



## An Ontological Approach to Representing Complex Temporal Expressions in OWL-Time

<sup>1</sup>Yusuf Aminat Bolatito, <sup>2</sup>Junaidu Sahalu Balarabe, <sup>2</sup>Obiniyi Afolayan Ayodele, <sup>2</sup>Aliyu Salisu, <sup>2</sup>Kana Armand, <sup>2</sup>Florentin Donfack

<sup>1</sup>Department of Information Communication Engineering,  
Usmanu Danfodiyo University Sokoto, Sokoto

<sup>2</sup>Department of Computer Science,  
Ahmadu Bello University Zaria, Kaduna Nigeria

### ABSTRACT

Temporal information encompasses both simple and complex time. Foundational dynamic ontologies typically prioritize the representation of simple time, often neglecting the intricacies associated with complex time. In this context, complex time involves recurring time, imprecise time, and multi-granularity time. Existing foundational dynamic ontologies, such as OWL-Time, are proficient in capturing qualitative information, precise dates, times, and durations. However, they face challenges in encapsulating the nuanced attributes inherent in complex time. Current ontology efforts, particularly within the OWL framework, address complex recurring time through individual extensions rather than by enriching foundational dynamic ontologies with dedicated concepts and relations. The challenge of recurring time, rooted in non-convex interval time, has been extensively discussed in classical literature. To address this challenge within OWL-Time, our work proposes a novel dynamic ontology using a lightweight approach that decomposes the primitive intervals and properties of OWL-Time. This approach also embeds periodic rules, which are advantageous as they defragment patterns of repetition to specific conceptual levels, moving beyond a narrow snapshot focus to a broader conceptual understanding. Toward validate the effectiveness of our approach, we conducted an empirical study assessing the complexity of the ontology using real-world datasets. The experimental results underscore the extended ontology's capacity to faithfully represent and reason of recurring temporal expressions, demonstrating its efficacy in covering a diverse array of recurring time-related classes across various domains.

### INTRODUCTION

The Semantic Web serves as a pivotal framework for organizing, capturing, and processing data into a machine-readable format. Within this landscape, ontologies have played a vital role in representing high-level concepts, their properties, and interrelationships. As the demand for sharing, exchanging, and querying temporal information across diverse domains has surged, existing ontologies encounter scalability limitations for automatic reasoning of temporal

### ARTICLE INFO

#### Article History

Received: September, 2023

Received in revised form: November, 2023

Accepted: January, 2024

Published online: June, 2024

### KEYWORDS

Temporal information, Foundational dynamic ontologies, Recurring time, OWL-Time

properties (Madkour et al., 2016), as they can only represent resources in their current state (Batsakis & Petrakis, 2011).

Various approaches have emerged to integrate temporal aspects into ontologies, encompassing the introduction of new datatypes, properties, and operators (Ghorbel et al., 2019). Foundational dynamic ontologies, including Cyc time (Lenat & Guha, 1991), Suggested Upper Merged Ontology (Pease et al., 2001), Descriptive Ontology for Linguistic and Cognitive Engineering

Corresponding author: Yusuf Aminat Bolatito

✉ [aminbolly@gmail.com](mailto:aminbolly@gmail.com)

Department of Information Communication Engineering, Usmanu Danfodio University, Sokoto, Sokoto State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved.



(Masolo et al., 2003), Basic Formal Ontology (Borgo et al., 2003), General Formal Ontology (Baumann et al., 2014), Web Ontology Language (OWL) Time Ontology (Hobbs & Pan, 2004), SWRL Temporal Ontology (O'Connor & Das, 2011), Reusable Time Ontology (Fikes & Zhou, 2002), TimeLine Ontology (Raimond & Samer, 2007), and SOWL Ontology (Batsakis et al., 2017), strive to address these challenges. However, dynamic ontologies often fall short in representing evolving temporal information comprehensively, particularly in managing complex time (Siebra & Wac, 2022).

Research indicates that a significant portion of web information, particularly in natural language texts, involves temporal information (Ijntema et al., 2010; Pan, 2007), covering both simple and complex time (Gu et al., 2023). Existing dynamic ontologies struggle to represent complex time due to the high diversity of natural language, hindering the full expression of temporal knowledge needed by various applications (Gu et al., 2023). For instance, OWL-Time, a widely used ontology, provides temporal information for both quantitative (e.g., dates) and qualitative terms (e.g., "During") but faces challenges in representing complex time expressions. OWL-Time is unable to represent important types of complex recurring time expressions (Bento et al., 2017; Pan, 2007), such as "every Wednesday at 10 A.M.," and faces challenges in providing complex time concepts and relations for multi-granularity (Zhang et al., 2021) and imprecise time (Ghorbel et al., 2019).

In various knowledge domains (e.g., health, education, and history), simple and complex temporal data coexist, playing a crucial role in understanding the domain of discourse for prediction, trajectory time tracking, and understanding patterns of occurrences (Abbasimehr & Baghery, 2022; Baji et al., 2022). The representation of complex times in diverse domains remains a challenge. The lack of expressive standards restricts the range of reasoning processes in OWL-Time because several time aspects, usually embedded in knowledge domains (e.g., health and education) (Cardoso et al., 2020), cannot be employed.

This paper specifically addresses the integration of complex recurring time as a crucial extension to the OWL-Time ontology. Existing approaches to handling recurring temporal information in OWL, such as those by Hu et al. (2022), tend to focus on individual domain modeling, overlooking the enrichment of the foundational dynamic ontology, OWL-Time. Few proposals in the Semantic Web field have aimed to represent and reason about recurring temporal information in OWL-Time (Bento et al., 2017; Poveda et al., 2014). However, these efforts typically introduce additional primitive entities that fall short in fully enriching the open domain ontology, leaving a gap in addressing the discourse problem. Our proposed ontology, OWL-BOLA, serves as an extension that introduces a novel class for non-convex intervals and adds more properties to the OWL-Time calendar description.

The paper is organized as follows. Section 2 presents related works, Section 3 details the materials and methods for the ontology-based approach, and Section 4 discusses the results of our evaluation with the literature. Finally, Section 5 draws conclusions and outlines future research directions.

## RELATED WORKS

The incorporation of temporal representation into ontologies and the semantic web have been the subject of much discussion. This study review effort in ontology in the context of OWL that addresses complex recurring time individually.

Time-Entry Ontology (Pan, 2007) addresses the challenges of recurring time representation by introducing a parallel structure to OWL-Time. It replicates OWL-Time's primitive entities and introduces a new class called TemporalSeq, specifically designed to handle recurring time instances. This method includes two primary calendar descriptions tailored to TemporalSeq. The TemporalAggregateDescription component demonstrates resilience in managing regular recurring time, such as twice every week and three times in 2023. While Pan's method excels in addressing recurring time instances, it falls short

---

Corresponding author: Yusuf Aminat Bolatito

✉ [aminbolly@gmail.com](mailto:aminbolly@gmail.com)

Department of Information Communication Engineering, Usmanu Danfodio University, Sokoto, Sokoto State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved.



in representing or reasoning about non-recurring time, such as events on every April 25 between 15:00 and 18:00, and every April 26 between 9:00 and 11:00, as highlighted by Bento et al. (2017). Additionally, it lacks capabilities for handling irregular recurring time like "every Monday, closed on holiday December 25.

Ontology by Poveda et al. (2014) present an approach that enhances the OWL-Time framework by focusing on non-convex intervals through the PeriodicInterval class. This class consolidates essential elements, such as the start and end of intervals, subinterval durations, overall period duration, and subinterval counts. The ontology effectively utilizes existing OWL-Time properties like `hasBeginning`, `hasEnd`, `DateTimeDescription`, and `DurationDescription`. Despite its strengths, the ontology primarily models regular recurring time and lacks formal reasoning, limiting its precision and clarity in temporal representation of irregular counterparts.

NCO Temporal Ontology (Bento et al., 2017) extends OWL-Time by introducing a new primitive concept, `NCInterval`, within the `TemporalEntities` class. It also develops a comprehensive calendar description framework by extending `GeneralDateTimeDescription` to create `GeneralDateTimeLimit`. This approach effectively manages recurring time and excels in non-recurring time scenarios. However, it may overlook aspects such as irregular recurring time.

The Temporal Event Ontology (TEO) by Li et al. (2020) addresses temporal aspects in the clinical domain, including recurring time, by introducing the concept of `PeriodicInterval`. This enhancement incorporates additional healthcare-specific features, enriching OWL. However, TEO does not directly extend OWL-Time but builds on the work of Tao et al. (2010). It falls short in representing imprecise time and certain aspects of recurring time, such as regular recurring time ("three times in 2023"), irregular recurring time ("every Monday, closed on December 25"), and non-recurring time as found in Bento et al. (2017).

Clinical Time Ontology (CTO) by Hu et al. (2022) extends TEO to address its limitations, particularly in handling imprecise time and some aspect of recurring time in the clinical domain. It

introduces the `ClinicalTime` class to manage potential time discrepancies and enriches OWL with domain-specific features. However, similar to TEO, it does not directly extend OWL-Time.

## METHODOLOGY

OWL-Time (Hobbs & Pan, 2004; Hobbs et al., 2017) enables the representation of time through both instantaneous points and intervals (convex intervals). However, several types of temporal events require representation over non-convex intervals, which involve periods with gaps, such as "every Monday at 10 AM" or "every Monday except for holidays like December 25th." To address this limitation, we propose an extension to the OWL-Time ontology, termed OWL-BOLA, which can represent non-convex intervals.

This section presents the methods adopted for the proposed OWL-BOLA ontology, the environment setup for the model, the dataset tested on the ontology and the performance metrics. Firstly, fundamental concepts and their associated properties were introduced, explaining how the approach extends OWL-Time by enhancing the basic elements to accommodate recurring time intervals, incorporating rules that govern the periodic nature of these intervals (inspired by the work of Faucher et al., 2012), and extending the OWL-Time calendar description to include periodic calendar elements. Subsequently, we delve into the practical implementation of the OWL-BOLA ontology, providing a comprehensive overview of the steps involved and the tools utilized. We also discussed the metrics used in evaluating the study against the state-of-the-art. Lastly, we detail the chosen dataset and explain how it was preprocessed to accommodate the specific objectives of our study.

## Proposed Ontology Modelling

The primary focus of this section is to construct a comprehensive and flexible model tailored to manage intricate recurring temporal concepts. These include primitive periodic entities, periodic rules, and periodic calendar descriptions. For each model, we discuss their classes and properties. The two extensions to OWL-Time are

---

Corresponding author: Yusuf Aminat Bolatito

✉ [aminbolly@gmail.com](mailto:aminbolly@gmail.com)

Department of Information Communication Engineering, Usmanu Danfodio University, Sokoto, Sokoto State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved.



based on a lightweight approach, which focuses on creating simple, easy-to-use classes with minimal complexity. This method ensures that the ontology remains user-friendly while effectively handling the intricacies of non-convex, recurring temporal events. At the beginning of each class in the model, we utilize the OWL-BOLA prefix to indicate our custom ontology and the OWL-Time prefix to denote the base ontology.

### Primitive Periodic Entities

A notable gap in the OWL-Time framework is its lack of representation for non-convex intervals, those with gaps. To address this, we extend the OWL-Time: Interval class by introducing OWL-BOLA: TemporalSeq as shown in Figure 1. This extension allows for the definition of non-convex intervals within the context of calendar date-time, facilitating the capture of recurring events, states, and processes. TemporalSeq introduces the ability to represent non-convex intervals composed of unconnected sub-intervals as shown in Figure 1. TemporalSeq class includes sub-properties to capture various aspects of recurring temporal entities. Given any recurring time concept, we identify it as an instance of TemporalSeq, which can comprise properties such as `hasIntervalTime`, `hasContextTimeRule`, `hasRecurrenceTimeRule`, and `hasExceptionTimeRule` when decomposed into smaller chunks. That is the TemporalSeq can have a boundary linked to the interval through the `hasIntervalTime` property, describing temporal dimensions as a subclass of Interval, making it possible to describe the starts or endings of recurring time concepts.

Subsequently, TemporalSeq can also have a pattern of regular repetition called

PeriodicEntity, introduced via the `hasContextTimeRule` property. PeriodicEntity serves as a superclass for ordered temporal recurring entities, which are further categorized into PeriodicInstant (e.g., every Monday) and PeriodicInterval (e.g., every March to June). These subclasses are designed to capture recurring temporal entities occurring instantaneously or within a duration. The domain of `hasContextTimeRule` is TemporalSeq, with the range defined in the ContextTime class in section 3.1.2. Additionally, TemporalSeq can have a pattern that further describes PeriodicEntity, named PeriodicTimeSpan, introduced via `hasRecurrenceTimeRule`. This sub-property defines the description of periodicity, offering granularity on PeriodicEntity. For example, in the phrase "on the first Wednesday of every month," "every month" is the PeriodicInstant and "first Wednesday" is the PeriodicTimeSpan. The domain of `hasRecurrenceTimeRule` is TemporalSeq, with the range defined in the RecurrenceTime class in section 3.1.2. Furthermore, TemporalSeq can also have irregular patterns in its description, named PeriodicException, introduced via `hasExceptionTimeRule`. This sub-property handles deviations from regular patterns, representing exceptions in recurring cycles. An example is "every day at 5 p.m. except on Sunday at 2 p.m.," where "every day" is the PeriodicInstant, "every 5 p.m." is the PeriodicTimeSpan of the regular PeriodicInstant, and "Sunday at 2 p.m." is the PeriodicException. The domain of `hasExceptionTimeRule` is TemporalSeq, with the range defined in the ExceptionTime class in section 3.1.2.

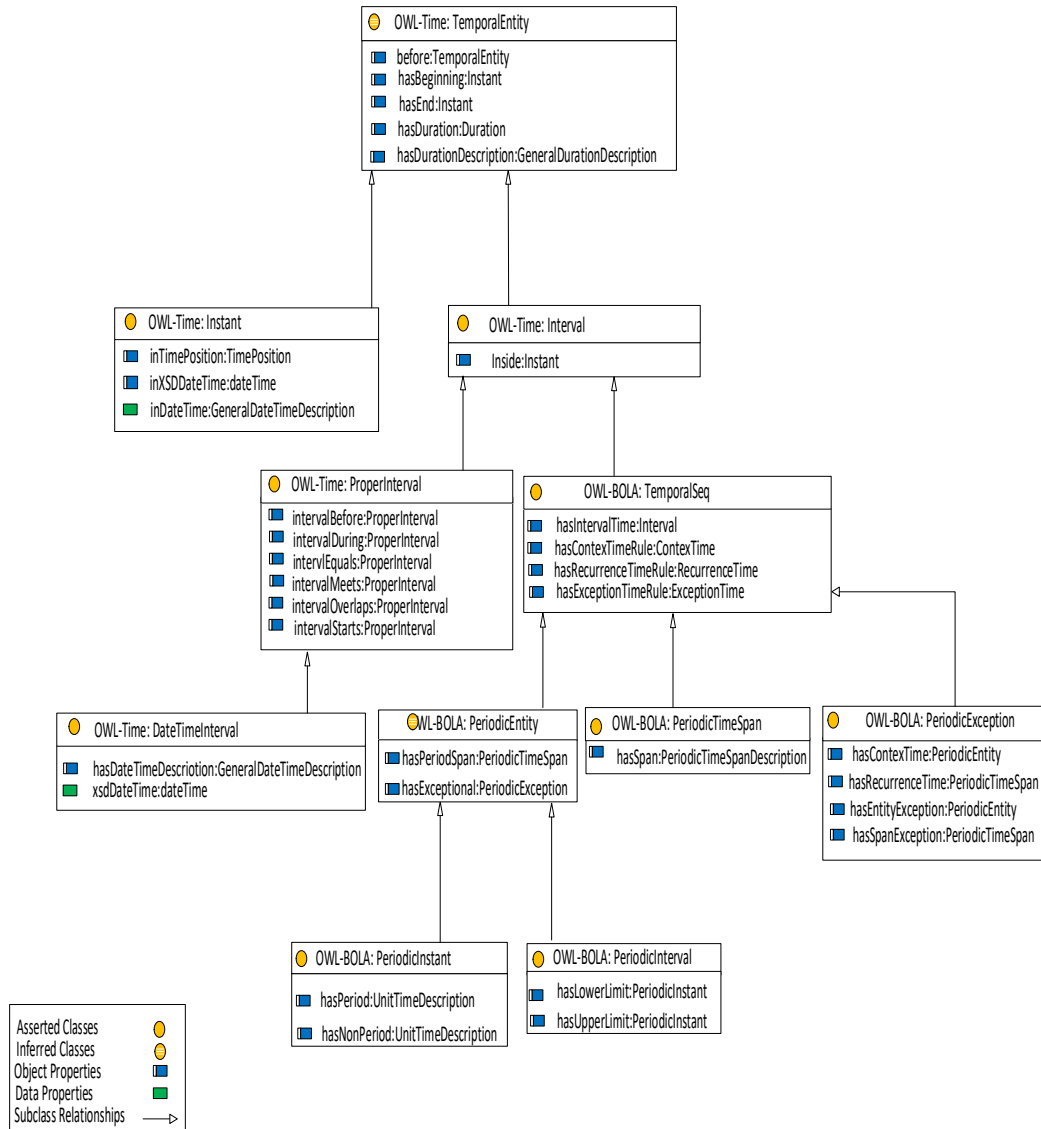
---

Corresponding author: Yusuf Aminat Bolatito

✉ [aminbolly@gmail.com](mailto:aminbolly@gmail.com)

Department of Information Communication Engineering, Usmanu Danfodio University, Sokoto, Sokoto State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved.



**Figure 1: Primitive Periodic Entities Classes**

To further enhance the framework, we introduce two object properties, `hasPeriod` and `hasNonPeriod`, for `PeriodicInstant`, establishing connections between `PeriodicInstant` and the periodic calendar description discussed in section 3.2. Similarly, to establish the relationship between the beginning and end of a `PeriodicInterval`, we employ the object properties

`hasLowerLimit` to define the starting point and `hasUpperLimit` to specify the endpoint. These properties enable the specification of `PeriodicInstants` as the starting or ending points of a periodic interval and facilitate the utilization of `UnitTimeDescription`. The `PeriodicTimeSpan` class uses the `hasSpan` property to associate with the periodic calendar description, serving as a

Corresponding author: Yusuf Aminat Bolatito

✉ [aminbolly@gmail.com](mailto:aminbolly@gmail.com)

Department of Information Communication Engineering, Usmanu Danfodio University, Sokoto, Sokoto State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved.

placeholder for its finest granularity. Detailed descriptions are provided in the `PeriodicTimeSpanDescription` in Section 3.2.

By incorporating these various periodic elements, the expanded OWL-Time framework offers a comprehensive approach to representing and reasoning about complex temporal scenarios, capturing the intricacies of temporal recurrence with greater precision. This enhanced framework not only fills existing gaps but also provides a robust structure for future temporal modeling and reasoning.

### Periodic Rule

In the OWL-BOLA ontology, we introduced the concept of `PeriodicRule` as shown

in Figure 2, which was not originally present in the OWL-Time ontology. This concept represents a sophisticated framework for modeling recurring time patterns, inspired by `iCalendar` but extended to enhance expressiveness and versatility. `PeriodicRule` serves as a pivotal container within OWL-BOLA, aggregating various recurring time patterns into subclasses—`ContextTime`, `RecurrenceTime`, and `ExceptionTime`. These subclasses serve as the range for the domain properties `hasContextTimeRule`, `hasRecurrenceTimeRule`, and `hasExceptionTimeRule`, respectively, as described in Section 3.1.1.

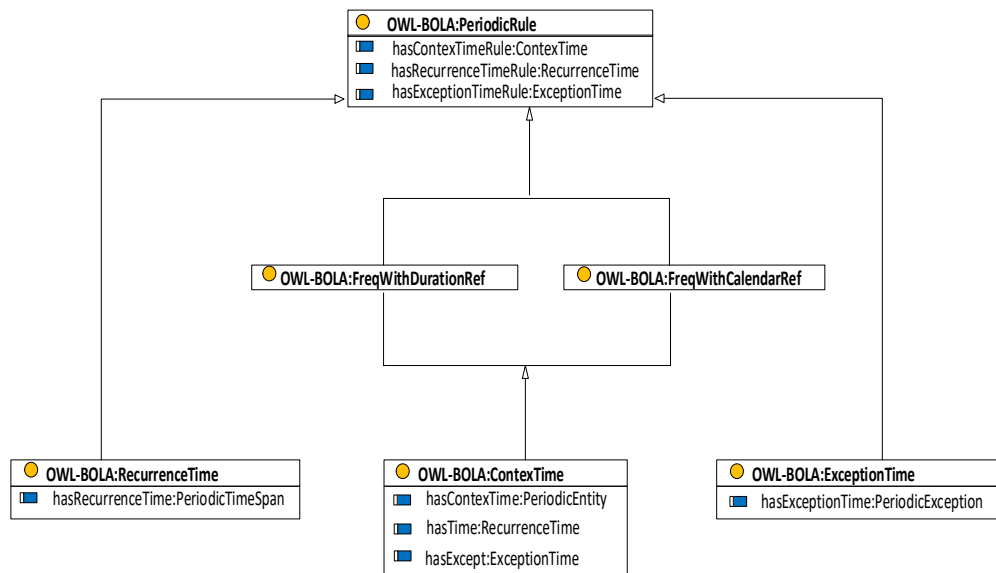


Figure 2: Classes of Periodic Rule

`ContextTime` within `PeriodicRule` defines patterns of periodic temporal distance. It includes patterns like "every week" or "every third month," categorized further into subclasses such as `FreqWithCalendarRef` (e.g., each Wednesday) and `FreqWithDurationRef` (e.g., every two weeks). The `ContextTime` class is linked to the property `hasContextTime`, whose range is `PeriodicEntity`.

`RecurrenceTime` represents patterns of periodic time spans within `ContextTime`. For instance, "three times every month" falls under

`RecurrenceTime`, linking instances to their corresponding periods or time spans. The `RecurrenceTime` class is linked to the property `hasRecurrenceTime`, whose range is `PeriodicTimeSpan`.

`ExceptionTime` addresses patterns of time exclusions or exceptions within `ContextTime`. It captures instances like "every month except April," where certain periods are explicitly excluded from the recurring pattern. `ExceptionTime` establishes relationships between

Corresponding author: Yusuf Aminat Bolatito

✉ [aminbolly@gmail.com](mailto:aminbolly@gmail.com)

Department of Information Communication Engineering, Usmanu Danfodio University, Sokoto, Sokoto State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved.

ContextTime instances and their irregularity rules. The ExceptionTime class is linked to the property hasExceptionTime, whose range is PeriodicException.

Since Figure 1 and Figure 2 are separate, we connected the two concepts—TemporalSeq and PeriodicRule—via the Aggregated class such that the two are parseType collections. This connection enables the reuse of knowledge, integrating TemporalSeq and PeriodicRule seamlessly within the ontology.

### Periodic Calendar Description

In the OWL-BOLA ontology, we expanded OWL-Time calendar descriptions of the GeneralDateTimeDescription class through lightweight ontology engineering techniques. This expansion includes explicit values previously implicitly defined in the primitive periodic entities of Section 3.1.1. The extended model, shown in

Figure 3, incorporates new classes and properties to enhance OWL-Time's capabilities.

OWL-BOLA: UnitTimeDescription handles periodic calendar descriptions for PeriodicInstant, PeriodicInterval, and PeriodicException. It extends OWL-Time: GeneralDateTimeDescription with properties such as unitType, hasGap, and hasList. The unitType property specifies frequency (e.g., unitYear for every year, unitMonth for every month, etc.), hasGap represents periodic entities with numeric values (e.g., every three months with hasGap: 3), and hasList links to ListDescription for specific days or months. OWL-BOLA: ListDescription, a subclass of UnitTimeDescription, links the hasList property with specific calendar values. Its properties include dayOfWeek for recurring days (e.g., Wednesday), day for specific calendar days (1-31), and month for specific months (1-12).

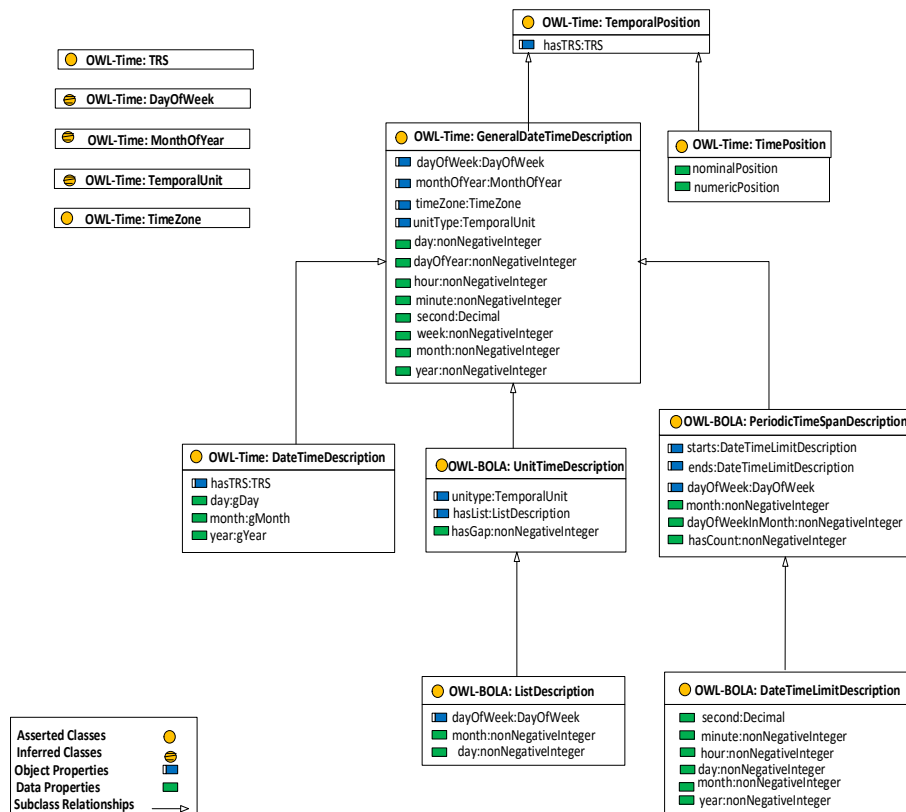


Figure 3: Classes of Periodic Calendar Description

Corresponding author: Yusuf Aminat Bolatito

[aminbolly@gmail.com](mailto:aminbolly@gmail.com)

Department of Information Communication Engineering, Usmanu Danfodio University, Sokoto, Sokoto State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved.

OWL-BOLA: PeriodicTimeSpanDescription specifies features for PeriodicTimeSpan and PeriodicException, extending OWL-Time: GeneralDateTimeDescription to express detailed periodic references. Properties of PeriodicTimeSpanDescription include starts and ends for defining starting and ending points, dayOfWeek for specific days of the week, month for specific months, dayOfWeekInMonth for specific days within a month, and hasCount for the number of occurrences (e.g., twice every month). Lastly, OWL-BOLA: DateTimeLimitDescription provides fine-grained time span representation

using properties down to second, minute, hour, day, month, and year. It supports explicit representation of start and end times in various formats, enabling detailed and precise calendar-based temporal expressions.

**Recurring Time Example:** In order to demonstrate how OWL-BOLA ontology can be used to model recurring time sentence, let's consider Example 1.

Example 1: "The lecture will be held until March 28, 2024, every Wednesday at 11 p.m."

```
1. <lecture> rdf:type TemporalSeq;
2.   hasIntervalTime <untilMarch282024>;
3.   hasContextTimeRule <every_Wednesday>;
4.   hasRecurrenceTimeRule <every_23_Hour>;
5. <untilMarch282024> rdf:type Interval;
6.   hasEnd <march282024>;
7. <march282024> rdf: type GeneralDateTimeDescription;
8.   unitType "unitDay";
9.   day "28";
10.  month "03";
11.  year "2024";
12. <every_Wednesday> rdf: type ContextTime;
13.   hasContextTime <everyWednesday>;
14. <everyWednesday> rdf: type PeriodicInstant;
15.   hasPeriod <periodWednesday>;
16. <periodWednesday> rdf: type UnitTimeDescription;
17.   unitType "unitDay";
18.   hasList <periodListWednesday>;
19. <periodListWednesday> rdf: type ListDescription;
20.   dayOfWeek "Wednesday";
21. <every_23_Hour> rdf: type RecurrenceTime;
22.   hasRecurrenceTime <every23Hour>;
23. <every23Hour> rdf: type PeriodicTimeSpan;
24.   hasSpan <period23Hour>;
25. <period23Hour> rdf: type PeriodicTimeSpanDescription;
26.   starts <23Hour>;
27. <23Hour> rdf: type DateTimeLimitDescription;
28.   hour "23";
```

Figure 4: Detail representation of Example 1

Figure 4 illustrates Example 1 within the OWL-BOLA ontology, encoding a representation of a repeating event in a TemporalSeq with three properties: hasIntervalTime (untilMarch282024) for its interval, hasContextTimeRule (every\_Monday) for its ContextTime, and hasRecurrenceTimeRule (every\_23\_Hour) for its RecurrenceTime.

The property hasEnd from OWL-Time is reused to represent the interval ending on "untilMarch282024," whose resource is then

"march282024." In this context, we extended the instance of "march282024," classifying it under GeneralDateTimeDescription and detailing its attributes: "unitDay" via the unitType property, "28" via the day property, "03" via the month property, and "2024" via the year property.

To encode "every\_Monday," we evoke the concept of ContextTime to denote its repetitive rule, using the property hasContextTime and linking it to a PeriodicEntity, specifically the PeriodicInstant identified as "every\_Wednesday."



For "everyWednesday," we utilize the property `hasPeriod`, linking it to a `UnitTimeDescription` specifying the recurring period, identified as `PeriodWednesday`. We expanded the instance "periodWednesday" within `UnitTimeDescription`, representing its "unitDay" via the `unitType` property. This instance is intricately linked to "periodListWednesday" through the `hasList` property of `UnitTimeDescription`, establishing a connection to `ListDescription`. For "periodListWednesday," we utilized the `dayOfWeek` property to specify it as "Wednesday."

Additionally, the time span of occurrence "every\_23\_Hour" is represented as a type of `RecurrenceTime` repetitive rule. This utilizes the property `hasRecurrenceTime` with the resource "every23Hour" of type `PeriodicTimeSpan`. This `PeriodicTimeSpan` is associated with a `PeriodicTimeSpanDescription` named `period23Hour`. We expanded "period23Hour" to extend to "23Hour" through the `starts` property. The "23Hour" is an instance of `DateTimeLimitDescription`, specifying the hour property as "23."

#### Computational environment setup

In this study, the computational setup was meticulously designed to ensure the robustness of the ontology modeling and reasoning process. For this purpose, Protégé 5.6.3 was used as the primary tool for ontology development. Protégé is a renowned open ontology editor which provides a user-friendly interface and extensive functionality making it a popular choice for researchers. The implementation was carried out on a personal computer with an Intel(R) Core i5-5200U CPU running at 2.20 GHz, 8 GB of RAM, and Windows 8.

Ontologies provide a formalism for specifying concepts or offering an abstract description of a system within domain-specific knowledge. The OWL-BOLA ontology, as presented in section 3.1, is based on the foundational OWL-Time constructs. To maintain consistency, we leveraged the OWL2 ontology language for reasoning, influenced by the improved features of OWL2-DL, which aligns with OWL-Time. The ontology modeling process involved several steps:

- i. Setup of Protégé: We installed Protégé 5.6.3, preparing the environment for creating ontology classes, properties, and axioms.
- ii. Instantiation of Classes: Ontology classes were instantiated within Protégé, reflecting the conceptual entities identified in section 3.1. Each class was defined with respective attributes and relationships, adhering to the model.
- iii. Property Encoding: Properties representing relationships between classes were defined and associated with their corresponding domains and ranges. This ensured the ontology captured the intricate connections between different temporal entities.
- iv. Axiom Encoding: Logical statements and constraints were integrated to formalize the relationships and rules governing the ontology, translating the three separate framework into a practical structure within Protégé.
- v. Visualization Tools: Visualization tools such as OWL Viz and OntoGraf plugins were used to generate graphical representations of the ontology structure, facilitating comprehension and communication.
- vi. Reasoner: The FaCT++ engine was employed to validate and reason over the ontology, ensuring syntactic and semantic correctness, inferring implicit knowledge, capturing relationships between temporal entities, and adhering to standardized OWL syntax and semantics.

#### Dataset Alignment

In this study, we integrate recurring time data from Faucher et al. (2012) into our ontological framework. Using the open-source natural language processing library SpaCy and the Protégé platform, we meticulously navigated through data preparation, correspondence identification, and integration with Protégé.

- i. Data Preparation and Structure: We acquired 513 anonymized datasets from Faucher et al. (2012), translated them from French to English using Google Translator, and normalized the dataset to align with our

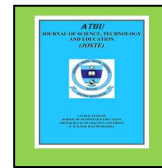
---

Corresponding author: Yusuf Aminat Bolatito

✉ [aminbolly@gmail.com](mailto:aminbolly@gmail.com)

Department of Information Communication Engineering, Usmanu Danfodio University, Sokoto, Sokoto State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved.



- ontology. Each statement was transformed into a list of POS-tagged word-tokens.
- ii. Identifying Correspondence and Annotation: We normalized and annotated key POS-tagged word-tokens in the dataset. Complex temporal phrases were structured into TemporalSeq representations. Numerical lexemes were standardized into specific formats, and each unit received comprehensive annotation to facilitate better understanding and integration with our ontology.
  - iii. Integration with Protégé: We established relationships between dataset lexemes and ontology classes, properties, or instances. This involved invoking corresponding OWL-BOLA or OWL-Time classes and properties based on the identified characteristics in the sentences. By doing so, we ensured that the dataset was thoroughly integrated with the OWL-BOLA ontology, enabling effective data alignment and ontological reasoning

#### Performance Metrics

The research presented in this paper significantly enhances the performance of the

OWL-Time ontology. To evaluate the effectiveness and significance of our knowledge representation, we compared the richness of our proposed ontology with OWL-Time and other related ontologies that have extended OWL-Time constructs using the complexity metrics. Complexity richness is the metric gauges the detailed and intricate organization and components of the ontology. A more complex ontology provides a more comprehensive and nuanced representation of domain knowledge. It includes the number of classes, number of object properties, number of individuals, depth of class hierarchy, number of axioms, number of logical axioms, number of role chains, number of equivalent classes, number of disjoint classes, and number of inferred subclass relationships.

#### EXPERIMENTAL RESULTS

In this study, an experiment was conducted on a recurring time dataset comprising 513 sentences from the work of Faucher et al. (2012) using Protégé. The performance of OWL-Time and the work of Bento et al. (2017) was evaluated and compared in terms of complexity richness. Table 1 highlights the richness.

**Table 1:** A Comparative Summary of the Metrics and Their Respective Counts in the State-of-the-Art Ontology

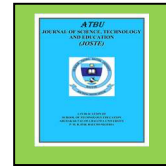
Complexity Richness				
Temporal Ontology				
	OWL-Time	(Bento et al., 2017)	Proposed Ontology (OWL-BOLA)	
Metrics	Count			Descriptions
Number of Classes	55	89	118	measures the overall count of classes defined within the ontology
Number of Object Properties	33	51	55	Assesses the total count of object properties specified within the ontology.
Number of Individuals	27	95	145	assesses the total number of instances present in the ontology

Corresponding author: Yusuf Aminat Bolatito

✉ [aminbolly@gmail.com](mailto:aminbolly@gmail.com)

Department of Information Communication Engineering, Usmanu Danfodio University, Sokoto, Sokoto State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved.



Depth of class hierarchy	4	2	6	evaluates the maximum depth of the inheritance chain within the class hierarchy
Number of axioms	208	386	540	evaluates the total count of axioms in the ontology, encompassing subclass relationships, property assertions, and constraints
Number of logical axioms	188	258	308	Counts the logical relationships and constraints, excluding annotations and metadata.
Number of role chains	7	8	10	Counts the sequences of object properties defined in the ontology.
Number of equivalent classes	4	5	6	measures the number of equivalent class axioms that define classes with the same meaning.
Number of disjoint classes	1	1	4	measures the number of disjointness axioms that state classes have no common instances.

The proposed ontology was evaluated using atomic metrics, such as class hierarchy depth, number of classes, object properties, axioms, and role chains, to assess its complexity and richness. The evaluation results indicate the ontology's extensive knowledge representation in the domain. For instance, the class hierarchy depth of our ontology is 6, compared to 2 for Bento et al. (2017) and 4 for OWL-Time. Our ontology's performance improved because we enhanced OWL-Time with additional features that capture the intricacies of recurring time, whereas Bento et al. (2017) focused only on a single problem in the domain. Similar improvements were observed in

other atomic metrics, underscoring the richness of our ontology.

## CONCLUSION

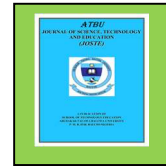
The primary objective of this study is to advance the field of temporal representation and reasoning by enhancing the OWL framework to effectively capture recurring temporal information in domain-specific contexts. This study introduced a novel ontology, the OWL-BOLA ontology, which extends the OWL-Time framework to better represent recurring time information. OWL-BOLA incorporates primitive periodic entities, periodic rules, and extends the OWL-Time calendar description to include periodic calendar

Corresponding author: *Yusuf Aminat Bolatito*

✉ [aminbolly@gmail.com](mailto:aminbolly@gmail.com)

Department of Information Communication Engineering, Usmanu Danfodio University, Sokoto, Sokoto State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved.



descriptions, ensuring comprehensive modeling of recurring time concepts and relations.

The effectiveness of OWL-BOLA was validated using real-world datasets and ontology metrics of complexity. The results demonstrated that the ontology provides a robust and practical solution for representing complex recurring temporal information, showing significant improvements over previous works like Bento et al. (2017).

However, OWL-BOLA has certain limitations, such as the lack of patterns for representing event individuals with dynamic statuses. Future work aims to enhance representation flexibility and incorporate automated annotation techniques using NLP.

## REFERENCES

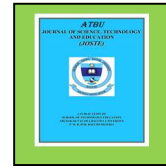
- Abbasimehr, H., & Bagheri, F. S. (2022). A novel time series clustering method with fine-tuned support vector regression for customer behavior analysis. *Expert Systems with Applications*, 204, 117584.
- Baji, O., Ben Said Amrani, N., & Sarsri, D. (2022). Online trajectory tracking control based on the explicit form of the equations of motion for serial manipulator using the new formulation. *Journal of Robotics*, 2022(1), 8715161.
- Batsakis, S., & Petrakis, E. G. (2011). SOWL: a framework for handling spatio-temporal information in OWL 2.0. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web* (pp. 242-249). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Batsakis, S., Petrakis, E. G. M., Tachmazidis, I., & Antoniou, G. (2017). Temporal representation and reasoning in OWL 2. *Semantic Web*, 8(6), 981-1000. <https://doi.org/10.3233/SW-160248>
- Baumann, R., Loebe, F., & Herre, H. (2014). Axiomatic theories of the ontology of time in GFO. *Applied Ontology*, 9(3-4), 171-215. <https://doi.org/10.3233/AO-140136>
- Bento, A. M., Viegas, D. C., & Correia, N. (2017). Temporal Reasoning with Non-convex Intervals. In P. Rózewski & C. Lange (Eds.), *Knowledge Engineering and Semantic Web* (Vol. 786, pp. 127-142). Springer International Publishing. [https://doi.org/10.1007/978-3-319-69548-8\\_10](https://doi.org/10.1007/978-3-319-69548-8_10)
- Cardoso, S. D., Da Silveira, M., & Pruski, C. (2020). Construction and exploitation of an historical knowledge graph to deal with the evolution of ontologies. *Knowledge-Based Systems*, 194, 105508.
- Faucher, C., Lafaye, J.-Y., & Bertrand, F. (2012). *Modelling composite periodic Events*. 15.
- Fikes, R., & Zhou, Q. (2002). A Reusable Time Ontology. *Proceeding of the AAAI Workshop on Ontologies for the Semantic Web*, 1-6.
- Ghorbel, F., Hamdi, F., Métais, E., Ellouze, N., & Gargouri, F. (2019). Ontology-based representation and reasoning about precise and imprecise temporal data: A fuzzy-based view. *Data & Knowledge Engineering*, 124, 101719. <https://doi.org/10.1016/j.datak.2019.10.1719>
- Gu, J., Wang, D., Hu, D., Gao, F., & Xu, F. (2023). Temporal Extraction of Complex Medicine by Combining Probabilistic Soft Logic and Textual Feature Feedback. *Applied Sciences*, 13(5), 3348. <https://doi.org/10.3390/app13053348>
- Hobbs, J. R., & Pan, F. (2004). An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing*, 3(1), 66-85. <https://doi.org/10.1145/1017068.1017073>
- Hobbs, J. R., Pan, F., Cox, S., & Little, C. (Eds.). (2017). *Time Ontology in OWL*. W3C working draft, W3C. <https://www.w3.org/TR/owl-time/>
- Hu, D., Wang, M., Gao, F., Xu, F., & Gu, J. (2022). Knowledge Representation and

Corresponding author: Yusuf Aminat Bolatito

✉ [aminbolly@gmail.com](mailto:aminbolly@gmail.com)

Department of Information Communication Engineering, Usmanu Danfodio University, Sokoto, Sokoto State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved.



- Reasoning for Complex Time Expression in Clinical Text. *Data Intelligence*, 4(3), 573–598. [https://doi.org/10.1162/dint\\_a\\_00152](https://doi.org/10.1162/dint_a_00152)
- Ijntema, W., Goossen, F., Frasinca, F., & Hogenboom, F. (2010, March). Ontology-based news recommendation. In *Proceedings of the 2010 EDBT/ICDT Workshops* (pp. 1-6).
- Lenat, D. B., & Guha, R. V. (1991). The evolution of CycL, the Cyc representation language. *ACM SIGART Bulletin*, 2(3), 84–87. <https://doi.org/10.1145/122296.122308>
- Li, F., Du, J., He, Y., Song, H.-Y., Madkour, M., Rao, G., Xiang, Y., Luo, Y., Chen, H. W., Liu, S., Wang, L., Liu, H., Xu, H., & Tao, C. (2020). Time event ontology (TEO): To support semantic representation and reasoning of complex temporal relations of clinical events. *Journal of the American Medical Informatics Association*, 27(7), 1046–1056. <https://doi.org/10.1093/jamia/ocaa058>
- Madkour, M., Benhaddou, D., & Tao, C. (2016). Temporal data representation, normalization, extraction, and reasoning: A review from clinical domain. *Computer Methods and Programs in Biomedicine*, 128, 52–68. <https://doi.org/10.1016/j.cmpb.2016.02.007>
- Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., & Schneider, L. (2003). Dolce: a descriptive ontology for linguistic and cognitive engineering. *WonderWeb Project, Deliverable D17 v2*, 1, 75-105.
- O'Connor, M. J., & Das, A. K. (2011). A Method for Representing and Querying Temporal Information in OWL. In A. Fred, J. Filipe, & H. Gamboa (Eds.), *Biomedical Engineering Systems and Technologies* (Vol. 127, pp. 97–110). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-18472-7\\_8](https://doi.org/10.1007/978-3-642-18472-7_8)
- Pan, F. (2007). *An Ontology of Temporal Concepts for the Semantic Web and Natural Language*. 66.
- Pease, A., Niles, I., & Li, J. (2002, July). The suggested upper merged ontology: A large ontology for the semantic web and its applications. In *Working notes of the AAAI-2002 workshop on ontologies and the semantic web* (Vol. 28, pp. 7-10).
- Poveda, V. M., Suárez, F. M. C., & Gómez, P. A. (2014). *A Pattern for Periodic Intervals*. 1–10.
- Raimond, Y., & Abdallah, S. (2007). The event ontology.
- Siebra, C. A., & Wac, K. (2022). Engineering uncertain time for its practical integration in ontologies. *Knowledge-Based Systems*, 251, 109152. <https://doi.org/10.1016/j.knosys.2022.109152>
- Tao, C., Wei, W.-Q., Solbrig, M. H. R., Savova, G., & Chute, C. G. (2010). CNTRO: A Semantic Web Ontology for Temporal Relation Inferencing in Clinical Narratives. 7, 27.
- Zhang, D., Li, M., Tian, D., Song, L., & Shen, Y. (2022). Intelligent text recognition based on multi-feature channels network for construction quality control. *Advanced Engineering Informatics*, 53, 101669.

---

Corresponding author: Yusuf Aminat Bolatito

✉ [aminbolly@gmail.com](mailto:aminbolly@gmail.com)

Department of Information Communication Engineering, Usmanu Danfodio University, Sokoto, Sokoto State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved.