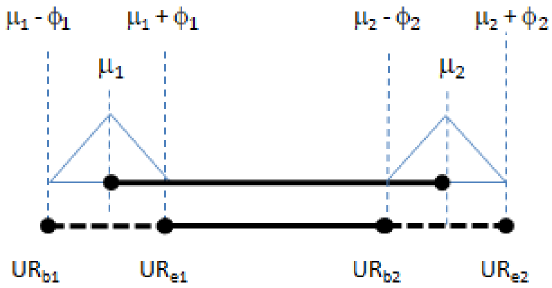


Fig. 5. Uncertain interval represented by two moments (T-functions).

We identify three subintervals in this example: [UR_{b1}, UR_{e1}], [UR_{b2}, UR_{e2}], and [UR_{e1}, UR_{b2}]. While the first two represent the uncertain part of the original interval, the latter represents its more precise part. The rationale is the same for D-functions or C-functions. In fact, both functions are simplifications of the T-function. This vision about intervals has implications on their relations, which are discussed in Sections 4.5 and 4.6.

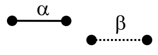
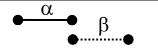
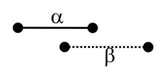
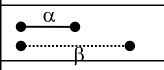
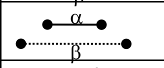
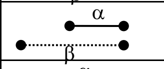
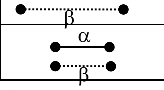
4.4 Precise Temporal Relations (R3)

Given the definitions of interval and moment, the next step is to specify the possible relations between these elements and their semantics. We start with the definition of relations between precise temporal notions (moment-moment, interval-interval, and moment-interval), according to the requirement R3, since their semantics are straightforward and based on the popular Allen's Interval Algebra [26], which is used in several other studies [4,18,27]. Moreover, the definition of precise temporal relations works as a basis for uncertain relations. Consider two time notions α and β that have a generic object property called *hasTemporalRelation*(α,β) between them. The first case is when α and β are both moments. In this situation, *hasTemporalRelation* assumes one of the three following relations:

- *before*(α,β): when $\alpha < \beta$. Equivalent to *after*(β,α);
- *after*(α,β): when $\alpha > \beta$. Equivalent to *before*(β,α);
- *equals*(α,β): when $\alpha = \beta$. Equivalent to *equals*(β,α).

The second case is when α and β are both intervals. In this situation, *hasTemporalRelation* assumes one of the thirteen Allen's relations specified in Table 2:

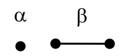
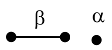
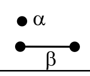
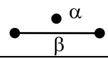
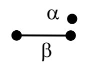
Table 2. Temporal relations between intervals

Relation	Conjunctive conditions	Example	Inverse Relation
before(α,β)	αe < βb		*after*(β,α)
meets(α,β)	αe = βb		*metBy*(β,α)
overlaps(α,β)	αb < βb βb < αe αe < βe		*overlappedBy*(β,α)
starts(α,β)	αb = βb αe < βe		*startedBy*(β,α)
during(α,β)	βb < αb αe < βe		*contains*(β,α)
ends(α,β)	βb < αb αe = βe		*enddBy*(β,α)
equals(α,β)	αb = βb αe = βe		*equals*(β,α)

αb = *beginning precise moment of the interval* α*;* αe = *end precise moment of the interval* α;
βb = *beginning precise moment of the interval* β*;* βe = *end precise moment of the interval* β.

The third case is when one of the temporal elements is a moment and the other is an interval. Consider, for example, that α is a moment and β is an interval. In this situation, *hasTemporalRelation* assumes one of the five relations in Table 3, which represent a subgroup of five Allen's relations that make sense for associations between a moment and an interval (e.g., *equal*, *meets* and *metBy*). Note that the same idea can be applied when α is an interval and β is a moment.

Table 3. Temporal relations between moment and interval

Relation	Conjunctive conditions	Example	Inverse Relation
before(α,β)	α < βb		*after*(β,α)
after(α,β)	βe < α		*before*(β,α)
starts(α,β)	α = βb		*startedBy*(β,α)
during(α,β)	βb < α α < βe		*contains*(β,α)
ends(α,β)	α = βe		*enddBy*(β,α)

α = precise moment; βb = beginning precise moment of the interval β; βe = end precise moment of the interval β.

4.5 Inference of Temporal Relations (R4)

The elegance of the practical representation of these relations comes from the structure illustrated in Fig. 3. We first need to define the thirteen Allen´s relations that will replace *hasTemporalRelation*. Then, the leaves of the temporal structure (interval, precise moment, and uncertain moment) are specified as *necessary and sufficient conditions*, which mean that a set of conditions is not only necessary for indicating instances of a concept, but also sufficient to determine that any individual that satisfies such a set must be an instance of this concept. Thus, the reasoning process can automatically identify the temporal concepts and the correct axioms that will be used to verify the temporal relations that hold between any two events.

In our study, the necessary and sufficient conditions are given by the type of information that a specific concept brings. For example, using the description logics syntax, we have the following concept definitions for precise moments (4) and intervals (5):

$$\text{PreciseMoment} \sqsubset \text{TimeNotion} \sqcap (\exists\ \text{hasPreciseTime.DataTimeStamp}) \quad (4)$$

$$\text{Interval} \sqsubset \text{TimeNotion} \sqcap (\exists\ \text{hasEnd.PreciseMoment} \sqcap \exists\ \text{hasBeginning.PreciseMoment}) \quad (5)$$

The first condition (4) says that any individual is an instance of the *PreciseMoment* concept if it is a time notion and has an associated *DataTimeStamp*, which is a concept of the OWL-Time ontology. The second condition (5) says that any individual is an instance of the *Interval* concept if it has both *beginning* and *end* information, which are from the *PreciseMoment* concept. The three data properties in (4) and (5) must also be specified as functional. This means, for any given individual, the property can have at most one value, which is coherent with their definitions (e.g., each interval can have only one moment for starting and only one for finishing).

Using these ideas, we can create rules to define the Allen´s relations, which work for relations involving intervals and moments. For example, consider the following definitions for the *before/after* relations in the Semantic Web Rules Language (SWRL) format. In this case, we have four rules with the same consequent because SWRL does not allow the explicit use of disjunctions in the rule body, maintaining in such a way the compatibility with the main reasoners. The elements *e1* and *e2* are two events that respectively occur in/at α and β. For the sake of simplicity, we first define Γ (6) as the common terms of all rules. Then, we define the semantics for the *before/after* relation considering moments (7), intervals (8), moment-interval (9), and interval-moment (10).

$$\Gamma \equiv \text{occurs}(?e1, ?a) \wedge \text{occurs}(?e2, ?b) \wedge \text{differentFrom}(?e1, ?e2) \quad (6)$$

$$\Gamma \wedge \text{hasPreciseTime}(?\alpha,?t1) \wedge \text{hasPreciseTime}(?\beta,?t2) \wedge \text{greaterThan}(?t2,?t1) \rightarrow \text{before}(?\alpha,?\beta) \wedge \text{after}(?\beta,?\alpha) \quad (7)$$

$$\Gamma \wedge \text{hasEnd}(?\alpha,?me) \wedge \text{hasBeginning}(?\beta,?mb) \wedge \text{hasPreciseTime}(?me,?t1) \wedge \text{hasPreciseTime}(?mb,?t2) \wedge \text{greaterThan}(?t2, ?t1) \rightarrow \text{before}(?\alpha,?\beta) \wedge \text{after}(?\beta,?\alpha) \quad (8)$$

$$\Gamma \wedge \text{hasPreciseTime}(?\alpha,?t1) \wedge \text{hasBeginning}(?\beta,?mb) \wedge \text{hasPreciseTime}(?mb,?t2) \wedge \text{greaterThan}(?t2, ?t1) \rightarrow \text{before}(?\alpha,?\beta) \wedge \text{after}(?\beta,?\alpha) \quad (9)$$

$$\Gamma \wedge \text{hasEnd}(?\alpha,?me) \wedge \text{hasPreciseTime}(?me,?t1) \wedge \text{hasPreciseTime}(?\beta,?t2) \wedge \text{greaterThan}(?t2, ?t1) \rightarrow \text{before}(?\alpha,?\beta) \wedge \text{after}(?\beta,?\alpha) \quad (10)$$

Observe this relation needs four rules to cover all situations. However this number depends on the relation. The meets/met-by relation, for example, only holds between intervals so it only needs one rule. It is also important to stress that such definitions do not change the transitivity axioms defined by Allen [26]. Thus, if the relations *before*(α,β) and *before*(β,λ) hold, then *before*(α,λ) also holds independently from the time notion type.

4.6 Initial Semantics for Uncertain Relations (R5)

The aim now is to extend the previous section so we can also consider uncertain temporal relations between classes, according to requirement R5. When we use a temporal relation such as *before*(α,β), we are sure that the time notion α occurs before the time notion β. However, consider the following situations with two uncertain moments (Fig. 6), which are represented by the reference time value μ and uncertain range value ϕ.

The first scenario (Fig. 6a) shows that α is before β even when we consider their uncertain ranges. This means, their uncertain ranges do not present intersections. However, it is not correct to state *before*(α,β) since the time moments are uncertain. Thus, we can use a subrelation of before called *almostSurelyBefore*(α,β). The second scenario (Fig. 6b) still presents a strong probability that α is before β. However, there is a minor chance that this is not true. Thus, we can use a subrelation of before called *strongSurelyBefore*(α,β). The third scenario (Fig. 6c) shows a strong probability that α is not before β. Thus, we can use a subrelation of before called *weakSurelyBefore*(α,β) since the relation can still hold. The fourth scenario (Fig. 6d) shows it is almost impossible for the relation to hold since α is after β and this case does not present any intersection between the uncertain ranges of α and β. Thus, the subrelation *almostSurelyNotBefore*(α,β) is used.

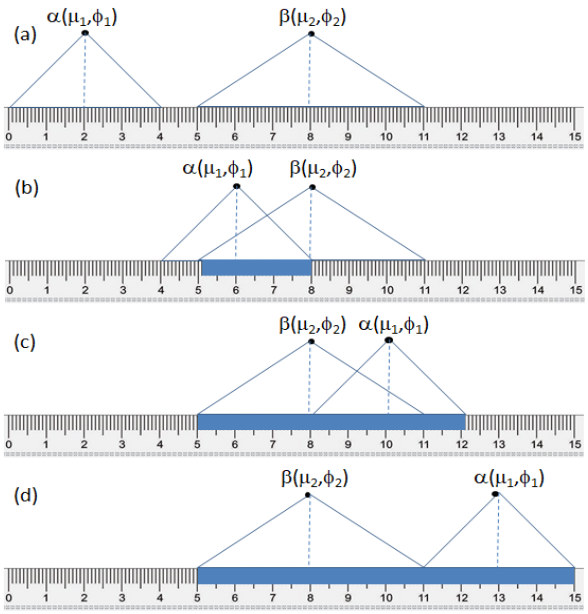


Fig. 6. Illustration of possible scenarios for temporal relations between two uncertain moments.

The next step is to formalize this approach. Therefore, consider Ω, which is defined in (11), as the variable that relates the uncertain positions between α and β.

$$\Omega = (\mu_2 - \phi_2) - (\mu_1 + \phi_1) \quad (11)$$

Then, we have the following conditions (12) to qualify the subrelations of *before*:

$$\begin{cases} \text{If } \Omega > 0, \text{ then } almostSurely \\ \text{If } 0 \geq \Omega > -(\phi_1 + \phi_2), \text{ then } strongSurely \\ \text{If } -(\phi_1 + \phi_2) \geq \Omega > -2(\phi_1 + \phi_2), \text{ then } weakSurely \\ \text{If } \Omega \leq -2(\phi_1 + \phi_2), \text{ then } almostSurelyNot \end{cases} \quad (12)$$

Example: Scenario (Fig. 6a) is *almostSurely* since,

$$\Omega = (\mu_2 - \phi_2) - (\mu_1 + \phi_1) = (8–3) - (2+2) = 1$$

As $\Omega > 0$, then *almostSurely*

Example: *Scenario* (Fig. 6b) is *strongSurely* since,

$\Omega = (\mu_2 - \phi_2) - (\mu_1 + \phi_1) = (8–3) – (6+2) = -3$

As $0 \geq \Omega > -5.0$, then *strongSurely*

Example: Scenario (Fig. 6c) is *weakSurely* since,

$\Omega = (\mu_2 - \phi_2) - (\mu_1 + \phi_1) = (8–3) – (10+2) = -7$

As $-5.0 \geq \Omega > -10.0$, then w*eakSurely*

Example: Scenario (Fig. 6d) is *almostSurelyNot* since,

$\Omega = (\mu_2 - \phi_2) - (\mu_1 + \phi_1) = (8–3) – (13+2) = -10$

As $\Omega \leq -10$, then *almostSurelyNot*

As the *after* relation is the inverse property of *before*, we just need to identify the subrelations of *after* as follows:

- *almostSurelyBefore*(α,β) ≡ *almostSurelyAfter*(β,α);
- *strongSurelyBefore*(α,β) ≡ *strongSurelyAfter* (β,α);
- *weakSurelyBefore*(α,β) ≡ *weakSurelyAfter* (β,α);
- *almostSurelyNotBefore*(α,β) ≡ *almostSurelyNotAfter* (β,α).

4.7 Allowing Flexibility for Uncertainty (R6)

The previous formalisation creates four subrelations for *before*. However, the number of such subrelations depends on the domain and should be configurable. In our approach, if any particular domain requires a more granular representation for subrelations, we only need to adjust the constants of the conditions. For example, consider that we need six rather than four subrelations (13). Then, the conditions could be (observe the constants in bold):

$$\begin{cases} \text{If } \Omega > 0, \text{ then } \textit{almostSurely} \\ \text{If } 0 \geq \Omega > \mathbf{-0.5}\,(\phi_1 + \phi_2), \text{ then } \textit{veryStrongSurely} \\ \text{If } \mathbf{-0.5}\,(\phi_1 + \phi_2) \geq \Omega > \mathbf{-1}(\phi_1 + \phi_2), \text{ then } \textit{StrongSurely} \\ \text{If } \mathbf{-1}(\phi_1 + \phi_2) \geq \Omega > \mathbf{-1.5}(\phi_1 + \phi_2), \text{ then } \textit{weakSurely} \\ \text{If } \mathbf{-1.5}\,(\phi_1 + \phi_2) \geq \Omega > \mathbf{-2}(\phi_1 + \phi_2), \text{ then } \textit{veryWeakSurely} \\ \text{If } \Omega \leq \mathbf{-2}(\phi_1 + \phi_2), \text{ then } \textit{almostSurelyNot} \end{cases} \tag{13}$$

This simple strategy of adjusting constants ensures a high flexibility to the representation, as required in R6.

4.8 Mixed Precise/ Uncertain Time Relations (R7)

The example in the previous schema (Fig. 6) uses triangular (T) functions to specify the uncertain moments. When the crescent (C) and decrescent (D) functions are included, we only need to redefine the equation (11) and conditions (12) according to Table 4. All these definitions involve two uncertain moments. However, we should also allow the mix between uncertain and precise moments. For example, the following schema (Fig. 7) presents two scenarios. The first scenario (Fig. 6a) has a precise moment (α) and an uncertain moment (β); while the second scenario (Fig. 6b) has an uncertain (α) and a precise moment (β). These scenarios show that precise moments can be considered as uncertain moments that have $\phi = 0$. Therefore, for such cases, we only need to consider the values in Table 5 when using the Ω function and conditions presented in Table 4.

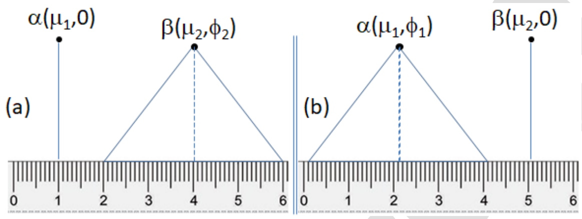


Fig. 7. Scenarios that involve precise and uncertain moments.

4.9 Extending the Semantics of Uncertain Relations (R5)

Complementing the implementation of R5, the next step is to specify the semantics of the remainder relations. Therefore, first consider the *equals* relation between two uncertain time points. This case is simpler than *before* because the *equals* relation is symmetric. Thus, we only need to consider subrelations such as *equals*(α,β) while $\alpha \leq \beta$. Consider the following schema (Fig. 8). The first scenario (Fig. 8a) shows that α is not equal to β even when we consider their uncertain ranges (no intersection). In this case, we use a subrelation called *almostSurelyNotEqual*(α,β), since it is not correct to state (*not equals*(α,β)) because the time moments are uncertain. The second scenario (Fig. 8b) presents a small intersection what means that there is a small probability that *equals*(α,β) holds. Thus, we can use a subrelation of *equals* called *weakSurelyEquals*(α,β).

Table 4. *before* subrelations including D and C functions

F(α)	F(β)	Equation (11) and conditions (12)
D	T	$\Omega = (\mu_2 - \phi_2) - (\mu_1 + \phi_1)$ If $\Omega > 0$, then *almostSurely* If $0 \geq \Omega > -(\phi_1 + \phi_2)$, then *strongSurely* If $-(\phi_1 + \phi_2) \geq \Omega > -(\phi_1 + 2\phi_2)$, then *weakSurely* If $\Omega \leq -(\phi_1 + 2\phi_2)$, then *almostSurelyNot*
T	D	$\Omega = \mu_2 - (\mu_1 + \phi_1)$ If $\Omega > 0$, then *almostSurely* If $0 \geq \Omega > -\phi_1$, then *strongSurely* If $-\phi_1 \geq \Omega > -(2\phi_1 + \phi_2)$, then *weakSurely* If $\Omega \leq -(2\phi_1 + \phi_2)$, then *almostSurelyNot*
C	T	$\Omega = (\mu_2 - \phi_2) - \mu_1$ If $\Omega > 0$, then *almostSurely* If $0 \geq \Omega > -\phi_2$, then *strongSurely* If $-\phi_2 \geq \Omega > -(\phi_1 + 2\phi_2)$, then *weakSurely* If $\Omega \leq -(\phi_1 + 2\phi_2)$, then *almostSurelyNot*
T	C	$\Omega = (\mu_2 - \phi_2) - (\mu_1 + \phi_1)$ If $\Omega > 0$, then *almostSurely* If $0 \geq \Omega > -(\phi_1 + \phi_2)$, then *strongSurely* If $-(\phi_1 + \phi_2) \geq \Omega > -(2\phi_1 + \phi_2)$, then *weakSurely* If $\Omega \leq -(2\phi_1 + \phi_2)$, then *almostSurelyNot*
D	C	$\Omega = (\mu_2 - \phi_2) - (\mu_1 + \phi_1)$ If $\Omega > 0$, then *almostSurely* If $0 \geq \Omega > -(\phi_1 + \phi_2)$, then *strongSurely* If $\Omega \leq -(\phi_1 + \phi_2)$, then *almostSurelyNot*
C	D	$\Omega = \mu_2 - \mu_1$ If $\Omega > 0$, then *almostSurely* If $0 \geq \Omega' > -(\phi_1 + \phi_2)$, then *weakSurely* If $\Omega \leq -(\phi_1 + \phi_2)$, then *almostSurelyNot*
C	C	$\Omega = (\mu_2 - \phi_2) - \mu_1$ If $\Omega > 0$, then *almostSurely* If $0 \geq \Omega > -\phi_2$, then *strongSurely* If $-\phi_1 \geq \Omega > -(\phi_1 + \phi_2)$, then *weakSurely* If $\Omega \leq -(\phi_1 + \phi_2)$, then *almostSurelyNot*
D	D	$\Omega = \mu_2 - (\mu_1 + \phi_1)$ If $\Omega > 0$, then *almostSurely* If $0 \geq \Omega > -\phi_1$, then *strongSurely* If $-\phi_1 \geq \Omega > -(\phi_1 + \phi_2)$, then *weakSurely* If $\Omega \leq -(\phi_1 + \phi_2)$, then *almostSurelyNot*

T = *triangular membership function, D = decrescent membership function, C = crescent membership function,* μ = *time reference,* ϕ = *uncertain range moment,* Ω = *numeric relation between two moments.*

Table 5. *before* relations considering precise moments

F(α)	F(β)	Adjust of values in Table 3
P	T	Line C-T, with $\phi_1 = 0$
T	P	Line T-D, with $\phi_2 = 0$
P	D	Line C-D, with $\phi_1 = 0$
D	P	Line D-C, with $\phi_2 = 0$
P	C	Line D-C, with $\phi_1 = 0$
C	P	Line C-D, with $\phi_2 = 0$

T = triangular membership function, D = decrescent membership function, C = crescent membership function, P = precise function, μ = *time reference*

The third scenario (Fig. 8c) shows a higher intersection what means that there is a high probability that *equals*(α,β) holds. Thus, we can use a subrelation of *equals* called *strongSurelyEquals*(α,β). The fourth scenario (Fig. 8d) shows the highest probability for *equals*(α,β) to hold. Then, the *almostSurelyEquals*(α,β) is used in such scenarios. Similarly to the *before* relation (Section 4.6), this example created four subrelations for *equals*. However, the number of such subrelations can be modified using different conditions (Section 4.7).

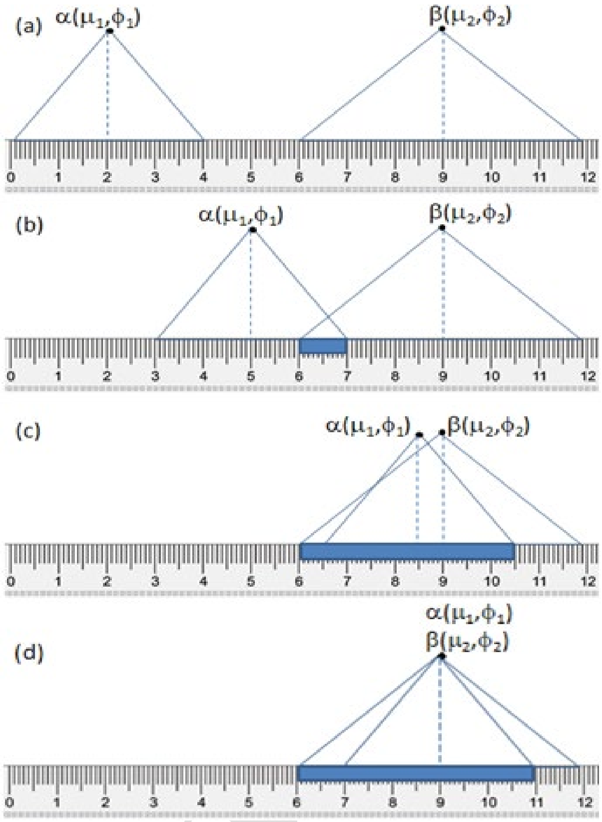


Fig. 8. Illustration of possible scenarios for subrelations of the "equals" temporal relation.

We use the same equation (11) to calculate the Ω value for the *equals* subrelations between triangular uncertain moments. However, the conditions are rewritten as (14):

$$\begin{cases} \text{If } \Omega > 0, \text{ then } \textit{almostSurelyNot} \\ \text{If } 0 \geq \Omega > -0.9\ (\phi_1 + \phi_2), \text{ then } \textit{weakSurely} \\ \text{If } -0.9\ (\phi_1 + \phi_2) \geq \Omega > -(\phi_1 + \phi_2), \text{ then } \textit{strongSurely} \\ \text{If } \Omega = -\ (\phi_1 + \phi_2), \text{ then } \textit{almostSurely} \end{cases} \qquad (14)$$

Note we have used a high constant value to define the *weakSurely* and *strongSurely* subrelations since we are dealing with equality between two moments. However, these values are easily adjusted and this flexibility is one of the main advantages of this approach.

As the *equals* relation is commutative, we also need to define the following equivalences:

- *almostSurelyNotEquals*(α,β) ≡ *almostSurelyNotEquals*(β,α);
- *weakSurelyEquals*(α,β) ≡ *weakSurelyAfter*(β,α);
- *strongSurelyBefore*(α,β) ≡ *strongSurelyAfter*(β,α);
- *almostSurelyEquals*(α,β) ≡ *almostSurelyEquals*(β,α);

Table 6 presents the adaptation of the conditions in (13) and equation in (11) for moments that use crescent (C) and decrescent (D) functions, together with precise moments (P).

Table 6. *equals* relations including D and C functions

<table>
<tr><th>F(α)</th><th>F(β)</th><th>Redefining equation (11) and conditions (14)</th></tr>
<tr><td>D</td><td>T</td><td rowspan="3">Same equation (11) and conditions in (14)</td></tr>
<tr><td>T</td><td>C</td></tr>
<tr><td>D</td><td>C</td></tr>
<tr><td>T | D</td><td>D | P</td><td>$\Omega = \mu_2 - (\mu_1 + \phi_1)$
Make $\phi_2 = 0$ in all conditions in (14)</td></tr>
<tr><td>C | P</td><td>T | C</td><td>$\Omega = (\mu_2 - \phi_2) - \mu_1$
Make $\phi_1 = 0$ in all conditions in (14)</td></tr>
<tr><td>C | P</td><td>D</td><td rowspan="2">$\Omega = \mu_2 - \mu_1$. The conditions are:
If $\Omega > 0$, then *almostSurelyNot*
If $\Omega = 0$, then a*lmostSurely*</td></tr>
<tr><td>C</td><td>P</td></tr>
</table>

T = triangular membership function, D = decrescent membership function, C = crescent membership function, P = precise function, μ = time reference, ϕ = uncertain range moment, Ω = numeric relation between two moments.

Our next aim is to show how to extend these uncertain relations between moments to relations between intervals and intervals-moments. Therefore, consider the conditions in Tables 2 and 3 for each of the Allen´s relations. Observe that all these relations are defined using moments and only two operations (“<” and “=”), which respectively represent the *before* and *equals* relations between moments. Thus, when the relation has only one condition, then the ideas discussed for relations involving uncertain moments (e.g., Ω equations, conditions in Table 4 and 5, etc.) are applied straightforward. For example, consider α and β as two uncertain intervals (Fig. 9) and the relation *meet*(α,β), whose condition is αe = βb (Table 2).

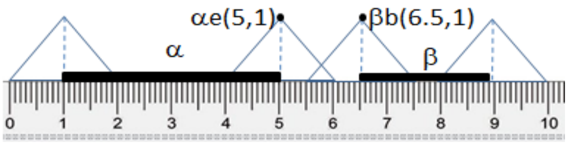


Fig. 9. Example of relation (*meet*) between two uncertain intervals.

The relation αe = βb is then mapped to *equals*(αe,βb) so we only need to use the equation (11) as follows:

$$\Omega = (\mu_2 - \phi_2) - (\mu_1 + \phi_1) = (6.5\text{-}1) - (5\text{+}1) = \text{-}0.5$$

As the uncertain functions are triangular and the relation is *equals*, then we use the second condition in (14) so the resultant qualifier is *weakSurely*. In conclusion, we have the relation *weakSurelyMeet*(α,β). This roadmap can be applied to any other relation of Tables 2 and 3 that have only one condition. When the relation has more than one condition (e.g., *overlaps* in Table 2), this process must be repeated to all conditions, so each relation will have two or three qualifiers. The intuition to deal with this situation comes from Fuzzy Logic, which represents intersections (and-fuzzy) as the minimum value between outcomes of membership functions. Therefore, we can specify the following resultant matrix (Table 7) for the qualifiers identified in (12) and (14). Note that the most uncertain qualifier is always returned as result.

Table 7. Resultant matrix for fuzzy logical conjunction

AND	**almostNot**	**weak**	**Strong**	**almost**
almostNot	*almostNot*	*almostNot*	*almostNot*	*almostNot*
Weak	*almostNot*	*weak*	*Weak*	*weak*
Strong	*almostNot*	*weak*	*Strong*	*strong*
Almost	*almostNot*	*weak*	*Strong*	*almost*

5. Evaluation

Three methods were used to evaluate our approach. Firstly, we employed the *OntOlogy Pitfall Scanner* (OOPS) checklist [30] to detected common issues in the ontology design. We chose this evaluation framework because it consolidates different types of ontology development issues based on the existing literature, also classifying such issues according to their level of importance. Moreover, OOPS has been extensively used in the literature. The second evaluation method was the Gold Standard-based approach [31]. This approach compares the new ontology with a previously created reference ontology known as the gold standard. Thus, the aim is to conduct a task like alignment or ontology mapping, verifying if the new ontology covers the elements of its gold standard version. Finally, we analyse if other approaches can represent the same knowledge of our ontology, emphasising in such a way their limitations and our contributions.

5.1 Applying the OntOlogy Pitfall Scanner (OOPS)

We have used the set of critical OOPS pitfalls [30] since their correction is crucial to ensure ontology consistency, reasoning, and applicability. This set is composed of 12 pitfalls, which are described in Table 8 together with comments about how we avoid them.

Table 8. OOPS-based analysis of our approach

Pitfall	**Conclusions of the OOPS-based analysis**
Creating Polysemous Elements	The analysis of the ontology lexicon (e.g., interval, moment, event, etc.) demonstrates that any of their terms present more than one conceptual idea.
Creating the Relationship "is" Instead of Using "rdfs:subClassOf", "rdf:type" or "owl:sameAs":	The analysis of the resultant owl file confirmed the use of "rdfs:subClassOf" instead of other options. The use of Protégé editor [32] automatically avoids this pitfall.
Defining Wrong Inverse Relationships	Our ontology is based on a consolidated formal temporal algebra [26] to define its inverse relationships.
Including Cycles in the Hierarchy	The visualization of the class hierarchy shows the inexistence of such cycles.
Misusing "owl:allValuesFrom"	Universal restrictions ("allValuesFrom") were not found in the resultant ontology (existential restrictions are used as the default qualifier instead of universal restrictions).
Misusing "not some" and "some not"	This pitfall was identified in the initial versions of the ontology. For example, an event occurs at a moment or during an interval. However, this fact does not imply that the sentences "some events ARE moments" and "some events are NOT moments" are equivalent. This latter is in fact equivalent to "NOT all events ARE moments". Thus, the correct use of "some" was fixed in our ontology.
Misusing Primitive and Defined Classes	Conditions used to define classes were converted from necessary into necessary and sufficient, ensuring the open-world assumption.
Swapping intersection and union (ranges and/or domain)	Our ontology follows the recommendation in [32], which advises against specifying ranges and domains for properties.
Defining wrong equivalent relationships	Equivalence between properties (*EquivalentTo*) was not used in our ontology.
Defining wrong symmetric relationships	Symmetric relationships were based on a consolidated formal temporal algebra [26].
Defining wrong transitive relationships	Transitivity is not directly defined in the ontology but inferred by the SWRL rules following the transitive table from the formal temporal algebra in [26], which ensures its correctness.
Defining wrong equivalent classes	We ensured that each class (e.g., moment, interval, etc.) has a unique and well-defined semantics.

This analysis (Table 8) shows that the ontology considered all the critical OOPS pitfalls. This fact avoids, for example, reasoning problems and wrong logical conclusions when the engines evaluate the SWRL inference rules. Detailed definitions of these pitfalls are out of the scope of this paper. These details, together with examples, are found in [30].

5.2 Gold Standard-based Evaluation

The gold standard for temporal aspects is the OWL-Time Ontology [18], which is also the W3C recommendation for temporal representations. Therefore, this evaluation considers the semantic similarities at the lexical level and the taxonomic overlap at the conceptual level between the ontologies. In other words, this evaluation verifies if our ontology can represent the same set of ideas that OWL-Time provides. Table 9 summarises the conclusions of our analysis.

Table 9. Summary of the gold standard-based evaluation of our approach

OWL-Time	Is it possible to represent this semantics? How?
Temporal-entity is the root class. It is linked to other classes using the *hasTime* property	Yes, *TimeNotion* is the root class. It is linked to other concepts using the *occurs* property.
Interval and *Instance* are classes derived from *Temporal-entity*.	Yes, *Interval* and *Moment* are concepts derived from *TimeNotion*
hasBeginning and *hasEnd* are properties of *Temporal-entity* used to define the limits of intervals. When *hasBeginning = hasEnd*, *Temporal-entity* is an *Instant*. Otherwise, it is an *Interval*.	Yes, but *hasBeginning* and *hasEnd* are properties of *Interval*. Thus, these properties are not used to define *Moments*. Moreover, engines can recognise *Intervals* only by checking if individuals (instances) have these properties.
hasDuration is a property of *Temporal-entity* and has the *Duration* class as its range. Thus, instances of *Duration* can present diverse related properties, which give more expressivity to the representation.	No, our approach does not present the *Duration* class because it increases the complexity of the uncertain representation. If an interval has a beginning or end (or both) uncertain moment, duration has infinity values. Thus, the concept of duration will be also fuzzy.
Provides all of Allen's relations as properties between *Intervals.*	Yes, all of Allen's relations are also provided as properties between *Intervals*.
Provides two additional temporal relations between intervals (*in* and *disjoint*) beyond Allen's relations.	No directly, but these relations can be represented as a logical disjunction of *during*, *starts,* and *finishes* properties for *in*; *before* and *after* for *disjoint*.
Allows the use of *before* and *after* properties to any two *Temporal-entity* individuals.	Yes, these properties can be applied to any two *TimeNotion* individuals.

OWL-Time defines several other classes and properties. As they are not part of the core structure of the representation, we can reuse such elements to extend our ontology. For example, we could relate the OWL-time classes of *TemporalUnit* and *DayOfWeek* to *Moment*, respectively using the OWL-time properties of *unitTime* and *dayOfWeek*.

5.3 Comparative Analysis

According to Table 1, and as far we know, the approach of Achich et al. [28] is the unique that considers the uncertainty of temporal classes and properties. We modelled (Fig. 10) the following sentence to compare our approach to Achich et al. [28]: "Marie use to wake up by 6:00h. Last night, the blood pressure device indicated she had a hypertension from 5:00h to around 5:45h".

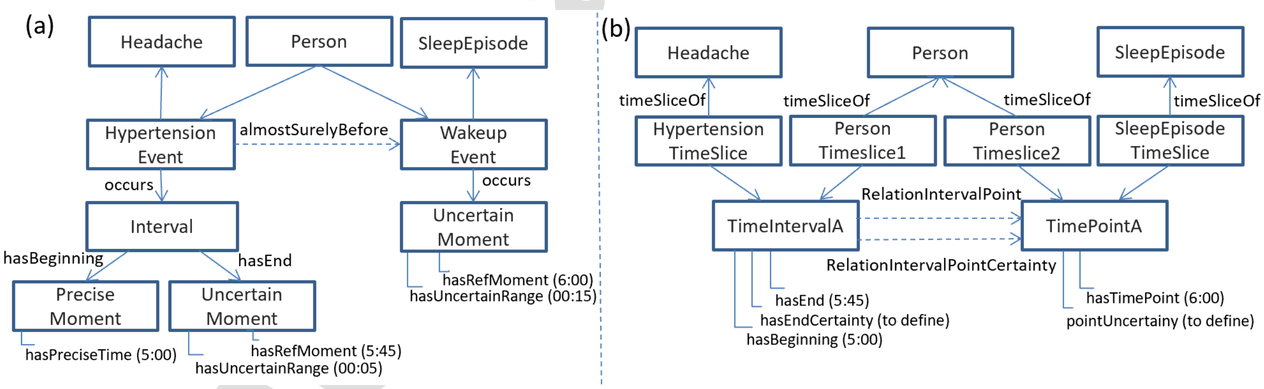


Fig. 10. Comparative modelling using our approach (a) and the approach discussed in [28] (b).

We have decided on a simpler structure where two concepts (e.g., Headache and Person) share the same event concept instead of having their own time slices. Moreover, we follow the OWL-Time standards, considering intervals and moments as subclasses of the same class (*TimeNotion* in Fig. 2). This strategy mainly allows the use of the same temporal properties between any two temporal instances, instead of creating specific properties according to the type of instances involved in such relations (e.g., *RelationIntervalPoint*). We also avoid the use of specific properties (e.g., *RelationIntervalPointCertainty*) to indicate the level of uncertainty. In our case, this semantics is qualitatively indicated using constructors (e.g., *almostSurely*) as part of the temporal properties, which are automatically generated by the set of rules discussed in this paper (Section 4). At the conceptual level, we defined clear semantics for properties. For example, *hasPreciseTime* is a data property of the *PreciseMoment* class, while *hasRefMoment* is a data property of the *UncertainMoment* class. As the ranges of these two properties have different semantics, we decided on names that emphasize this distinction. Apart from these aspects, the most important difference and conceptual contribution of our approach is its clear and well-defined semantics, which use relative distances between temporal elements to generate temporal relations. For example, we can visually see and understand the rationale of uncertainty definitions and use the fuzzy fundaments to augment their coverage (e.g., Table 7).

6. Case Example: nutrition assessment

This case example shows how our approach extends a static ontology, aimed at monitoring the nutritional habits of mobile users, with temporal concepts. Therefore, this section first presents the static ontology (Section 6.1) and the process of extension (Section 6.2). Then we introduce the scenario and form of evaluation, which is based on competence questions (Section 6.3). After that, we specify these competence questions as SQRLW queries and discuss the results (Section 6.4). This section concludes by discussing the contributions and limitations found in this case example (Section 6.5).

6.1 Ontology

The specification of the ontology, partially illustrated in Fig. 11, followed the guideline of Noy and McGuinness [32] to define the concepts and relations of this domain.

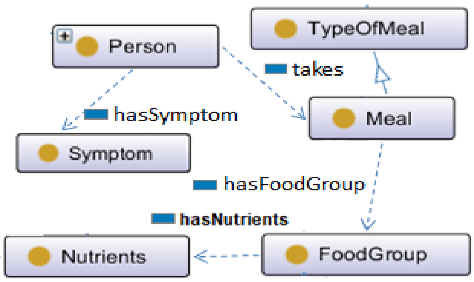


Fig. 11. Simplified representation of the nutritional habits' ontology.

The source material was the analysis of the mobile applications in Table 10, which lists four Android and four iOS apps associated with nutritional aspects and with the top evaluations (maximum rate = 5) in their app stores.

Table 10. Mobile apps relate to nutrition habits

App	*Rate*	*No of ratings*
MyFitnessPal apps.apple.com/us/app/my-fitnesspal/id341232718	4.7	1.0M
Lifesum 80mil apps.apple.com/us/app/lifesum-diet-macro-tracker/id286906691	4.6	81.3K
Progress Body Tracker & Health apps.apple.com/us/app/progress-body-tracker-health/id583840813	4.6	3.9K
WebDiet https://play.google.com/store/apps/details?id=br.com.webdiet.webdiet&hl=pt&gl=US	4.8	20.0K
Dietbox play.google.com/store/apps/details?id=com.craftbox.dietbox&hl=en_IN	4.7	4.6K
Diet Diary play.google.com/store/apps/details?id=com.canyapan.dietdiaryapp&hl=en	4.7	1.0K
Health Diet Foods Fitness Help play.google.com/store/apps/details?id=com.medical.guide_health.diet.tips&hl=en_IN	4.6	3.4K
Calorie Counter https://play.google.com/store/apps/details?id=com.fatsecret.android&hl=en_IN	4.5	2.2M

We only considered apps that are freely available and have more than 1000 ratings. Academic studies were also used to adjust some classes and properties of the ontology, such as the NESTORE models [33] and the e-NUTRI project [34]. This ontology (Fig. 11) indicates that individuals can take meals, which are composed of *FoodGroup* classes (e.g., eggs, fruits, grains, oil, vegetables, etc.), while each of these classes has nutritional components (e.g., fats, fibers, minerals, vitamins, etc.). Moreover, a meal has different types such as vegetarian, non-vegetarian, carbon-free, etc.

6.2 From Static to Temporal

The next algorithm represents a set of domain-independent steps, which are used to guide the conversion of static ontologies into their temporal versions. Each step has an example based on the nutritional habits ontology (Fig. 11).

Step 1. Define user-oriented interrogatives (competency questions) that allow thinking about temporal extensions for classes and properties of the target ontology.

Example. (Q1) When did Person p take any Meal m that is rich in the Nutrients fiber? (Q2) Did Person p take any TypeOfMeal liquid together with Meal m that is rich in the Nutrient protein before feeling some Symptom s? (Q3) Did Person p take any TypeOfMeal non-vegetarian during the two first days?

Step 2. Create new temporal classes that support the generation of answers for the previous questions.

Example. The class *EatingEvent* will represent the temporal event related to the *take* property (see Fig. 2). A second class *FeelingSimptomEvent* will represent the temporal event related to the *hasSymptom* property.

Step 3. Integrate the new classes into the original ontology with the modification of the domain and range of the object property that connects the classes involved into the temporal relation.

Example. The *takes* property (Fig. 11) is no longer a relation that only has Person as domain and Meal as range as they are now related to the new *EatingEvent* concept. Therefore, the new domain of the *take* property is the union (*Person* ∪ *EatingEvent*), while the range of the *take* property is the union (*EatingEvent* ∪ *Meal*).

Step 4. *Connect the temporal structure to each new temporal event using the "occurs" object property.*

Example. EatingEvent ⊓ occurs.TimeNotion

Step 5. *Update the knowledge base with the individuals (instances), their object and data properties*

Example. The time notion of all eating events must be inserted into the knowledge base. This notion can be a moment (John ate soup at 20:00), an interval (John started its dinner at 18:00 and finished at 19:00), an uncertain moment (John ate soup around 20:00), or an uncertain interval (John started its dinner by 18:00 and finished after 19:00).

Regarding the Step 5, the following object/data properties must be included into the time notions depending on their type:

- PreciseMoment
  - hasPreciseTime.dateTimeStamp
- PreciseInterval
  - hasBeginning.PreciseMoment
  - hasEnd.PreciseMoment
- UncertainMoment
  - hasRefTime.timeDataStamp
  - hasUncertainRange.duration
  - hasUncertainFunction.UncertainFunction
- UncertainInterval
  - hasBeginning.(PreciseMoment or UncertainMoment)
  - hasEnd.(PreciseMoment or UncertainMoment)

It is also important to consider the subrelations granularity of temporal relations. We are using four subrelations as default (*almostSurely*, *strongSurely*, *weakSurely*, and *almostSurelyNot*). However, this is a decision of the domain experts. Section 4.7 details how these uncertainty qualifiers can be modified.

6.3 Evaluation Setup

Evaluation of domain ontologies are commonly performed against a frame of reference, such as a set of competency questions (CQ), as discussed by Gomez-Perez [35] in her studies in the Knowledge System Laboratory at the University of Stanford. CQ is a strategy used for several methodologies as part of the specification and evaluation of ontologies [5, 36]. According to this strategy, we must define a set of questions in natural language that represent the user demands, given a scenario related to a domain of discourse. Therefore, CQs provide ontology engineers with a simple way to verify requirements' satisfiability by either knowledge retrieval or by entailment on its axioms and answer checking [37].

As an evaluation case, we apply our approach to support the identification of issues involving food intake and the health symptoms of an individual. Therefore, we use an adaptation (Table 11) of the *Elimination Diary Pad*, from the Scottish Nutrition and Diet Resources Initiative, as an assessment instrument to organize some fictitious information. This instrument is a template diary that helps record foods eaten to link symptoms/reactions with any dietary cause while following an elimination programme. The first column indicates the interval from the beginning of food intake until the end of the first digestion, which occurs when the food leaves the stomach (1º digestion), and the individual starts feeling hungry again. Individuals must also indicate when they feel some symptom, which normally happens until three hours after the intake start.

Table 11 - Food/symptoms assessment instrument

Intake + 1º digestion	Food & drink	Symptoms	Moment felt
DAY 01 (+0)			
At 6 by 9	**Breakfast**: milk, eggs and bread	Yes	About 8
At 12 to 15	**Lunch**: rice and red meat	No	-
Early 20 to 23	**Dinner**: soup and bread	No	-
DAY 02 (+24)			
At 6 by 9	**Breakfast**: milk, eggs and coffee	No	-
About 12 to about 15	**Lunch**: legume (potato) and fish	No	-
At 19 to 22	**Dinner**: vegetables (salad)	No	-
DAY 03 (+48)			
At 6 by 9	**Breakfast**: fruit and cereals	No	-
About 12 to early 15	**Lunch**: legume (potato) and cheese	No	-
At 17 to 20	**Snacks**: cake	No	-
At 18 to 21	**Dinner**: vegetables (salad)	Yes	By 19

We used the next set of competence questions, which are usually employed by a domain expert (nutritionist), considering a patient called Marie. We also included the rationale of the question given by the expert as follows:

- CQ1: When did Marie take any meal rich in fiber? (*to know when and the frequency of taking specific nutrients such as fiber*);
- CQ2: Did Marie take any liquid together with any food that is rich in protein when she felt some symptom? (*to identify situations where the digestive system is unable to adapt its secretions to the consistency of meals that are hard to break down*);
- CQ3: Did Marie eat any non-vegetarian meal during the two first days of the assessment? (*red meat and fish can take as long as two days to fully digest since they contain complex molecules that take longer for our body to pull apart. Thus, nutritionists may ask some patients to avoid eating these meals more than once on short periods such as two consecutive days*);
- CQ4: What did Marie eat when she felt some symptom? (*to identify possible nutrients that are the reason for such symptoms*);
- CQ5: Which nutrients were present more than once in meals that lead Marie to feel symptoms? (*to support the search for evidence on food allergy to specific nutrients*);
- CQ6: Is Marie taking food in short spaces (less than 3 hours)? (*the interval between meals should be at least three hours*);
- CQ7: Which are the time intervals between the moments that Marie felt symptoms? (*to identify the frequency that symptoms are occurring*).

All resources used in this evaluation are publically available as online supplemental material[1], allowing the replication of these results. They are the temporal ontology, ontology with instances of test (nutritional domain); text file with the SWRL rules, and text file with SQWRL queries. We employed the Protégé version 5.5.0 over these experiments, together with the Pellet Reasoner Plug-in 2.2.0.

6.4 Queries and Results

We employed the Semantic Query-Enhanced Web Rule Language (SQWRL) [38] to codify the previous seven competence questions (CQ1 to CQ7), which are described as follows together with the answers

[1] http://onto-mqol.unige.ch:3030/

returned, and their level of uncertainty (Table 12). Moreover, we discuss issues found during such a codification, which indicate limitations in the semantic expressiveness of the temporal ontology.

Table 12 – Query Results

CQ	Answer	Uncertainty
CQ1	meal:DinnerDay03, when: I08 meal: BreakfastDay03, when: I05 meal:DinnerDay02, when: I04	Precise
CQ2	meal:BreakfastDay01	Almost Surely
CQ3	meal:LunchDay01	Precise
	meal:LunchDay02	Almost Surely
CQ4	e2: FeelingSymptomEvent01, fg: Bread fg: Eggs fg: Milk e2: FeelingSymptomEvent02, fg: Cake fg: Vegetable	Almost Surely
CQ5	e1: FeelingSymptomEvent01, e3: FeelingSymptomEvent02, nut1: Carbohydrate	Almost Surely
CQ6	meal1: Snack01 meal2: DinnerDay3	Precise
CQ7	e2:FeelingSymptomEvent01 e1:FeelingSymptomEvent02 hours: 59	Partial support for this definition

The first codification (CQ1) is simple and exemplifies how we can return time elements that are associated with other classes of the ontology. The uncertainty is always the same as the uncertainty of the time notion returned.

CQ1: *takes(Marie, ?e) ^ takes(?e, ?meal) ^ hasFoodGroup(?meal, ?fg) ^ hasNutrient(?fg, Fiber) ^ occurs(?e, ?time) → sqwrl:select(?meal) ^ sqwrl:select(?time)*

In the second codification (CQ2), *hasContainsRelation* represents one or more definitions of the *contains* relation. For example, we used *contains* and *almostSurelyContains* to return the results in Table 12. While the former did not return answers, the later returned one answer. We use this same idea in the following queries. In such cases, the uncertainty level of the answer is the same as the relation used in the query.

CQ2: *feels(Marie, ?e0) ^ feels(?e0, StomachAche) ^ takes(Marie, ?e1) ^ takes(?e1, ?meal) ^ hasFoodGroup(?meal, ?fg1) ^ hasFoodGroup(?meal, ?fg2) ^ differentFrom(?fg1, ?fg2) ^ hasNutrient(?fg1, Water) ^ hasNutrient(?fg2, Protein) ^* ***hasContainsRelation(?e1, ?e0)*** *→ sqwrl:selectDistinct (?meal)*

The codification of the third question (CQ3) shows the strategy of defining intervals of interest (e.g., *TwoFirstDaysEvents*) as new events that limit the period that we intend to analyse. Thus, we can identify the events of interest as those that occur during this period.

***CQ3**: takes(Marie, ?e) ^ takes(?e, ?meal) ^ NonVegetarian(?meal) ^ **hasDuringRelation(?e, TwoFirstDaysEvent)** → sqwrl:select(?meal)*

The fourth codification (CQ4) is interesting because it discovered a food group (cake) that was not directly associated with the symptom in the food diary (Table 11).

***CQ4**: takes(Marie, ?e1) ^ takes(?e1, ?meal) ^ hasFoodGroup(?meal, ?fg) ^ feels(Marie, ?e2) ^ **hasDuringRelation (?e2, ?e1)** → sqwrl:select(?e2, ?fg) ^ sqwrl:orderBy(?e2)*

This fifth codification (CQ5) requires two temporal relations (*hasContainsRelation*). Thus, the uncertainty of the answers considers the principles of Table 7. For example, in this case, the answers were obtained with two *almostSurelyContains* relation. Thus, the answers have the same level of uncertainty (Table 12).

CQ5: *feels(Marie, ?e1) ^ takes(Marie, ?e2) ^ takes(?e2, ?meal1) ^ differentFrom(?e1, ?e2) ^ **hasContainsRelation (?e2, ?e1)** ^ feels(Marie, ?e3) ^ takes(Marie, ?e4) ^ takes(?e4, ?meal2) ^ differentFrom(?e3, ?e4) ^ **hasContainsRelation (?e4, ?e3)** ^ hasFoodGroup(?meal1, ?fg1) ^ hasNutrient(?fg1, ?nut1) ^ hasFoodGroup(?meal2, ?fg2) ^ hasNutrient(?fg2, ?nut2) ^ differentFrom(?e1, ?e3) ^ sameAs(?nut1, ?nut2) → sqwrl:select(?e1, ?e3, ?nut1)*

The sixth codification (CQ6) uses the *overlaps* relation to indicate that one meal starts before the end of the stomach digestion of the previous meal. As this stomach digestion typically takes three hours, the interval between these meals was less than 3 hours (there was an overlap).

CQ6: *takes(Marie, ?e1) ^ takes(?e1, ?meal1) ^ takes(Marie, ?e2) ^ takes(?e2, ?meal2) ^ differentFrom(?e1, ?e2) ^ **overlaps(?e1, ?e2)** → sqwrl:select(?meal1, ?meal2)*

The seventh codification (CQ7) only partially specifies the query proposed. Our approach allows the indication of quantitative values for moments, so the difference between these values gives the quantitative temporal distance between two moments. This idea is correct when the moments are precise. However, if one of these values is uncertain, a level of uncertainty should also be given for this difference. An initial intuition is to use the same framework presented here to also qualify the quantitative distance between them. This extension is part of our future studies.

CQ7: *feels(Marie, ?e1) ^ Moment(?m1) ^ occurs(?e1, ?m1) ^ hasRefMoment(?m1, ?r1) ^ feels(Marrie, ?e2) ^ Moment(?m2) ^ occurs(?e2, ?m2) ^ hasRefMoment(?m2, ?r2) ^ almostSurelyAfter(?e1, ?e2) ^ swrlb:subtract(?r3, ?r1, ?r2) ^ swrlb:divide(?hours, ?r3, 60) -> sqwrl:select(?e2, ?e1, ?hours)*

6.5 Discussion

One of the main contributions of this section was to show the process of extending a static ontology to its temporal version using our approach. The five steps presented are relatively straightforward and they do not require significant changes in the original ontology. This knowledge engineering process is another contribution of this study, and its specification in the form of a tutorial is part of our current efforts. This case example also shows how competency questions are mapped to logic queries, which use the expressivity of the ontology regarding uncertain times to reason and generate new knowledge.

However, our approach also presents some limitations. The semantic modelling of uncertain time used fuzzy-based functions to model natural language terms such as about 12:00, by 12:00 and early 12:00. Consider now the expression "Marie felt a symptom between 12:00 and 15:00". In other words, all moments between 12:00 and 15:00 have the same probability to be the moment that Marie felt the symptom. Our approach does not support this semantic. An initial intuition to cover this situation is to use a quadratic fuzzy function and a further set of rules that consider this new option.

A second pragmatic limitation is an explosion in the number of rules when the number of subrelations increases. This issue mainly happens because we cannot use disjunctions to connect terms of rules. Thus, we need to specify very similar rules. For example:

First rule: *Moment(?a) ^ Moment(?b) ^ differentFrom(?a, ?b) ^ hasUncertainFunction(?a, Decrescent) ^ hasRefMoment(?a, ?m1) ^ <u>hasPreciseTime(?b, ?m2)</u> ^ hasUncertainRange(?a, ?o1) ^ swrlb:add(?r1, ?m1, ?o1) ^ swrlb:subtract(?omega, ?m2, ?r1) ^ swrlb:greaterThan(?omega, 0) -> almostSurelyBefore(?a, ?b)*

Second rule: *Moment(?a) ^ Moment(?b) ^ differentFrom(?a, ?b) ^ hasUncertainFunction(?a, Decrescent) ^ hasRefMoment(?a, ?m1) ^ <u>hasUncertainFunction(?b, Decrescent) ^ hasRefMoment(?b, ?m2)</u> ^ hasUncertainRange(?a, ?o1) ^ swrlb:add(?r1, ?m1, ?o1) ^ swrlb:subtract(?omega, ?m2, ?r1) ^ swrlb:greaterThan(?omega, 0) -> almostSurelyBefore(?a, ?b)*

There is only a slight difference between these two rules. Thus, the disjunction operator could merge them. However, this no-use of disjunctions is a compulsory feature of DL that avoids computational complexity since every disjunction possibly multiplies process time by 2. Therefore, if we have *n* disjunctions, we tend to get 2 to the power of n possibilities to explore (satisfiability is NP-complete).

7. Conclusions

This approach presents, as main contributions, an ontological representation for temporal concepts and a set of rules to implement the semantics of the relations between these concepts. As this set of rules includes all the possible relations between two time notions, the transitivity tables between relations are already ready for use.

The case example showed that this ontology supports the definition of deductive reasoning, which was particularly implemented using the SQWRL framework. However, a potential application of our approach is to support the generation of explanations for inductive systems. This is possible because our approach generates rich temporal semantic descriptions between elements of a domain. Thus, if temporal aspects have an influence on the outcomes of an inductive system, such as a neural network, then a neuro-symbolic Network [39] could use this ontology to augment the space of search for explanations.

Directions of this study intend to cover the concept of uncertainty applied to the quantitative temporal distance between events, investigate other temporal natural language terms that are not currently covered, and verify the potential of these semantic descriptions in experiments using neuro-symbolic frameworks.

Acknowledgment

This project (Onto-mQoL) has received funding from the European Union´s Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grand agreement H2020-MSCA-IF-2020-101024693. The authors wish to thank Dr. Faustina Hwang from the Department of Biomedical Engineering - University of Reading (UK), for the discussions on the nutrition domain.

References

[1] F. Baader, I. Horrocks, And U. Sattler, “Description logics,” Foundations of Artificial Intelligence, 3, 135-179, 2008.

[2] H.T. Wang and A.U. Tansel, “Temporal Extensions to RDF,” *Journal of Web Engineering*, 18(1), 125-168, 2019.

[3] S. D. Cardoso, M. Da Silveira, and C. Pruski, “Construction and exploitation of an historical knowledge graph to deal with the evolution of ontologies,” *Knowledge-Based Systems*, 2020, 194, 105508.

[4] S. Batsakis, EG. Petrakis, I. Tachmazidis, G. Antoniou, “Temporal representation and reasoning in OWL 2,” *Semantic Web*, 8(6), 981-1000, 2017.

[5] C. Bezerra, F. Freitas, and F. Silva, "Evaluating Ontologies with Competency Questions," *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies* (IAT), 3:284-285, 2013.

[6] H. Bolhasani, M. Mohseni, and A. M. Rahmani, A. M., "Deep learning applications for IoT in health care: A systematic review," Informatics in Medicine Unlocked, 23, 100550, 2021.

[7] M. H. Kashani, M. Madanipour, M. Nikravan, P. Asghari, and E. Mahdipour, "A systematic review of IoT in healthcare: Applications, techniques, and trends," Journal of Network and Computer Applications, 192, 103164, 2021.

[8] K. Wac, "From quantified self to quality of life," In Digital Health, 83-108. Springer, Cham, 2018.

[9] Yeh, W. C., & Lin, J. S. (2018). New parallel swarm algorithm for smart sensor systems redundancy allocation problems in the Internet of Things. The Journal of Supercomputing, 74(9), 4358-4384.

[10] S. Wulfovich, J. Buur, and K. Wac, "Unfolding the Quantification of Quality of Life," In Quantifying Quality of Life, 3-24. Springer, Cham, 2022

[11] A. H. Mokdad, J. S. Marks, D. F. Stroup, and J. L. Gerberding, "Actual causes of death in the United States, 2000," Jama, 291(10), 1238-1245, 2004.

[12] K. Wac, "The Future of Quantifying Behaviors, Health, and Quality of Life," Quantifying Quality of Life, 2, 585, 2022.

[13] K. Wac, and S. Wulfovich, "Quantifying Quality of Life: Incorporating Daily Life into Medicine," Health Informatics Book Series, 602, Srpinger, 2022.

[14] J. Leo, N. Matentzoglu, U. Sattler, and B. Parsia, "A systematic survey of temporal requirements of bio-health ontologies," *Semantic Web*, 11(4), 657-688, 2020.

[15] M. Moens, and M. Steedman, "Temporal ontology in natural language" In 25th Annual Meeting of the Association for Computational Linguistics, pp. 1-7, 1987.

[16] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, and M. Zakharyaschev, "First-order rewritability of temporal ontology-mediated queries," In Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[17] C. Pedrinaci, J. Domingue, and A. Medeiros, "A core ontology for business process analysis," In European Semantic Web Conference, pp. 49-64, Springer, Berlin, Heidelberg, 2008.

[18] S. Cox, and C. Little, Time ontology in OWL. W3C Working Draft 12 July 2016. W3C. Retrieved from http://www. w3.org/TR/owl-time.

[19] P. Cabalar, and T. Schaub, "Temporal Logic Programs with Temporal Description Logic Axioms," In *Description Logic, Theory Combination, and All That*, pp. 174-186, Springer, Cham, 2019.

[20] C. Lutz, "Description Logics with Concrete Domains - A Survey," *In Advances in Modal Logics*, pp. 265-296, 2003.

[21] C. Gutierrez, C. Hurtado, and A. Vaisman, "Temporal RDF," In European Semantic Web Conference, pp. 93-107, 2005.

[22] G. Nagypál, and B. Motik, "A Fuzzy Model for Representing Uncertain, Subjective, and Vague Temporal Knowledge in Ontologies", In OTM Confederated International Conferences, On the Move to Meaningful Internet Systems, pp. 906-923, Springer, Berlin, Heidelberg, 2003.

[23] H. Cheng, L. Yan, Z. Ma, and S. Ribarić, "Fuzzy spatio-temporal ontologies and formal construction based on fuzzy Petri

nets. Computational Intelligence," 35(1), 204-239, 2019.

[24] C. Welty, R. Fikes, and S. Makarios. "A Reusable Ontology for Fluents in OWL," *Formal Ontology in Information Systems*, 150, 226–236, 2006.

[25] M. Giunti, G. Sergioli, G. Vivanet, and S. Pinna. "Representing n-ary relations in the Semantic Web," *Logic Journal of the IGPL*, jzz047, 2019.

[26] L. Allen, "Towards a General Theory of Action and Time," *Artificial Intelligence*, 23(2): 123-154, 1984.

[27] F. Ghorbel, F. Hamdi, E. Métais, N. Ellouze, and F. Gargouri, Ontology-based representation and reasoning about precise and imprecise temporal data: A fuzzy-based view. *Data & Knowledge Engineering*, 124, 101719, 2019.

[28] N. Achich, F. Ghorbel, F. Hamdi, E. Métais, and F. Gargouri, "Approach to reasoning about uncertain temporal data in OWL 2," Procedia Computer Science, 176, 1141-1150, 2020

[29] B. Bahadorani, and A. Zaeri, "Comparison of time concept modeling for querying temporal information in OWL and RDF," *International Journal of Information Technology and Computer Science* (IJITCS), 9(7), 26-34, 2017.

[30] M. Poveda-Villalón, A. Gómez-Pérez, and C.M. Suárez-Figueroa, "Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation," *International Journal on Semantic Web and Information Systems*, 10(2): 7-34, 2014.

[31] A. Maedche, and S. Staab, "Measuring similarity between ontologies," *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, Ontologies and the Semantic Web*, pp. 251-263, 2002.

[32] NF. Noy, and DL McGuinness, "Ontology development 101: A guide to creating your first ontology," Knowledge Systems Laboratory, Stanford University, 2001.

[33] A. Mastropietro, C. Roecke, S. Porcelli, J. del Bas, N. Boquè, L. Maldonado, G. Rizzo, "Multi-domain Model of Healthy Ageing: The Experience of the H2020 NESTORE Project," *In Italian Forum of Ambient Assisted Living*, 2018, pp. 13-21.

[34] R. Fallaize, R.Z. Franco, J. Pasang, F. Hwang, J.A. Lovegrove, "Popular Nutrition-Related Mobile Apps: An Agreement Assessment Against a UK Reference *Method," JMIR mHealth and uHealth*, vol. 7, no. 2, e9838, 2019.

[35] A. Gomes-Perez, "Some Ideas and Examples to Evaluate Ontologies," *Proceedings the 11th Conference on Artificial Intelligence for Applications*, 1995, pp. 299-305.

[36] M. Fernandez-López, A. Gomez Perez, and N. Juristo, "Methontology: From ontological art towards ontological engineering,"in *Ontological Engineering on Spring Symposium Series*, Stanford, 1997.

[37] J. Ashraf, E. Chang, O. Hussain, and F. K. Hussain, "Ontology usage analysis in the ontology lifecycle: A state-of-the-art review," *Knowledge-Based Systems*, 2015, 80, 34-47.

[38] MJ. O'Connor, and AK Das, "SQWRL: a query language for OWL," *Proceedings of the 6th International Conference on OWL: Experiences and Directions*, vol. 529, pp. 208-215, 2009.

[39] L. Bühmann, J. Lehmann, and P. Westphal, "DL-Learner—A framework for inductive learning on the Semantic Web," *Journal of Web Semantics*, vol. 39, pp. 15-24, 2016.

HIGHLIGHTS

- Semantics definition of precise and uncertain time notions and relations
- Support for representations of different levels of uncertainty between events
- Compatibility with the current Semantic Web standards
- Knowledge engineering method to include time notion in static ontologies

Sample CRediT author statement

Clauirton Siebra: Conceptualization, Methodology, Validation, Writing Reviewing and Editing .

Katarzyna Wac: Supervision, Project administration, Writing- Reviewing and Editing.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: