IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 20, 2020, accepted July 8, 2020, date of publication July 13, 2020, date of current version July 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3008688

stRDFS: Spatiotemporal Knowledge Graph Modeling

LIN ZHU, NAN LI, LUYI BAI, YUNQING GONG, AND YIZONG XING
School of Computer and Communication Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China

Corresponding author: Luyi Bai (baily@neuq.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61402087, in part by the Natural Science Foundation of Hebei Province under Grant F2019501030, in part by the Natural Science Foundation of Liaoning Province under Grant 2019-MS-130, in part by the Key Project of Scientific Research Funds in Colleges and Universities of the Hebei Education Department under Grant ZD2020402, and in part by the Fundamental Research Funds for the Central Universities under Grant N2023019.

ABSTRACT In Semantic Web, modeling knowledge graph based on RDF becomes more and more popular. There is quite a lot of spatiotemporal information in Semantic Web, and recent works focus on not only general data but also spatiotemporal data. Existing efforts are mainly to add spatiotemporal labels to RDF, which expand RDF triple into quad or quintuple. However, extra labels often cause additional overhead for the system and lead to inefficient information organization management. In order to overcome this limitation, we propose an stRDFS model by labeling properties with spatiotemporal features and the corresponding determination methods of topological relations among different spatiotemporal entities. stRDFS considers spatiotemporal attribute as a part of the RDF model, which can record spatiotemporal information without changing the current RDF standard. Our approach improves the ability of recording and linking spatiotemporal data. More importantly, depending on formatting of spatiotemporal attributes in stRDFS, it will improve the semantic inferring ability, and the users are not required to be familiar with the underlying representations of spatiotemporal data.

INDEX TERMS Knowledge graph, spatiotemporal data, stRDFS.

I. INTRODUCTION

With the prompt development of the Internet, knowledge graph is rapidly emerging, which contributes a lot to the knowledge organization and intelligent application on the Internet [39], [40], and has significant meanings for artificial intelligence. Knowledge graph [56] is a knowledge base that represents objective concepts/entities and their relationships in the form of graph. It constitutes a huge semantic network diagram where nodes represent entities or concepts, and edges are composed of attributes or relationships.

In Semantic Web, entities can be mainly divided into general entities and spatiotemporal entities. As for general data, the knowledge graph can be expressed as RDF [10], [19], [29]. RDF (Resource Description Framework), a language proposed by the World Wide Web Consortium (W3C), which can express the semantics of knowledge graph formally. RDF is inclusive, exchangeable and easy to extend, control and integrate in data processing, so Tong [43] maps object-oriented database models into RDF and Takama and Hattori [41] study the mining association rules for adaptive search engine based on RDF. For spatiotemporal data, there have been some achievements in representing spatiotemporal entities, such as temporal data model, spatial data model and spatiotemporal data model.

In the representation of temporal data, some researchers try to take temporal data as linked data and establish the corresponding temporal models [14], [21], [24], [34], [37]. For example, Lutz *et al.* [34] study the TDL (Temporal Description Logic) model representing temporal entities and temporal description logics. However, TDL model is not compatible well with the current mainstream Semantic Web editing tools, which leads to the inability to be widely used in the representation of large-scale temporal information. Batsakis and Petrakis [5] establish standards of the Semantic Web and the 4D-fluents approach for representing the evolution of temporal information in entities. Based on the 4D-fluents approach, the 4-Fluents Plug-In (a tool for handling temporal ontology in Protégé) has been proposed. But the complexity and inflexibility of the 4-Fluents plug-in

The associate editor coordinating the review of this manuscript and approving it for publication was Manik Sharma.

bring great challenges to temporal semantics and data self-updating. Peng *et al.* [37] explore a general relation extraction framework based on graph LSTMs (graph long short-term memory networks). It can be easily extended to cross-sentence n-ary temporal relation extraction. Unfortunately, the complexity of temporal relations makes this model weak, so the model is unsuitable for recording large-scale temporal data. There are also some unstructured models of temporal information. For instance, SWRL [20] (The Semantic Web Rule Language) provides a standard method for representing temporal data. Watkins and Nicole [52] demonstrate the use of named graphs. In [42], Tappolet *et al.* present a syntax and storage format based on named graphs to express temporal RDF. However, researchers seldom pay attention to the extension of RDF model to represent temporal entities. Since all entities and relations can link to each other by labels, Hernández *et al.* [23] add temporal qualifiers and values to the RDF model representation in the form of labels. Then RDF triples are expanded into a quintuple (s, p, o, q, v), where (s, p, o) refers to the primary relation, q is the temporal qualifier property, and v is the temporal qualifier value. Besides, there are some models that label properties with the time interval [22], [38].

In addition to temporal data, prior works also focus on the representation of spatial data [1], [15], [26]. Spatial information in the RDF data model is usually represented as serializations of geometries accompanied with a Coordinate Reference System (CRS). In CRS, it defines how to relate these serializations to real geometries on the surface of Earth. W3C GEO [32], an RDF vocabulary, can represent simple location information in RDF. W3C GEO provides the basic terminology for serializing point geometries. It represents latitude, longitude and other information about spatially-located things by a namespace. GeoRDF [8], [9], [13] is an RDF compatible profile for geometric information (points, lines, and polygons) and can be used for representing any point on the earth. GeoMetadataOverSvg is a geographic information notation for GeoRDF, which plays a significant role in the spatial data connection in the Semantic Web. Unfortunately, GeoMetadataOverSvg can only represent three-dimensional geospatial metadata and it fails to link to temporal data. Then, Batsakis and Petrakis [6], [7] put forward SOWL, which Builds upon well established standards of the semantic Web and the 4D-fluents approach for representing the evolution of temporal information in ontologies. SOWL illustrates how spatial and spatiotemporal information and evolution in space and time can be efficiently represented in OWL.

With the development of spatial models and temporal models, some researchers try to combine spatial data with temporal data to form spatiotemporal data, and establish the corresponding models [25], [28], [31], [33], [50], [54]. Among them, the researchers take fuzziness [33] into account, and also focus on cloud detection [54] and practical application [31]. The most mature spatiotemporal models are the YAGO2 [25], gst-Store [50] and stRDF [28], the first two expand RDF triple into quintuple and the last one is quad. In fact, both quintuple and quad method may bring additional overhead to the system, leading to inefficient connection of large amounts of spatiotemporal data. This paper focuses on stRDF model, a structured spatiotemporal RDF developed from the spatial model GeoRDF, which is proposed by Koubarakis *et al.* [28] and extended on the RDF. However, stRDF model is inflexible that it is weak in recording dynamically changing multistate data. When the knowledge graph is updating, the changes of spatiotemporal attribute values cannot be captured in time. Even worse, it is also not good at dealing with flexible relations among spatiotemporal entities. Thus, it is important to improve the model to record changing data and relations, and capture changes of massive dynamic spatiotemporal attribute values. Motivated by such an observation, this paper aims to provide a spatiotemporal knowledge graph model on the basis of RDF without changing current RDF standard. Our solution relies on the effort of the RDF triple to record spatiotemporal data. In this case, we can divide the entities into spatiotemporal entities and non-spatiotemporal entities according to attribute names. Spatiotemporal entities describe the evolution of relations and objects in spatiotemporal dimensions, and non-spatiotemporal entities describe static relations and objects without temporal attributes, spatial attributes or spatiotemporal attributes. In order to describe the relations between different spatiotemporal entities, we construct the determination methods of topological relations in stRDFS.

To summarize, the contributions of this paper are the following:

- We propose a spatiotemporal RDF model, called stRDFS, which can record spatiotemporal data without changing the current RDF standard. In the mean time, it solves the problem of synchronous updates successfully.
- We explore the determination methods of topological relations between different spatiotemporal entities.

The rest of the paper is organized as follows. We introduce the related work in Section 2. The modeling approach is proposed in Section 3. Section 4 is dedicated to the determination methods of topological relations, Section 5 makes a comparison and Section 6 concludes the paper.

II. RELATED WORK

The recent research results presented in this section mainly include spatiotemporal data models based on RDF, such as temporal data model, spatial data model and spatiotemporal data model.

A. TEMPORAL DATA MODEL ON RDF

When the temporal concept is not included in many Semantic Web tools and techniques, the most important step is to identify models that can introduce time. These temporal models can be roughly divided into two categories. The first category is an ontology method which obtains no temporal data from the users. Temporal concept and temporal relations are later added to the ontology and these operations are transparent to the users, for example, Wangni [51] adopts this model. The second category is that the users create temporal entities rather than adding them later. This approach is adopted in most models and will be discussed in detail in the following.

TDL (Temporal Description Logic) model [34] combines standard DLs with temporal data. It is a relatively primitive model whose main operators are 'since', 'until', 'always in the past', 'sometime in the future' and 'in the next moment in future'. Ho *et al.* [24] study how to represent fuzzy temporal data based on this model. Unfortunately, TDL is not compatible well with the current mainstream Semantic Web editing tools. Within a time interval, the entities of 4D-Fluents [5], [17], [30] are represented by the temporal part of the entity. However, in practical applications, the model is not only complicated but also inflexible. The N-ary relation [21], [35], [37] suggests properties of two object and a new object which occurs during time intervals. Compared to other methods, this method results in the smallest time unit being useless and the structure is complicated. Named graph [14] is a subgraph of the ontology RDF attributes graph, which can be specified by distinct names. However, many platforms do not support this model. In [16], the SWRL provides a standard method for representing temporal data. Then Wlodarczyk *et al.* [53] develop SWRL-F based on the above rules to describe fuzzy temporal data and relations. Besides, Treur [45] proposes an approach on the basis of Reification. Reification can represent the N-element temporal relationship, but its semantic express ability is limited and scalability is not strong. There are also some other models that use temporal tags to extend RDF. For example, Gutierrez *et al.* [22] and Pugliese *et al.* [38] introduce time into RDF. In [23], temporal qualifiers and values are added to the RDF model. The RDF triple is expanded into a quintuple (s, p, o, q, v), where s denotes the subject, p represents the predicate, o expresses the objects, q depicts a temporal qualifier property, and v is a temporal qualifier value. Yet the quintuple or quad representation method may bring additional overhead to the system, leading to inefficient connection of large amounts of temporal data.

With improvement of theory and advancement of technology, temporal RDF models are still developing. The spatiotemporal model proposed in this paper absorbs the advantages of above models in temporal data representation, improves scalability and self-renewal ability of the model, and contributes to correlate a large amount of temporal data.

B. SPATIAL DATA MODEL ON RDF

In order to represent the spatial information (e.g. longitude, latitude and altitude) of the entities, several models improved on traditional RDF are proposed. Among spatial models, one of the most mature developments is GeoRDF. Therefore, we will focus on GeoRDF [8], [9], [13] in this section.

RDFIG defines a simple vocabulary for expressing points on the earth in WGS84 form. Based on the spatial vocabulary, a more mature model GeoRDF is established. GeoRDF defines three main classes: *geo: SpatialObject*, *geo: Feature* and *geo: Geometry*. At present, the GeoRDF model has been applied to the project and has corresponding processing platforms and tools. For example, the project LinkedGeoData[1] focuses on publishing OpenStreetMap[2] data as linked data. Besides, Sparqlify[3] has taken advantage of the GeoRDF. Toward the geospatial information resides in a spatially enabled relational database, Geometry2RDF[4] is the first tool to allow users to convert geospatial data into an RDF graph. In addition to above tools, TripleGeo[5] is developed in the GeoKnow.[6] The GeoRDF model achieves remarkable results in the representation of spatial data and can record various points on the earth. At the same time, the tools that support GeoRDF can absorb different forms of data sets and have been widely used.

C. SPATIOTEMPORAL DATA MODEL ON RDF

An ever-increasing number of real-life applications produce spatiotemporal data that record the position of moving objects. So some researchers are focusing on spatiotemporal data modeling. For example, Chang *et al.* [11] propose a temporo-spatial model on the basis of MML and software framework, which encourages reusability, sharing and storage.

At present, spatiotemporal data are usually provided as relational tables [27] or XML documents [4], which can be mapped into the RDF data model using R2RML [46]. R2RML is a standard language that allows defining customized mappings from relational databases to RDF datasets. In [46], data are spatiotemporal in nature and R2RML can produce spatiotemporal Linked Open Data. Data generated in this way are used to populate a SPARQL endpoint. This endpoint is implemented using Strabon, a spatiotemporal RDF triple store built by extending the RDF store Sesame. Di *et al.* [18] combine spatiotemporal information with RDF and present a novel representation model of spatiotemporal RDF. Besides, in order to study the efficient spatiotemporal RDF query processing, Vlachou *et al.* [48] represent spatiotemporal data in RDF and store it in knowledge bases with the following notable features: (a) the data is dynamic, since new spatiotemporal data objects are recorded every second, and (b) the size of the data is vast and can easily lead to scalability issues. As a result, this raises the need for efficient management of large-scale, dynamic, spatiotemporal RDF data. There are also some studies on modeling uncertain spatiotemporal data based on RDF [49].

D. ALGEBRA OF RDF GRAPHS

In order to represent the spatiotemporal information of the entities such as longitude, latitude and altitude, several extended models of traditional RDF are proposed [2], [3],

[1] http://linkedgeodata.org
[2] https://www.openstreetmap.org
[3] http://aksw.org/Projects/Sparqlify.html
[4] https://github.com/boricles/geometry2rdf/tree/master/Geometry2RDF
[5] https://github.com/GeoKnow/TripleGeo
[6] https://web.imsi.athenarc.gr/redmine/projects/geoknow_public

[13]. stRDF [3], [28], [36] is the most mature development among this works and we will introduce it in this section.

Although the GeoRDF model is widely used for the expression of geographic information, since the GeoRDF dataset does not have temporal properties, it cannot represent spatiotemporal data. Now, there are some studies on querying temporal RDF [55]. Therefore, the link of spatiotemporal data has attracted the attention of many scholars. For example, Cheng and Ma [12] propose a kind of fuzzy spatiotemporal description logic. Nikitopoulos *et al.* [36] explore distributed spatiotemporal RDF queries on Spark. Besides, the stRDF model [28], [44], [47] has been proposed and established. stRDF model is an extension of the W3C standard RDF. It can represent geospatial data that changes over time. The stRDF model has been recognized by most scholars in the representation of spatiotemporal data. Although the stRDF model seems to be perfect, there are still many problems. For example, stRDF uses spatiotemporal datasets to record entities whose spatiotemporal attributes are changing with time and space. However, there is no one-to-one correspondence between spatiotemporal data and spatiotemporal attributes, which may result in a large number of spatiotemporal relationships expressing ambiguity. Due to the above features, the stRDF model can only link a small amount of spatiotemporal data and cannot record changing data accurately.

In this paper, we propose an stRDFS model that solves the above two problems: The first point is to establish the correspondence between attributes and spatiotemporal entities. The second point is that the stRDFS model can capture varying spatiotemporal attribute values. In particular, we establish a class set that describes spatiotemporal attributes for the stRDFS model.

III. DATA MODEL

In this section, we will propose a spatiotemporal model based on RDF. In order to distinguish it with stRDF model, we will propose stRDFS data model which is good at recording dynamically changing data and flexible relations of spatiotemporal data at first. Then, we will define the main classes of stRDFS and describe them.

For the description of spatiotemporal data, there has been stRDF [28] quad (s, p, o, τ), where s represents subject, p represents property name, o represents spatial object with spatial data, and τ represents temporal data. For example, the stRDF graph is shown in Figure 1 which represents the temporal data and spatial data of mobile receivers. In the graph, the attribute *HasGeometry* represents geometric coordinates of the receiver position; the attribute *TimeSlice* represents temporal data of signal which is propagated by the receiver. As shows in Figure 1, temporal data of stRDF model is stored together in the form of a set, such as $T = \{[8t, 15t], 17t\}$, which causes that temporal data is difficult to correspond to a certain part of the object. At the same time, $T = \{[8t, 15t], 17t\}$ has no correspondence with spatial position, that is, cannot describe temporal data of the receiver's signal at a certain location, resulting in the fuzzy

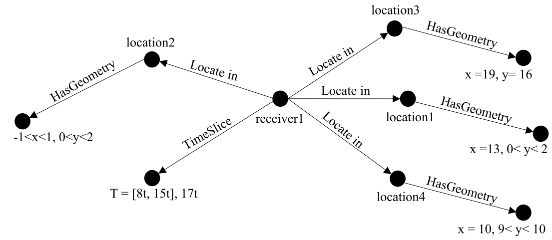


FIGURE 1. The stRDF graph of the mobile receiver.

information indicating time and space. When we use stRDF model to describe changing relations among spatiotemporal entities, spatial data and temporal data are separated, which leads to the uncertainty of information representation. To solve this problem, this paper extends RDF model and proposes stRDFS model. stRDFS model establishes the relations between spatial data and temporal data of the same object, and better describes the changing spatiotemporal state in four-dimensional space.

A. THE stRDFS DATA MODEL

By extending RDF data model (s, p, o), stRDFS model can be formed. The specific representation of stRDFS is defined as follows:

Definition 1: Given a URI set R, empty vertex set B, text description set K, temporal data set I and spatial data set S, an stRDFS expression is $g\ (s, p{:}\ {<}t, l{>}, o)$ where:

- s is a resource name and $s \in R \cup B$.
- p is a property name and $p \in R$.
- o is a value and $o \in R \cup B \cup K \cup I \cup S$.
- $t \in I$ is temporal data.
- $l \in S$ is spatial data.

In Definition 1, stRDFS uses attributes to associate temporal data with spatial data, linking attributes with spatiotemporal data to form spatiotemporal attributes p. When spatiotemporal data changes, spatiotemporal attributes associated with them will change as well. Therefore, the upper layer processing mechanism only needs to perceive the changes of spatiotemporal attributes instead of understanding the changes of spatiotemporal data accurately.

Definition 2: A mapping from x to y is denoted as f_{x-y}, where x represents s or p and y represents t, l or o.

For example, f_{s-o} represents the mapping relationship from s to o and f_{s-t} is the mapping relationship from s to t.

Definition 3: Given an stRDFS expression $g\ (s, p{:}\ {<}t, l{>}, o)$, $U = \{f_{s-o}, f_{s-t}, f_{s-l}, f_{p-t}, f_{p-l}\}$ is a mapping set of g.

In Definition 3, f_{s-o} represents the mapping, whose function is expressed as attribute name, from s to o. The mapping f_{s-t} indicates that s has linked with temporal data, and the formed triple is (s, p, t). In (s, p, t), the property represents "temporal information" and the attribute value t represents temporal data. The mapping f_{s-l} indicates that s has linked with spatial data and the expression is (s, p, l). In the triple, the attribute represents "spatial information" and the attribute value l represents spatial data. The f_{p-t} indicates that p has linked with temporal data and combines with other mappings to form an stRDFS tuple. When f_{p-t} is combined with f_{s-o}, the tuple is formed as (s, p: t, o), indicating that the temporal data describes the valid time of (s, p, o). When f_{p-t} is combined with f_{s-t}, the formed tuple is (s, p: t_2, t_1), indicating that the valid time of s is t_1, and the valid time of the tuple (s, p, t_1) is t_2. When f_{p-t} is combined with f_{s-l}, a tuple (s, p: t, l) is formed, indicating that s has linked with spatial data l, and the valid time of tuple (s, p: t, l) is t. The mapping f_{p-l} represents that p has linked with spatial data and combines with other mappings to form an stRDFS tuple. When f_{p-l} is combined with f_{s-o}, the tuple is formed as (s, p: l, o), indicating that the spatial data l describes (s, p, o). When f_{p-l} is combined with f_{s-t}, the formed tuple is (s, p: l, t), indicating that the valid time of s is t, and the spatial data of (s, p, t) is l. When f_{p-l} is combined with f_{s-l}, a tuple (s, p: l_2, l_1) is formed, indicating that s has linked with spatial data of l_1, and spatial data of (s, p, l_1) is l_2. The mapping f_{p-o} is illegal. The f_{o-t} and f_{o-l} logically represent temporal data and spatial data of o, respectively. In the stRDFS structure, f_{o-t} and f_{o-l} are converted into f_{s-t} and f_{s-l}, and they can appear as tuple mappings alone. For instance, (s_1, p, o: t) can be converted into two tuples (s_1, p_1, o) and (s_2, p_2, t), where $s_2 = o$ and p_2 represents "temporal information". Similarly, (s_1, p, o: l) can be converted into two tuples (s_1, p_1, o) and (s_2, p_2, l), where $s_2 = o$ and p_2 represents "spatial information".

We will give the definition of stRDFS graph in the following. Before this, we define the value range at first.

Definition 4: The value range of the mapping f_{x-y} is denoted as *Range* (f_{x-y}) where *Range* (f_{x-y}) = y.

Definition 5: Given an stRDFS expression g (s, p: <t, l>, o), an stRDFS graph for g is a labeled graph G (V, E, F, λ, T, L) where:

- $V = s \cup Range(U)$ is the set of vertices.
- $E = \{(r, r')\}$ is the set of edges from r to r' where $\forall r$, $r' \in V$
- $F\ (r, r') = \{f \mid (r, f\!: <t, l>, r') \in G\}$ is the mappings set of E where $\forall r, r' \in V*$.
- λ is the set of labels given by vertices or edges.
- $T \in Range\ (f_{s-t} \cup f_{p-t})$.
- $L \in Range\ (f_{s-l} \cup f_{p-l})$.

According to Definition 5, there are two cases: the first one is that stRDFS graph vertices contain spatiotemporal information, in this case $T \in Range(f_{s-t})$ and $L \in Range$ (f_{s-l}), as shown in Figure 2(a), the second is that the stRDFS graph edges contain spatiotemporal information, in this case $T \in Range\ (f_{p-t})$ and $L \in Range\ (f_{p-l})$, as shown in Figure 2. The expression of T and L will be defined in the following.

Definition 6: Given an stRDFS expression g (s, <p: <t, l>, o) and (v_i, f_{vi-vj}, v_j) $\in g$, the temporal data of (v_i, f_{vi-vj}, v_j) is represents by $Ti = f\ (N, k)$ where:

- $N = [t_{s(f)}, t_{e(f)}]$ is valid time where $t_{s(f)}$ represents start time of mapping f_{vi-vj}, and $t_{e(f)}$ represents terminal time of mapping f_{vi-vj}.
- $k = t_{r(f)}$ is reference time recorded as now.

In Definition 6, N represents the valid time of the spatiotemporal data. If the valid time is a time point, then $t_{s(f)} = t_{e(f)}$. If the valid time is a time period, then $t_{s(f)} < t_{e(f)}$. The reference time, which is a measure of the valid time at the time axis position, is denoted by k. In stRDFS model, k is set to "now" and temporal states of the model (s, < p: <t, l>, o) are determined by k. The relationship is as follows:

	Before	After	Now
$t_{s(f)} < t_{e(f)}$	$t_{e(f)} < k$	$k < t_{s(f)}$	$t_{s(f)} < k < t_{e(f)}$
$t_{s(f)} = t_{e(f)}$	$t_{e(f)} = t_{e(f)} < k$	$k < t_{s(f)} = t_{e(f)}$	$t_{s(f)} = t_{e(f)} = k$

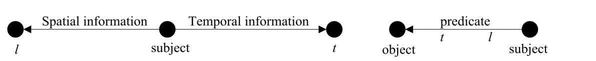
If there is no spatial data, stRDFS model becomes a tRDFS model (s, p: Ti, o) with only temporal data, where s is a resource name, p is a property name with temporal data Ti, and $o \in R \cup B \cup K \cup I$ is the value of p. In tRDFS model, there are three mappings of f_{s-o}, f_{s-t} and f_{p-t}. When only f_{s-o} exists, that is, when there is no temporal data, the formed expression is (s, p, o). When only f_{s-t} exists, it indicates that temporal data of s is Ti and the formed expression is (s, p, Ti) where the property represents "temporal information" and the attribute value is the temporal data of s. When f_{p-t} and f_{s-o} are present at the same time, the resulting tRDFS model is (s, p: Ti, o), indicating that temporal data of the model is Ti. When only f_{p-t} and f_{s-t} are formed, the expression is (s, p: Ti_1, Ti_2), indicating that temporal data of the model expression is Ti_1, and temporal data of s is Ti_2. If given a tRDFS model m (s, p: Ti, o), a tRDFS graph for m is a labeled graph M (V', E', F', λ', T') where:

- $V' = s \cup Range\ (f_{s-o} \cup f_{s-t})$ is the set of vertices.
- $E' = \{(r, r')\}$ is the set of edges where $\forall r, r' \in V'$.
- $F'\ (r, r') = \{f \mid (r, f : Ti, r') \in M\}$ is the mappings set of E' where $\forall r, r' \in V'$.
- λ' is the label given by vertices or edges.
- $T' \in Range\ (f_{s-t} \cup f_{p-t})$.

In tRDFS graph, there are two cases: the first one is that vertices contain temporal information, in this case $T' \in Range$ (f_{s-t}), as shown in Figure 3(a), the second is that tRDFS graph edges contain temporal information, in this case $T' \in Range\ (f_{p-t})$, as shown in Figure 3(b).

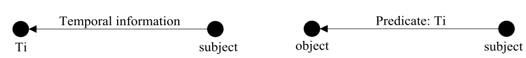
In order to illustrate the practical application of tRDFS clearly, we put forward the following examples:

Example 1: When data set A in the database is called by the program, the time slice is 1t, 4t and [8t, 13t] (current



(a) $T \in Range\ (f_{s-t})$ and $L \in Range\ (f_{s-l})$ (b) $T \in Range\ (f_{p-t})$ and $L \in Range\ (f_{p-l})$

FIGURE 2. Representation of spatiotemporal information in stRDFS.



(a) $T' \in Range\ (f_{s-t})$ (b) $T' \in Range\ (f_{p-t})$

FIGURE 3. Representation of temporal information in tRDFS graph.

time is represented by 0t), then tRDFS model is expressed as: (program, call1: (1t, 0t), A), (program, call2: (4t, 0t), A) and (program, call3: ([8t, 13t], 0t), A). Without the form of (program, call, A, [1t, 4t, [8t, 13t]]), tRDFS model links temporal data and attributes to form temporal properties. When the temporal data changes, the corresponding temporal properties change. For example, (program, call1: (1t, 0t), A) and (program, call2: (4t, 0t), A) are different model expressions and *call1*, *call2* and *call3* are different temporal properties. Next, we use the (s, p, o, t) model to represent temporal data in Figure 4 and use tRDFS model in Figure 5. By comparing these two figures, we can conclude that tRDFS model can show temporal information more clearly and accurately than the (s, p, o, t) model does.

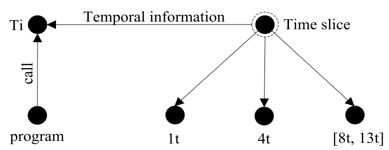


FIGURE 4. The (s, p, o, t) model graph of Example 1.

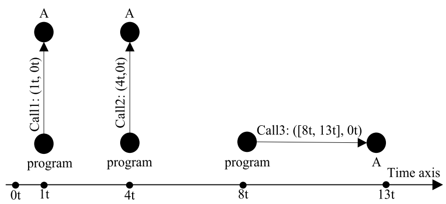


FIGURE 5. The tRDFS graph of Example 1.

Definition 7: Given an stRDFS expression g (s, p: <t, l>, o) and (v_i, f_{vi-vj}, v_j) $\in$ g, the spatial data of (v_i, f_{vi-vj}, v_j) is represented by $Si = f\ (L, D, H)$ where:

- $L \in (0°, 90°\text{N}) \cup (0°, 90°\text{S})$ which stands for latitude.
- $D \in (0°, 180°\text{E}) \cup (0°, 180°\text{W})$ which stands for longitude.
- H is the height above the sea level.

In real life, there may be situations that stRDFS model describes the spatial data of large areas. Therefore, in stRDFS model, longitude, latitude, and altitude may be intervals rather than certain values. In stRDFS model, we use $x \sim C$ to describe the range, where x represents L, D or H, $\sim$ denotes a set of $\{<, \leq, \geq, =, >, \neq\}$ and C stands for rational numbers with unit. For example, *AreaA* has a latitude range of (30°N, 40°N), a longitude range of (116°E, 118°E), and its average elevation is 50 m. In stRDFS model, *Si* is expressed as ((30°N, 40°N), (116°E, 118°E), 50m). The stRDFS description of spatial data is shown as the following:

```
ex: areaA
strdfs: hasGeometry
"(30°N < L < 40°N) and (116°E < D < 118°E) and (H = 50m)" ^^
strdfs: SemiLinearPointSet.
```

Definition 8: Given an stRDFS model g (s, p: <t, l>, o) and (x, f_{x-y}: <t, l>, y) $\in$ g, then for (x, f_{x-y}: <t, l>, y), $t = Ti\ (f_{x-y})$ and $l = Si\ (f_{x-y})$ which is represented as Definition 6 and Definition 7, respectively.

If there is no temporal data in stRDFS model, it turns into an sRDFS model (s, p: Si, o) where s is a resource name, p is a property name with spatial data Si, and $o \in R \cup B \cup K \cup S$ is the value of p. The model sRDFS, which is an stRDFS model with only spatial data, is based on RDF model and better than sRDF model (s, p, o, l). In sRDFS model, there are three types of mappings: f_{s-o}, f_{s-l} and f_{p-l}. When only f_{s-o} exists, that is, when there is no spatial data in sRDFS, the formed expression is (s, p, o). When only f_{s-l} exists, it indicates that the spatial data of s is Si. The formed expression is (s, p, Si) where the property represents "spatial information" and the property value is the spatial data of s. When f_{p-l} and f_{s-o} are present, the resulting sRDFS model is (s, p: Si, o), indicating that the spatial data of the p is Si. When only f_{p-l} and f_{s-l} are formed, the formed sRDFS is (s, p: Si_1, Si_2), indicating that the spatial data of the property is Si_1 and the spatial data of s is Si_2. If given an sRDFS model q (s, p: Si, o), an sRDFS graph for q is a labeled graph Q (V'', E'', F'', λ'', L') where:

- $V'' = s \cup Range\ (f_{s-o} \cup f_{s-l})$ is the set of vertices.
- $E'' = \{(r, r')\}$ is the set of edges where $\forall r, r' \in V''$.
- $F''(r, r') = \{f \mid (r, f : Si, r') \in Q\}$ is the mappings set of E'' where $\forall r, r' \in V''$.
- λ'' is the label given by vertices or edges.
- $L' \in Range\ (f_{s-l} \cup f_{p-l})$.

In sRDFS graphs, there are two cases: the first one is that vertices contain spatial data, $L' \in Range\ (f_{s-l})$ at this time, as shown in Figure 6(a); the second is that edges contain spatial data, as Figure 6(b) shows, $L' \in Range\ (f_{p-l})$ at this time. Then the following examples verify this model.

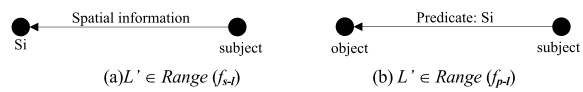


FIGURE 6. Representation of spatial information in sRDFS graph.

Example 2: Mary has taken some pictures in many areas: area1 ([23.7°N, 24.0°N], [116.5°E, 116.8°E], 50m), area2 ([42.5°N, 43°N], [51.3°E, 51.4°E], 178m), area3 ([35.3°N, 35.8°N], [169.4°E, 169.8°E], 247m) and area4 ([43.4°N, 43.5°N], [74°W, 74.4°W], 12m). Described by sRDF model (s, p, o, l), the expression is: (picture, locate in, [area1, are2, area3, area4]), the resulting sRDF graph is shown in Figure 7. Before accessing spatial data through these nodes, we should search for the "area" node at first. Based on this, it will incur unnecessary overhead of searching for spatial data. This problem can be solved with the help of sRDFS model, in which the information can be described as: (Mary, took1: area1, picture), (Mary, took2: area2, picture), (Mary, took3: area3, picture) and (Mary, took4: area4, picture). The sRDFS graph is shown in Figure 8. The sRDFS model uses both spatial data and the attribute "took" to form spatial attributes. Due to various spatial locations, the spatial attributes *took*1, *took*2, *took*3 and *took*4 are different. When querying spatial data, we can obtain the data directly by accessing the properties.

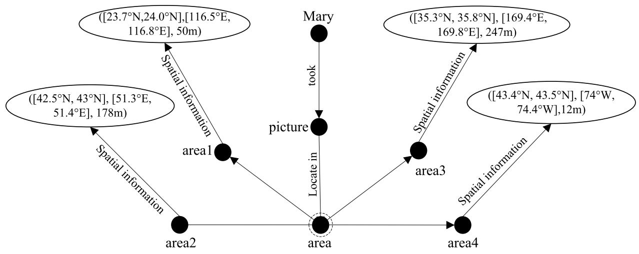


FIGURE 7. The knowledge graph by sRDF model.

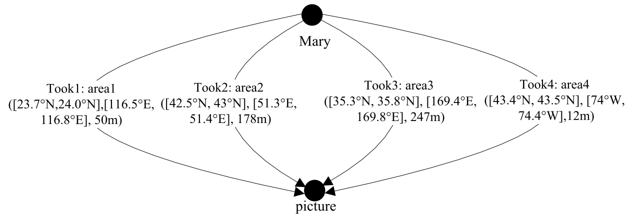


FIGURE 8. The knowledge graph by sRDFS model.

Example 3: Figure 1 shows the spatial data and temporal data of mobile receiver through stRDF model. This example uses stRDFS model to describe the mobile receiver by the form of stRDFS graph which is shown in Figure 9.

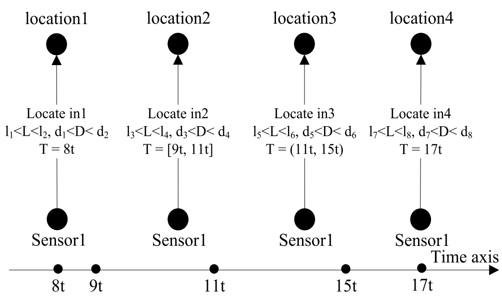


FIGURE 9. The stRDFS graph of Example 3.

B. CLASSES AND DESCRIPTIONS OF stRDFS

In order to introduce the stRDFS description more clearly, this section introduces several main classes: *strdfs: SpatialObject*, *strdfs: SpatialGeometry*, *strdfs: SpatialFeature*, *strdfs: TemporalObject*, *strdfs: TimeSlice*, *strdfs: TemporalFeature*, *strdfs: SpatiotemporalObject*, *strdfs: SpatiotemporalGeo* and *strdfs: SemiLinearPointSet.* Their relations are shown in Figure 10.

1) SPATIAL CLASSES AND DESCRIPTIONS IN SPATIOTEMPORAL DOMAIN

This section defines several main stRDFS classes that describe spatial data in spatiotemporal domain. They are: *strdfs: SpatialObject*, *strdfs: SpatialGeometry* and *strdfs: SpatialFeature*.

The class *strdfs: SpatialObject* is equivalent to the RDFS class *geo: SpatialObject*. The classes set of RDFS contains *geo: SpatialObject* and the classes set of stRDFS contains *strdfs: SpatialObject.* The class *strdfs: SpatialObject*

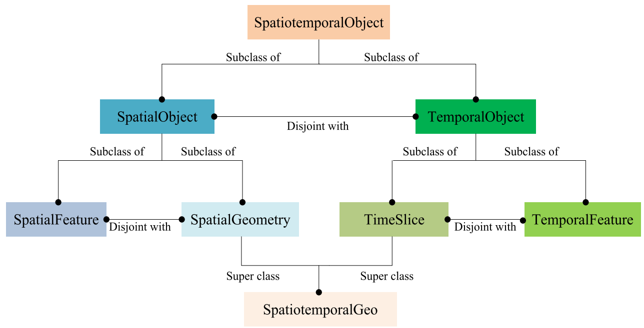


FIGURE 10. The relations of main stRDFS classes.

represents a set of all entities with only spatial information and its description is as follows:

strdfs: SpatialObject	a rdfs: Class, owl:Class;
rdfs: label	"Spatial Object" @en;
rdfs: comment	"The class SpatialObject represents everything that can have spatial information. It is super class of *SpatialFeature* and *SpatialGeometry*" @en

The class strdfs: SpatialGeometry is a subclass of strdfs: SpatialObject. It describes *Si* in stRDFS model (*s*, *p*: <*Ti*, *Si*>, o), including the data of latitude, longitude and altitude. The description of strdfs: SpatialGeometry is as follows:

strdfs: SpatialGeometry	a rdfs: Class, owl: Class;
rdfs: label	"SpatialGeometry" @ en;
rdfs:subClassOf	strdfs: SpatialObject;
owl:disjointWith	strdfs:SpatialGeometry;
rdfs:comment	"This class represents the spatial geometry characteristics of geographic locations. This class is equivalent to *geo: Geometry* for RDFS model, and it is superclass of all geometry types." @ en

The class *strdfs: SpatialFeature* is a subclass of strdfs: SpatialObject, which contains geographic information that is used for describing landform, terrain and so on. Besides, *strdfs: SpatialFeature* and *strdfs: SpatialGeometry* are exclusive mutually. The description is as follows:

strdfs: SpatialFeature	a rdfs: Class, owl: Class;
rdfs: label	"Spatial Feature" @en;
rdfs: subClassOf	strdfs: SpatialObject;
owl: disjointWith	strdfs: SpatialGeometry;
rdfs: comment	"This class represents the spatial characteristics of geographic locations except geometric information. This class is equivalent to GFI_Feature defined in ISO 19156 and geo: Feature for RDFS model, and it is superclass of all feature types." @en

Let us illustrate spatial classes more clearly by Example 4. Since stRDF model can also describe spatial data, we compare stRDFS model with stRDF model by Example 4.

Example 4: There is a mobile large sound wave receiver whose several parts are placed in different places. By describing their spatial data, we compare stRDF model with stRDFS model. The description of spatial data in stRDF model is as follows:

ex: receiver1	rdf: type	ex: SoundWaveReceiver
ex: receiver1	ssn: measures	ex: sound
ex: receiver1	ssn: hasLocation	ex: location
ex: location	strdf: hasGeometry	"(20.9°N < L < 21°N and 45.8°E < D <= 46°E) or (L = 21°N and D = 46°E) or (29°N < L < 30°N and 49°E < D <= 51°E)" ^^ strdf: SemiLinearPointSet

In the description of the above sound wave receiver, ssn is the namespace of CSIRO/SSN Ontology. The ex: location represents spatial location of sound wave receiver, and "(20.9°N < L < 21°N and 45.8°E < D <= 46°E) or (L = 21°N and D = 46°E) or (29°N < L < 30°N and 49°E < D <= 51°E)" is the attribute value of ex: location. The description of the spatial data in stRDFS model is as follows:

ex: receiver1	rdf: type	ex: SoundWaveReceiver
ex: receiver1	ssn: measures	ex: sound
ex: receiver1	ssn: hasLocation1	ex: location1
ex: receiver1	ssn: hasLocation2	ex: location2
ex: receiver1	ssn: hasLocation3	ex: location3
ssn: hasLocation1	strdfs: SpatialGeometry	"(20.9°N < L < 21°N and 45.8°E < D <= 46°E)" ^^ strdfs: SemiLinearPointSet
ssn: hasLocation2	strdfs: SpatialGeometry	"(L = 21°N and D = 46°E)" ^^ strdfs: SemiLinearPointSet
ssn: hasLocation3	strdfs: SpatialGeometry	"(29°N < L < 30°N and 49°E < D <= 51°E)" ^^ strdfs: SemiLinearPointSet

By comparing the descriptions of stRDF and stRDFS, we can draw a conclusion: *strdf: hasGeometry* describes the spatial data of a spatial object; *strdfs: SpatialGeometry* describes the spatial data of any parts of spatial object or partial spatial data of the whole spatial object. On this basis, it can be concluded that stRDFS model supports fast search for spatial data and can represent changes of spatial data either overall or partial.

2) TEMPORAL CLASSES AND DESCRIPTIONS IN SPATIOTEMPORAL DOMAIN

This section defines several main stRDFS classes that describe temporal data in spatiotemporal domain. They are: *strdfs: TemporalObject*, *strdfs: TimeSlice* and *strdfs: TemporalFeature*.

The class *strdfs: TemporalObject* is a set of all entities that contain temporal data. The description of *strdfs: TemporalObject* is as follows:

strdfs: TemporalObject	a rdfs: Class, owl: Class;
rdfs: label	"Temporal Object" @ en;
rdfs: comment	"The class TemporalObject represents everything that can have temporal information. It is superclass of *TemporalFeature* and *TimeSlice*" @en

The class *strdfs: TimeSlice* is a subclass of *strdfs: TemporalObject*. It is a set of valid time and reference time of entities, including time points and time intervals. In stRDFS model, *strdfs: TimeSlice* describes parameter Ti. The description is as follows:

strdfs: TimeSlice	a rdfs: Class, owl: Class;
rdfs: label	"TimeSlice" @ en;
rdfs: subClassOf	strdfs: TemporalObject;
owl: disjointWith	strdfs: TemporalFeature;
rdfs:comment	"This class represents the time occupied by everything with temporal information. This class describes the time point or time interval" @ en

The class *strdfs: TemporalFeature* is a subclass of *strdfs: TemporalObject* and it is mutually exclusive in *strdfs: TimeSlice*. It includes other temporal data of temporal entities, such as time zone, tense, time dimension and the existence time. The description of *strdfs: TemporalFeature* is as follows:

strdfs: TemporalFeature	a rdfs: Class, owl: Class;
rdfs: label	"Temporal Feature" @en;
rdfs: subClassOf	strdfs: TemporalObject;
owl: disjointWith	strdfs: TimeSlice;
rdfs:comment	"This class represents the temporal information of everything except the valid time and reference time." @ en

The temporal classes can be illustrated more clearly with Example 5. Since stRDF model can also describe temporal data, we compare stRDFS model with stRDF model by the description of Example 5.

Example 5: There is a Java program program1, which is called by the computer at 8t and 18t. There is a python program program2, which is called by the computer at [9t, 14t]. We set 0t as the reference time, and use this example to compare stRDF and stRDFS models. stRDF model describes the above information as follows:

ex: program1	rdf: type	ex: JavaProgram
ex: program2	rdf: type	ex: PythonProgram
ex: program1	om: procedure	ex: CountProgram
ex: program2	om: procedure	ex: OutputProgram
ex: program1	om:hasPro1Call	ex: TimeSlice1
ex: program2	om: hasPro2Call	ex: TimeSlice2
ex: TimeSlice1	strdf: TimeSlice	"(t = 8t and t = 18t) and k = 0t." ^^ strdf: SemiLinearPointSet.
ex: TimeSlice2	strdf: TimeSlice	"9t $\leq t \leq$ 14t and k = 0t." ^^ strdf: SemiLinearPointSet

In the description of the above program, om is the namespace of O&M-OWL ontology and ex represents an example ontology. The ex: TimeSlice1 represents the time slice when program1 was called and ex: TimeSlice2 represents the time slice when program2 was called. The description of stRDFS model is as follows:

ex: program1	rdf: type	ex: JavaProgram
ex: program2	rdf: type	ex: PythonProgram
ex: program1	om: procedure	ex: CountProgram
ex: program2	om: procedure	ex: OutputProgram
ex: program1	om:hasPro1Call1	ex: TimeSlice1
ex: program1	om: hasPro1Call2	ex: TimeSlice2
ex: program2	om: hasPro2Call1	ex: TimeSlice2
om:hasPro1Call1	strdfs: TimeSlice	"t = 8t and k = 0t." ^^ strdfs:SemiLinearPointSet.
om:hasPro1Call2	strdfs: TimeSlice	"t = 18t and k = 0t."^^ strdfs: SemiLinearPointSet.
om:hasPro2Call1	strdfs: TimeSlice	"9t $\leq t \leq$ 14t and k = 0t."^^ strdfs: SemiLinearPointSet.

The class strdfs: TimeSlice represents the valid time and reference time of the program. When the valid time is the time point, stRDFS expresses temporal data by equation, such as t = 8t and t = 18t. When the valid time is a time period, an inequation represents the time interval, such as 9t ≤ t ≤ 14t. Compared with stRDF model, stRDFS makes temporal data link with the attributes: om: hasPro1Call1, om: hasPro1Call2 and om: hasPro2Call1, where om: hasPro1Call1 represents the first call of program1, om: hasPro1Call2 represents the second call of program1 and om: hasPro2Call1 represents the first call of program2. Based on this feature, the stRDFS model can represent temporal data of a temporal entity at any time and record changes of temporal attributes at any time.

3) SPATIOTEMPORAL CLASSES AND DESCRIPTIONS IN SPATIOTEMPORAL DOMAIN

In this section, we define several main stRDFS classes to describe spatiotemporal data in the spatiotemporal domain which are: *strdfs: SpatiotemporalObject*, *strdfs: SpatiotemporalGeo* and *strdfs: SemiLinearPointSet*.

The class strdfs: SpatiotemporalObject is a set of all spatiotemporal entities, which is a superset of *strdfs: TemporalObject* and *strdfs: SpatialObject*. The description is as follows:

strdfs: SpatiotemporalObject	a rdfs: Class, owl: Class;
rdfs: label	"Spatiotemporal Object" @ en;
rdfs: comment	"The class: SpatiotemporalObject represents everything that can have spatiotemporal information. It is super class of SpatialObject and TemporalObject" @ en

The class *strdfs: SpatiotemporalGeo* describes geometric data of spatiotemporal entities and it is a superset of *strdfs: SpatialGeometry* and *strdfs: TimeSlice*. The description is as follows:

strdfs: SpatiotemporalGeo	a rdfs: Class, owl: Class;
rdfs: label	"Spatiotemporal Geometry" @ en;
rdfs: comment	"The class is based on *strdf: hasTrajectory*. It is super class of SpatialGeometry and TimeSlice " @ en

The class *strdfs: SemiLinearPointSet* is the set of rational numbers to represent time values, longitude values, latitude values and altitude values, etc. The description is as follows:

strdfs: SemiLinearPointSet	a rdfs: Class, owl: Class;
rdfs: label	"SemiLinearPointSet" @ en;
rdfs: comment	"This class is the set of rational numbers" @ en

We illustrate the spatiotemporal classes through Example 6, and compare the descriptions of stRDF with stRDFS.

Example 6: There is a large sonic receiver which calls corresponding programs to analyze the sound waves after receiving them. We make a comparison between stRDF model and stRDFS model in the description of spatiotemporal information. stRDF model describes the above information as follows:

ex: program1	rdf: type	ex: JavaProgram
ex: program2	rdf: type	ex: PythonProgram
ex: program1	om: procedure	ex: CountProgram
ex: program2	om: procedure	ex: OutputProgram
ex: program1	om: hasPro1Call	ex: receiver2
ex: program2	om: hasPro2Call	ex: receiver2
ex: receiver2	rdf: type	ex: SoundWaveReceiver
ex: receiver2	ssn: measures	ex: sound
ex: receiver2	ssn: hasLocation	ex: location1
ex: location1	strdf: hasTrajectory	"(t = 6t and t = 16t or 7t $\leq t \leq$ 14t) and k = 0t and ((21.8°N < L < 22.1°N and 45.9°E < D <= 46°E) or (L = 22.1°N and D = 46°E))" ^^ strdf: SemiLinearPointSet.

In stRDF model, strdf: hasTrajectory is the attribute value of strdf: hasTrajectory of location1 which describes the spatiotemporal data. It is noted that only spatiotemporal data of the whole entities can be recorded in the spatiotemporal dimension instead of the spatiotemporal data of a certain part of the object. For instance, stRDF model cannot represent the spatial data at t = 6t and the temporal data when the object is at (L = 22.1°N and D = 46°E). Even worse, if there is an attribute value changing at a certain time or in a certain spatial position, stRDF is inaccurate in recording data for its weak capability in linking spatiotemporal data. The stRDFS model solves this problem and the results are as follows:

ex: program1	rdf: type	ex: JavaProgram
ex: program2	rdf: type	ex: PythonProgram
ex: program1	om: procedure	ex: AnalysisProgram
ex: program2	om: procedure	ex: AnalysisProgram

ex: program1	om:hasPro1Call	ex: receiver2
ex: program2	om: hasPro2Call	ex: receiver2
ex: receiver2	rdf: type	ex: SoundWaveReceiver
ex: receiver2	ssn: measures	ex: sound
ex: receiver2	ssn: hasLocation1	ex: location1
ex: receiver2	ssn: hasLocation2	ex: location2
om: hasPro1Call	strdfs: SpatiotemporalGeo	“t = 6t and t = 16t and k = 0t and (21.8°N < L < 22.1°N and 45.9°E < D <= 46°E)” ^^ strdfs: SemiLinearPointSet
om: hasPro2Call	strdfs: SpatiotemporalGeo	“7t ≤ t ≤ 14t and L = 22.1°N and D = 46°E and k = 0t” ^^ strdfs: SemiLinearPointSet
ssn: hasLocation1	strdfs: SpatialGeometry	“(21.8°N < L < 22.1°N and 45.9°E < D <= 46°E)” ^^ strdfs: SemiLinearPointSet
ssn: hasLocation2	strdfs: SpatialGeometry	“(L = 22.1°N and D = 46°E)” ^^ strdfs: SemiLinearPointSet

The receiver calls program1 to analyze the data at Location1 when t = 6t and t = 16t, and calls program2 to analyze the data at Location2 when t ∈ [7t, 14t]. In stRDFS model, spatiotemporal data describes om: hasPro1Call and om: hasPro2Call. Different from stRDF, stRDFS model can represent spatiotemporal data and record changes of spatiotemporal attributes at any time or in any location because that facilitates the modification of common attribute values, such as program names.

4) A CASE OF SPATIOTEMPORAL KNOWLEDGE GRAPH MODEL

In order to verify the usability of the proposed model, it is applied in this subsection. We choose the flight of Southwest 1524, from Los Angeles to San Francisco, took off at 5:23 a.m. and landed at 6:14 a.m. on 9 August 2019. Picking the point in the flight path every two minutes and name it Pi ($i = 1, 2, \ldots$), the flight information is shown in Table 1.

Taking $P2$ as an example, as shown in Figure 11, we will apply the methodology above section proposed to aeronautics field.

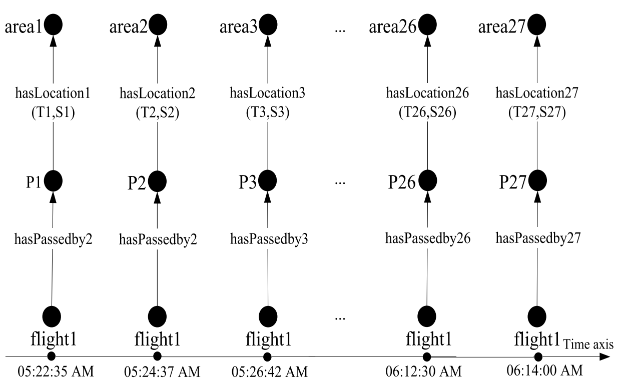


FIGURE 11. The stRDFS graph of Southwest 1524.

TABLE 1. Spatiotemporal data of Southwest 1524.

Point	Time (CST)	Latitude	Longitude	Feet/m
P1	05:22:35 AM	33.9389	-118.3900	150
P2	05:24:37 AM	33.8877	-118.4981	4,150
P3	05:26:42 AM	33.8086	-118.6459	9,550
P4	05:28:48 AM	33.7913	-118.8754	14,375
P5	05:30:45 AM	33.8788	-119.0721	19,925
P6	05:32:35 AM	34.0390	-119.2378	24,675
P7	05:34:33 AM	34.2246	-119.4261	28,800
P8	05:36:33 AM	34.4233	-119.6296	32,050
P9	05:38:46 AM	34.6417	-119.8546	36,000
P10	05:40:24 AM	34.8029	-120.0215	36,000
P11	05:42:37 AM	35.0218	-120.2497	36,000
P12	05:44:37 AM	35.2173	-120.4552	36,000
P13	05:46:49 AM	35.4360	-120.6779	36,000
P14	05:48:49 AM	35.6380	-120.8695	36,000
P15	05:50:40 AM	35.8254	-121.0478	36,000
P16	05:52:32 AM	36.0149	-121.2312	34,775
P17	05:54:29 AM	36.2047	-121.4151	29,650
P18	05:56:31 AM	36.3925	-121.5987	25,125
P19	05:58:37 AM	36.5898	-121.7745	20,625
P20	06:00:26 AM	36.7608	-121.8987	16,700
P21	06:02:26 AM	36.9641	-121.9578	12,850
P22	06:04:26 AM	37.1471	-122.0294	10,000
P23	06:06:30 AM	37.3027	-122.0903	7,450
P24	06:08:39 AM	37.4542	-122.1459	4,950
P25	06:10:31 AM	37.5614	-122.1943	2,750
P26	06:12:30 AM	37.5886	-122.2975	1,075
P27	06:14:00 AM	37.6109	-122.3506	46

According to $Ti = f\ (N, k\ (i = 1, 23\ldots)$ in Definition 6, and “N = 05:24:37 AM, k = 0t” in Table 1, whose temporal data can be represented by $T2$ = (05:24:37 AM, 0t).

According to $Si = f\ (L, D, H)\ (i = 1, 23\ldots)$ in Definition 7, and “L = 33.8877, D = −118.4981, H = 4150m”, whose spatial data can be represented by $S2$ = (33.8877, −118.4981, 4150m).

According to stRDFS expression $g\ (s, p: <t, l>, o)$ in Definition 8, whose stRDFS model can be represented by $(P2, Locate\ in2: <T2, S2>, area2)$.

According to Section 3.2, the description of Figure 11 is as follows:

ex: flight1	rdf: name	ex: Southwest 1524
ex: flight1	rdf: type	ex: B737
ex: flight1	om: hasPassedby1	ex: P1
ex: flight1	om: hasPassedby2	ex: P2
ex: flight1	om: hasPassedby3	ex: P3
…	…	…
ex: flight1	om: hasPassedby26	ex: P26
ex: flight1	om: hasPassedby27	ex: P27
ex: P1	ssn: hasLocation1	ex: area1
ex: P2	ssn: hasLocation2	ex: area2
ex: P3	ssn: hasLocation3	ex: area3
…	…	…
ex: P26	ssn: hasLocation26	ex: area26
ex: P27	ssn: hasLocation27	ex: area27
om: hasPassedby1	strdfs: SpatiotemporalGeo	“t=05:24:37 AM t and k=0t and (L = 33.9389, D=−118.3900 and H=150m)” ^^ strdfs: SemiLinearPointSet
om: hasPassedby2	strdfs: SpatiotemporalGeo	“t=05:24:37 AM and k=0t and (L=33.8877, D=−118.4981 and H=4150m)” ^^ strdfs: SemiLinearPointSet

om: hasPassedby3	strdfs: SpatiotemporalGeo	"t=05:26:42 AM and k=0t and (L=33.8086, D=−118.6459 and H=9550m)" ^^ strdfs: SemiLinearPointSet
…	…	…
om: hasPassedby26	strdfs: SpatiotemporalGeo	"t=06:12:30 AM and k=0t and (L=37.5886, D=−122.2975 and H=1,075m)" ^^ strdfs: SemiLinearPointSet
om: hasPassedby27	strdfs: SpatiotemporalGeo	"t=06:14:00 AM and k=0t and (L=37.6109, D=−122.3506 and H=46m)" ^^ strdfs: SemiLinearPointSet
ssn: hasLocation1	strdfs: SpatialGeometry	"(L=33.9389, D=−118.3900 and H=150m)" ^^ strdfs: SemiLinearPointSet
ssn: hasLocation2	strdfs: SpatialGeometry	"(L=33.8877, D=−118.4981 and H=4150m)" ^^ strdfs: SemiLinearPointSet
ssn: hasLocation3	strdfs: SpatialGeometry	"(L=33.8086, D=−118.6459 and H=9550m)" ^^ strdfs: SemiLinearPointSet
…	…	…
ssn: hasLocation26	strdfs: SpatialGeometry	"(L=37.5886, D=−122.2975 and H=1,075m)" ^^ strdfs: SemiLinearPointSet
ssn: hasLocation27	strdfs: SpatialGeometry	"(L=37.6109, D=−122.3506 and H=46m)" ^^ strdfs: SemiLinearPointSet

IV. TOPOLOGICAL RELATIONS OF ENTITIES

In order to describe the relations among spatiotemporal entities, this section defines eleven kinds of topological relations, which are: *Equal*, *Disjoint*, *Meet*, *Overlap*, *Cover*, *CoveredBy*, *Inside*, *Contain*, *Before*, *Now* and *After*. The spatial topological relations are *Equal*, *Disjoint*, *Meet*, *Overlap*, *Cover*, *CoveredBy*, *Inside* and *Contain*, so their domains are *strdfs: SpatialObject*. The temporal topological relations are *Before, Now and After*, so their domain is *strdfs: TemporalObject*. The relations among relation names, relation URI and domain are shown in Table 2.

In the following, we define the following determination methods of topological relations among different spatiotemporal entities based on the stRDFS model:

Definition 9: Given two spatiotemporal entities A (s_A, p_A: <Ti_A, Si_A>, o_A) and B (s_B, p_B: <Ti_B, Si_B>, o_B), Equal (A, B) $\Leftrightarrow s_A = s_B \wedge o_A = o_B \wedge p_A = p_B \wedge Si_A = Si_B$. ($Ti_A = Ti_B$?)

Definition 10: Given two spatiotemporal entities A (s_A, p_A: <Ti_A, Si_A>, o_A) and B (s_B, p_B: <Ti_B, Si_B>, o_B), A is disjoint with B where:

- If o denotes spatial data, Disjoint $(A, B) \Leftrightarrow Range\ (f_{sA-SisA} \cup f_{pA-SiA}) \cap Range(f_{sB-SisB} \cup f_{pB-SiB}) = \emptyset$.
- If o doesn't denote spatial data, Disjoint $(A, B) \Leftrightarrow Range\ (f_{pA-SiA}) \cap Range\ (f_{pB-SiB}) = \emptyset$.

By Definition 10, if o represents spatial data, stRDF model can be expressed as (s_A, p_A: <Ti_A, Si_A>, Si_{SA}). As shown in Definition 3, Si_{SA} represents spatial data of s_A and Si_A represents spatial data of attributes p_A. The same is B. Therefore, both SiSA and SiA need to be considered, that is, Range $(f_{sA-SisA} \cup f_{pA-SiA}) \cap$ Range $(f_{sB-SisB} \cup f_{pB-SiB}) = \emptyset$. If o doesn't represent spatial data, stRDF model can be expressed as (s_A, p_A: <Ti_A, Si_A>, o_A) where oA is the object without spatial data. Therefore, only Si_A need to be considered.

TABLE 2. Topological relations of spatiotemporal classes.

Relation Name	Relation URI	Domain/Range
Equals	strdfs: geoEquals	strdfs: SpatialObject
Disjoint	strdfs: geoDisjoint	strdfs: SpatialObject
Meet	strdfs: geoMeet	strdfs: SpatialObject
Overlap	strdfs: geoOverlap	strdfs: SpatialObject
Covers	strdfs: geoCovers	strdfs: SpatialObject
CoveredBy	strdfs:geoCoveredBy	strdfs: SpatialObject
Inside	strdfs: geoInside	strdfs: SpatialObject
Contains	strdfs: geoContains	strdfs: SpatialObject
Before	strdfs: timBefore	strdfs:TemporalObject
Now	strdfs: timNow	strdfs:TemporalObject
After	strdfs: timAfter	strdfs:TemporalObject

Definition 12: Given two spatiotemporal entities A (s_A, p_A: <Ti_A, Si_A>, o_A) and B (s_B, p_B: <Ti_B, Si_B>, o_B), A overlaps B where:

- If o depicts spatial data, Overlap $(A, B) \Leftrightarrow Range^\circ\ (f_{sA-SisA} \cup f_{pA-SiA}) \cap Range^\circ\ (f_{sB-SisB} \cup f_{pB-SiB}) \neq \emptyset \wedge Range\ (f_{sA-SisA} \cup f_{pA-SiA}) \cap Range\ (f_{sB-SisB} \cup f_{pB-SiB}) \neq Range\ (f_{sA-SisA} \cup f_{pA-SiA}) \wedge Range\ (f_{sA-SisA} \cup f_{pA-SiA}) \cap Range\ (f_{sB-SisB} \cup f_{pB-SiB}) \neq Range(f_{sB-SisB} \cup f_{pB-SiB})$.
- If o doesn't depict spatial data, Overlap $(A, B) \Leftrightarrow Range^\circ\ (f_{pA-SiA}) \cap Range^\circ\ (f_{pB-SiB}) \neq \emptyset \wedge Range\ (f_{pA-SiA}) \cap Range\ (f_{pB-SiB}) \neq Range\ (f_{pA-SiA}) \wedge Range\ (f_{pA-SiA}) \cap Range\ (f_{pB-SiB}) \neq Range\ (f_{pB-SiB})$.

Definition 13: Given two spatiotemporal entities A (s_A, p_A: <Ti_A, Si_A>, o_A) and (s_B, p_B : <Ti_B, Si_B>, o_B), A covers B where:

- If o expresses spatial data, Cover $(A, B) \Leftrightarrow Range^\circ\ (f_{sA-SisA} \cup f_{pA-SiA}) \cap Range\ (f_{sB-SisB} \cup f_{pB-SiB}) = Range\ (f_{sB-SisB} \cup f_{pB-SiB})$
- If o doesn't express spatial data, Cover $(A, B) \Leftrightarrow Range^\circ\ (f_{pA-SiA}) \cap Range\ (f_{pB-SiB}) = Range\ (f_{pB-SiB})$

Definition 14: Given two spatiotemporal entities A (s_A, p_A: <Ti_A, Si_A>, o_A) and B (s_B, p_B: <Ti_B, Si_B>, o_B), A is covered by B where:

- If o represents spatial data, CoveredBy $(A, B) \Leftrightarrow Range\ (f_{sA-SisA} \cup f_{pA-SiA}) \cap Range^\circ\ (f_{sB-SisB} \cup f_{pB-SiB}) = Range\ (f_{sA-SisA} \cup f_{pA-SiA})$.
- If o doesn't represent spatial data, CoveredBy $(A, B) \Leftrightarrow Range\ (f_{pA-SiA}) \cap Range^\circ\ (f_{pB-SiB}) = Range\ (f_{pA-SiA})$

Definition 15: Given two spatiotemporal entities A (s_A, p_A: <Ti_A, Si_A>, o_A) and B (s_B, p_B: <Ti_B, Si_B>, o_B), A is inside B where:

- If o denotes spatial data, Inside $(A, B) \Leftrightarrow Range\ (f_{sA-SisA} \cup f_{pA-SiB}) \cap Range\ (f_{sB-SisB} \cup f_{pB-SiB}) = Range\ (f_{sA-SisA} \cup f_{pA-SiB}) \wedge Range^\circ\ (f_{sA-SisA} \cup f_{pA-SiB}) \cap Range^\circ\ (f_{sB-SisB} \cup f_{pB-SiB}) \neq \emptyset$.

- If o doesn't denote spatial data, Inside $(A, B) \Leftrightarrow Range$ $(f_{pA-SiA}) \cap Range$ $(f_{pB-SiB}) = Range$ $(f_{pA-SiA}) \wedge Range^{\circ}$ $(f_{pA-SiA}) \cap Range^{\circ}$ $(f_{pB-SiB}) \neq \emptyset$.

Definition 16: Given two spatiotemporal entities A (s_A, p_A: $<Ti_A$, $Si_A>$, o_A) and B (s_B, p_B: $<Ti_B$, $Si_B>$, o_B), A contains B where:

- If o represents spatial data, Contain $(A, B) \Leftrightarrow Range$ $(f_{sA-SisA} \cup f_{pA-SiB}) \cap Range$ $(f_{sB-SisB} \cup f_{pB-SiB}) = Range$ $(f_{sB-SisB} \cup f_{pB-SisB}) \wedge Range^{\circ}$ $(f_{sA-SisA} \cup f_{pA-SiB}) \cap Range^{\circ}$ $(f_{sB-SisB} \cup f_{pB-SiB}) \neq \emptyset$.
- If o doesn't represent spatial data, Contain $(A, B) \Leftrightarrow$ $Range$ $(f_{pA-SiA}) \cap Range$ $(f_{pB-SiB}) = Range$ $(f_{pB-SiB}) \wedge$ $Range^{\circ}$ $(f_{pA-SiA}) \cap Range^{\circ}$ $(f_{pB-SiB}) \neq \emptyset$.

Definition 17: Given two spatiotemporal entities A (s_A, p_A: $<Ti_A$, $Si_A>$, o_A) and B (s_B, p_B: $<Ti_B$, $Si_B>$, o_B), A is earlier than B where:

- If o depicts temporal data, Before $(A, B) \Leftrightarrow \Pi_N$ $Range$ $(f_{sA-TisA}) < \Pi_N$ $Range$ $(f_{sB-TisB}) \wedge \Pi_N$ $Range$ $(f_{pA-TiA}) < \Pi_N$ $Range$ (f_{pB-TiB}).
- If o doesn't depict temporal data, Before $(A, B) \Leftrightarrow \Pi_N$ $Range$ $(f_{pA-TiA}) < \Pi_N$ $Range$ (f_{pB-TiB}).

In Definition 17, Π_N (x) represents the projection of x on N. For example, Π_N $Range$ $(f_{sA-TisA})$ represents parameter N of the temporal data Ti_{sA}. If o represents temporal data, stRDF model can be expressed as (s_A, p_A: $<Ti_A$, $Si_A>$, Ti_{SA}). Ti_{SA} represents temporal data of s_A and Ti_A represents temporal data of attributes p_A. B is the same. If o doesn't represent temporal data, stRDF model can be expressed as (s_A, p_A: $<Ti_A$, $Si_A>$, o_A) where o_A is the object without temporal data. Therefore, only Ti_A needs to be considered.

Definition 18: Given two spatiotemporal entities A (s_A, p_A: $<Ti_A$, $Si_A>$, o_A) and B (s_B, p_B: $<Ti_B$, $Si_B>$, o_B), A is later than B where:

- If o is temporal data, After $(A, B) \Leftrightarrow \Pi_N$ $Range$ $(f_{sA-TisA}) > \Pi_N$ $Range$ $(f_{sB-TisB})$ and Π_N $Range$ $(f_{pA-TiA}) > \Pi_N$ $Range$ (f_{pB-TiB}).
- If o isn't temporal data, After $(A, B) \Leftrightarrow \Pi_N$ $Range$ $(f_{pA-TiA}) > \Pi_N$ $Range$ (f_{pB-TiB}).

Definition 19: Given two spatiotemporal entities A (s_A, p_A: $<Ti_A$, $Si_A>$, o_A) and B (s_B, p_B: $<Ti_B$, $Si_B>$, o_B), A and B are at the same time where:

- If o represents temporal data, Now $(A, B) \Leftrightarrow \Pi_N$ $Range$ $(f_{sA-TisA}) = \Pi_N$ $Range$ $(f_{sB-TisB})$ and Π_N $Range$ $(f_{pA-TiA}) = \Pi_N$ $Range$ (f_{pB-TiB}).
- If o doesn't represent temporal data, Now $(A, B) \Leftrightarrow \Pi_N$ $Range$ $(f_{pA-TiA}) = \Pi_N$ $Range$ (f_{pB-TiB}).

In the following, we will describe the topological relations of spatiotemporal entities through Example 7.

Example 7: Given the six spatiotemporal entities in Example 4: A, B, C, D, E and F, their topological relations are shown in the Table 3.

Based on stRDFS model, topological relations are transformed into an stRDFS graph, as shown in Figure 12. The dotted elliptical portion indicates the spatial position range. In Figure 12, the position of A is equal to B, so subjects of A and B are expressed as an ellipse A/B, properties are P_A/P_B and objects are O_A/O_B. The position of A/B is disjoint with C, D and E, so the dotted ellipse of A/B is disjoint with the dotted ellipses of C, D and E. The dotted ellipse of A/B is contained in the dotted ellipse of F because that the position of A/B is inside of F. The temporal relation is $T_A = T_B = T_F < T_E < T_C = T_D$, and. The position of A/B and F changes into the position of E at Ti_E and D at Ti_D, respectively.

TABLE 3. Topological relations in Example 7.

Relation Name	Relation URI	Relation Name	Domain/Range
A	strdfs:geoEquals	*B*	strdfs:SpatiotemporalObject
A, B	strdfs:geoDisjoint	*C, D, E*	strdfs:SpatiotemporalObject
C	strdfs: geoMeet	*D*	strdfs:SpatiotemporalObject
C	strdfs:geoOverlap	*E*	strdfs:SpatiotemporalObject
F	strdfs:geoCovers	*D*	strdfs:SpatiotemporalObject
E	strdfs:geoCoveredBy	*A, B*	strdfs:SpatiotemporalObject
A, B	strdfs: geoInside	*F*	strdfs:SpatiotemporalObject
F	strdfs:geoContains	*A, B*	strdfs:SpatiotemporalObject
A, B, F	strdfs: timBefore	*C, D, E*	strdfs:SpatiotemporalObject
C	strdfs: timNow	*D*	strdfs:SpatiotemporalObject
C, D, E	strdfs: timAfter	*A, B, F*	strdfs:SpatiotemporalObject

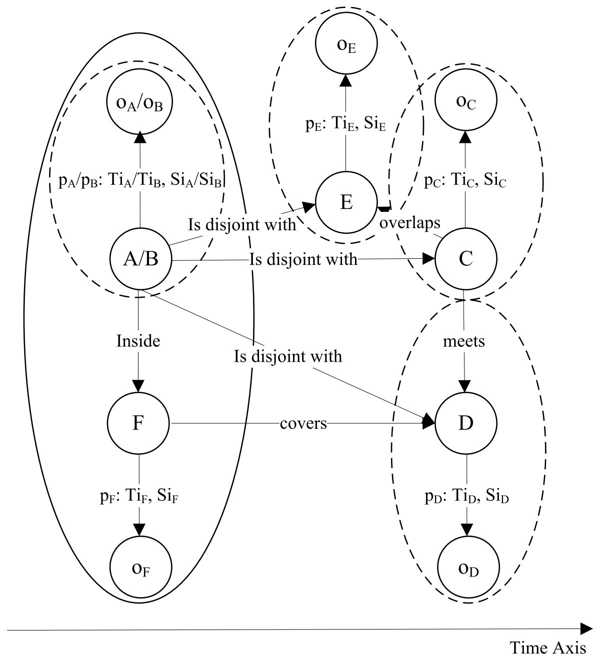
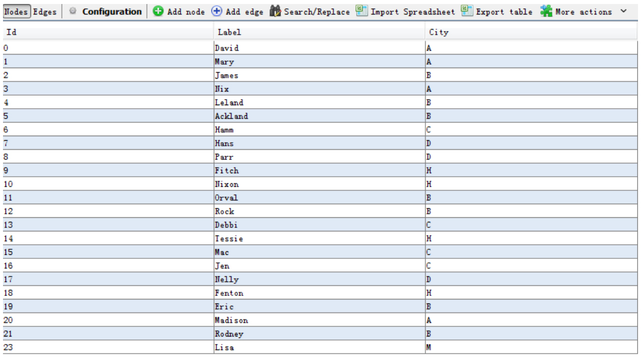


FIGURE 12. The stRDFS graph of Example 7.

This paper selects a part of David's circle of friends as experimental data. By recording their names, relationships, and locations, Gephi (a cross-platform complex network analysis software based on JVM and can display knowledge graphs with spatiotemporal information) can show the relationship between each person's spatial locations. The name of a person is the value of S, the relationship is the value of P, and the city is the value of O. Since the rest of the parameters have little effect on the spatial relationships determination of subjects, it is not necessary to introduce them into the data table. Figure 13 is a spatiotemporal data set that contains ids of people, names of people and cities in which people are



Id	Label	City
0	David	A
1	Mary	A
2	James	B
3	Nix	A
4	Leland	B
5	Ackland	B
6	Hamm	C
7	Hans	D
8	Parr	D
9	Fitch	H
10	Nixon	H
11	Orval	B
12	Rock	B
13	Debbi	C
14	Tessie	H
15	Mac	C
16	Jen	C
17	Nelly	D
18	Fenton	H
19	Eric	B
20	Madison	A
21	Rodney	B
23	Lisa	M

FIGURE 13. Data tables for some names and names of cities in which they are located.

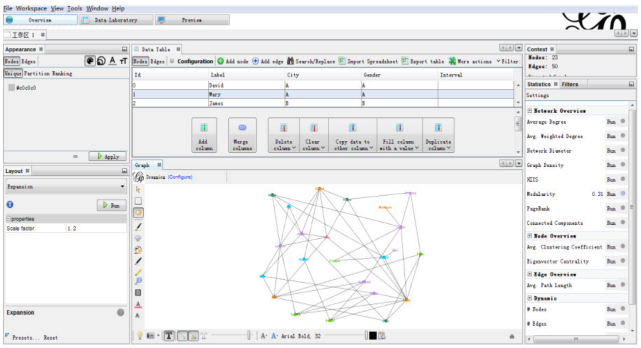


FIGURE 14. The screen snapshot of running result in Gephi.

TABLE 4. Comparative study on spatiotemporal data modeling.

	Spatial feature	Temporal feature	Spatial (S) or Temporal (T) Expansion	Spatial (S) or Temporal (T) labeling of properties
Temporal RDF[22][42]		√		T
Reifying RDF[23]		√	T	
Named graph[14]		√		
tRDF[38]		√		T
GEO[32]	√			
GeoRDF[13][8][9]	√			
YAGO2[25]	√	√	S and T	
g^{st}-Store[50]	√	√	S and T	
stRDF[28]	√	√	T	
Our work	√	√		S and T

located. The data in Figure 13 are input in Gephi, and the stRDFS model of these data is established and Figure 14 is the running result of the spatial position relationships among subjects. As shown in Figure 14, if person *A* meets person *B* (two persons are adjacent), they will be represented by the same color, and if they are disjoint, they will be represented by different colors.

V. COMPARISONS AND DISSCUSSION

In order to illustrate the novelity of stRDFS proposed in this paper clearly, we set a comparative study on modeling spatiotemporal data based on RDF.

As shown in Table 4, Temporal RDF [22], [42], Named graph [14] and tRDF [38] are focus on temporal data modeling; GEO [32] and GeoRDF [8], [9], [13] are the representation of spatial data; stRDF [28], YAGO2 [25], g^{st}-Store [50] and stRDFS devote to modeling spatiotemporal data. Among the spatiotemporal models, stRDF expands RDF triples into quad, and YAGO2 and g^{st}-Store expand RDF triples into quintuple. The expansion of triples could easily solve the express of spatiotemporal features. However, the extra labels often cause data redundancy and lead to additional overhead for the system. stRDFS is an extending RDF by labeling properties with spatiotemporal features without expanding RDF triples, which meets the needs of the purpose in this paper.

VI. CONCLUSION AND FUTURE WORK

Incorporation of spatiotemporal information in data model has been an important topic of database community because such information extensively exists in real-world applications, in which spatiotemporal data plays an important role in nature. Both classical RDF model and previous studies on RDF extension cannot satisfy the need for modeling and processing spatiotemporal data. Therefore, we explore the method to model spatiotemporal knowledge graph based on RDF. In this paper, we establish the stRDFS data model and introduce the classes and descriptions in spatiotemporal domain. This model can correlate temporal data with spatial data through properties and capture changes of spatiotemporal attribute value in time. Besides, in order to depict relations among different spatiotemporal entities, we define eleven kinds of topological relations and the determination methods, and then we describe them with stRDFS model.

Future work mainly concentrates on the following aspects: (*i*) We will try to reduce parameters and improve efficiency without affecting accuracy. (*ii*) Exploring semantic implication, graph algebra and query language. (*iii*) Our model will be extended in order to deal with uncertain spatiotemporal data. (*iv*) Creating an stRDFS dataset and platform.

REFERENCES

[1] F. Ademaj, S. Schwarz, T. Berisha, and M. Rupp, "A spatial consistency model for geometry-based stochastic channels," *IEEE Access*, vol. 7, pp. 183414–183427, Dec. 2019.

[2] J. Amjadi and M. Soroudi, "Twin signed total roman domination numbers in digraphs," *Asian-European J. Math.*, vol. 11, no. 03, Jun. 2018, Art. no. 1850034.

[3] H. Arenas, N. Aussenac-Gilles, C. Comparot, and C. Trojahn, "Ontologie pour l'intégration de données d'observation de la Terre et contextuelles basée sur les relations topologiques," in *Proc. IC*, Nancy, France, 2018, pp. 5–20.

[4] L. Bai, J. Liu, and C. Yu, "A spatiotemporal data model and marking dictionaries for sea surface meteorological data in XML," in *Proc. 12th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Aug. 2016, pp. 2184–2188.

[5] S. Batsakis and E. G. M. Petrakis, "Representing temporal knowledge in the semantic Web: The extended 4d fluents approach," in *Combinations of Intelligent Methods and Applications*, vol. 8, 4th ed. Berlin, Germany: Springer, 2011, pp. 55–69.

[6] S. Batsakis and E. G. M. Petrakis, "SOWL: Spatio-temporal representation, reasoning and querying over the semantic web," in *Proc. I-SEMANTICS*, Graz, Austria, 2010, pp. 1–9.

[7] S. Batsakis and E. G. M. Petrakis, "SOWL: A framework for handling spatio-temporal information in OWL 2.0," in *Proc. Int. Workshop Rules Rule Markup Lang. Semantic Web* vol. 6826, 6th ed. Berlin, Germany: Springer, 2011, pp. 242–249.

[8] R. Battle and D. Kolas, "Enabling the geospatial semantic Web with parliament and GeoSPARQL," *Semantic Web*, vol. 3, no. 4, pp. 355–370, 2012.

[9] A. Brodt, D. Nicklas, and B. Mitschang, "Deep integration of spatial query processing into native RDF triple stores," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. (GIS)*, San Jose, CA, USA, 2010, p. 33.

[10] V. A. and A. A., "Semantic Web mining using RDF data," *Int. J. Comput. Appl.*, vol. 133, no. 10, pp. 14–19, Jan. 2016.

[11] D. C. Chang, S. Dokos, and N. H. Lovell, "Temporo-spatial model construction using the MML and software framework," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 12, pp. 3528–3531, Dec. 2011.

[12] H. Cheng and Z. Ma, "f-*ALC* (D)-LTL: A fuzzy spatio-temporal description logic," in *Proc. KSEM*, vol. 10412. Melbourne, VIC, Australia, vol. 2017, pp. 93–105.

[13] S. Consoli, M. Mongiovic, A. G. Nuzzolese, S. Peroni, V. Presutti, D. Reforgiato Recupero, and D. Spampinato, "A smart city data model based on semantics best practice and principles," in *Proc. 24th Int. Conf. World Wide Web (WWW Companion)*, Florence, Italy, 2015, pp. 1395–1400.

[14] R. Crouch and A. L. Kalouli, "Named graphs for semantic representation," in *Proc. SEM@NAACL-HLT*, New Orleans, LA, USA, vol. 2018, pp. 113–118.

[15] Q. Cui, N. Wang, and M. Haenggi, "Vehicle distributions in large and small cities: Spatial models and applications," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10176–10189, Nov. 2018.

[16] R. Dautov and S. Distefano, "Semantic Web languages for policy enforcement in the Internet of Things," in *Proc. HCC*, Mérida, Mexico, 2018, pp. 183–188.

[17] C. Debruyne and D. O'sullivan, "Client-side processing of GeoSPARQL functions with triple Pa ern fragments," presented at 26th Workshop Linked Data Web, Perth, NSW, Australia, Apr. 2017.

[18] X. Di, J. Wang, S. Cheng, and L. Bai, "Pattern match query for spatiotemporal RDF graph," in *Proc. Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery*. Berlin, Germany: Springer, 2019, pp. 532–539.

[19] N. M. Elzein, M. A. Majid, M. Fakherldin, and I. A. T. Hashem, "Distributed join query processing for big RDF data," *Adv. Sci. Lett.*, vol. 24, no. 10, pp. 7758–7761, Oct. 2018.

[20] T. M. de Farias, A. Roxin, and C. Nicolle, "SWRL rule-selection methodology for ontology interoperability," *Data Knowl. Eng.*, vol. 105, pp. 53–72, Sep. 2016.

[21] M. Fossati, E. Dorigatti, and C. Giuliano, "N-ary relation extraction for simultaneous T-Box and A-Box knowledge base augmentation," *Semantic Web*, vol. 9, no. 4, pp. 413–439, Jun. 2018.

[22] C. Gutierrez, C. Hurtado, and A. Vaisman, "Temporal RDF," in *Proc. ESWC*, Heraklion, Greece, 2005, pp. 93–107.

[23] D. Hernández, A. Hogan, and M. Krötzsch, "Reifying RDF: What works well with Wikidata?" in *Proc. ISWC*, Bethlehem, PA, USA, 2015, pp. 32–47.

[24] L. T. T. Ho, S. Arch-int, and N. Arch-int, "Introducing fuzzy temporal description logic," in *Proc. 3rd Int. Conf. Ind. Bus. Eng. (ICIBE)*, 2017, pp. 77–80.

[25] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from wikipedia," *Artif. Intell.*, vol. 194, pp. 28–61, Jan. 2013.

[26] Z. Jiang, "Spatial structured prediction models: Applications, challenges, and techniques," *IEEE Access*, vol. 8, pp. 38714–38727, Feb. 2020.

[27] D. H. Kim, K. H. Ryu, and H. S. Kim, "A spatiotemporal database model and query language," *J. Syst. Softw.*, vol. 55, no. 2, pp. 129–149, Dec. 2000.

[28] M. Koubarakis and K. Kyzirakos, "Modeling and querying metadata in the semantic sensor Web: The model stRDF and the query language stSPARQL," in *Proc. ESWC*, Heraklion, Greece, 2010, pp. 425–439.

[29] O. Lassila and R. Swick, *Resource Description Framework (RDF) Model and Syntax Specification*, document REC-rdf-syntax-19990222, Mar. 1999.

[30] K. Y. Lee, Y. H. Lee, J. J. Kim, G. S. Choi, and K. S. Jang, "Spatio-temporal ontology for the semantic Web," *Int. Inf. Inst. (Tokyo) Inf.*, vol. 18, no. 1, pp. 329–334, Jan. 2015.

[31] C. Li, J. Huang, B. Wang, Y. Zhou, Y. Bai, and Y. Chen, "Spatial-temporal correlation prediction modeling of origin-destination passenger flow under urban rail transit emergency conditions," *IEEE Access*, vol. 7, pp. 162353–162365, 2019.

[32] J. Lieberman, R. Singh, and C. Goad, "W3C geospatial vocabulary," Incubator Group Rep., W3C, Washington, DC, USA, 2007.

[33] X. Lu, T. Hu, and F. Yin, "A novel spatiotemporal fuzzy method for modeling of complex distributed parameter processes," *IEEE Trans. Ind. Electron.*, vol. 66, no. 10, pp. 7882–7892, Oct. 2019.

[34] C. Lutz, F. Wolter, and M. Zakharyaschev, "Temporal description logics: A survey," in *Proc. 15th Int. Symp. Temporal Represent. Reasoning*, Montreal, QC, Canada, Jun. 2008, pp. 3–14.

[35] A. Mandya, D. Bollegala, F. Coenen, and K. Atkinson, "Combining long short term memory and convolutional neural network for cross-sentence n-ary relation extraction," Dept. Comput. Sci., Univ. Liverpool, Liverpool, U.K., Tech. Rep. L69 3BX, Nov. 2018.

[36] P. Nikitopoulos, A. Vlachou, C. Doulkeridis, and G. A. Vouros, "DiStRDF: Distributed Spatio-temporal RDF queries on spark," in *Proc. EDBT/ICDT*, vol. 2083. Vienna, Austria, 2018, pp. 125–132.

[37] N. Peng, H. Poon, C. Quirk, K. Toutanova, and W.-T. Yih, "Cross-sentence N-ary relation extraction with graph LSTMs," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 101–115, Dec. 2017.

[38] A. Pugliese, O. Udrea, and V. S. Subrahmanian, "Scaling RDF with time," in *Proc. 17th Int. Conf. World Wide Web (WWW)*, Beijing, China, 2008, pp. 605–614.

[39] J. Qian, X.-Y. Li, C. Zhang, L. Chen, T. Jung, and J. Han, "Social network de-anonymization and privacy inference with knowledge graph model," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 4, pp. 679–692, Jul. 2019.

[40] Q. Song, Y. Wu, P. Lin, L. X. Dong, and H. Sun, "Mining summaries for knowledge graph search," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 10, pp. 1887–1900, Oct. 2018.

[41] Y. Takama and S. Hattori, "Mining association rules for adaptive search engine based on RDF technology," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 790–796, Apr. 2007.

[42] J. Tappolet and A. Bernstein, "Applied temporal RDF: Efficient temporal querying of RDF data with SPARQL," in *Proc. ESWC*, Heraklion, Greece, 2009, pp. 308–322.

[43] Q. Tong, "Mapping object-oriented database models into RDF (S)," *IEEE Access*, vol. 6, pp. 47125–47130, 2018.

[44] B. H. Tran, C. Plumejeaud-Perreau, and A. Bouju, "A Web interface for exploiting spatio-temporal heterogeneous data," in *Web and Wireless Geographical Information Systems*. Cham, Switzerland: Springer, Apr. 2018, pp. 118–129.

[45] J. Treur, "Network reification as a unified approach to represent network adaptation principles within a network," in *Proc. Int. Conf. Theory Pract. Natural Comput.*, vol. 11324, Nov. 2018, pp. 344–358.

[46] A. Vaisman and K. Chentout, "Mapping spatiotemporal data to RDF: A SPARQL endpoint for brussels," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 8, p. 353, Aug. 2019.

[47] V. Velayutham, S. Chandrasekaran, and S. Mohan, "Web user interface based on OGC standards for sensor cloud using big data," *Int. J. Commun. Netw. Distrib. Syst.*, vol. 20, no. 4, p. 389, 2018.

[48] A. Vlachou, C. Doulkeridis, A. Glenis, G. M. Santipantakis, and G. A. Vouros, "Efficient spatio-temporal RDF query processing in large dynamic knowledge bases," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Limassol, Cyprus, Apr. 2019, pp. 439–447.

[49] J. Wang, X. Di, J. Liu, and L. Bai, "A constraint framework for uncertain spatiotemporal data in RDF graphs," in *Proc. Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery*. Berlin, Germany: Springer, 2019, pp. 727–735.

[50] D. Wang, L. Zou, and D. Zhao, "gst-Store: An engine for large RDF graph integrating spatiotemporal information," presented at the 17th EDBT, Athens, Greece, Mar. 2014.

[51] J. Wangni, "Stochastic orthant-wise limited-memory quasi-newton method," Apr. 2017, *arXiv:1704.07987v1*. [Online]. Available: https://arxiv.org/pdf/1704.07987v1

[52] E. R. Watkins and D. A. Nicole, "Named graphs as a mechanism for reasoning about provenance," in *Proc. APWeb*, Harbin, China, 2006, pp. 943–948.

[53] T. W. Wlodarczyk, C. Rong, M. O'Connor, and M. Musen, "SWRL-F: A fuzzy logic extension of the semantic Web rule language," in *Proc. WIMS*, Sogndal, Norway, 2011, p. 39.

[54] L. Xu, A. Wong, and D. A. Clausi, "A novel Bayesian spatial–temporal random field model applied to cloud detection from remotely sensed imagery," *Proc. IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 4913–4924, Sep. 2017.

[55] F. Zhang, K. Wang, Z. Li, and J. Cheng, "Temporal data representation and querying based on RDF," *IEEE Access*, vol. 7, pp. 85000–85023, 2019.

[56] X. Zhang, Q. Yang, J. Ding, and Z. Wang, "Entity profiling in knowledge graphs," *IEEE Access*, vol. 8, pp. 27257–27266, 2020.

LIN ZHU worked at Northeastern University at Qinhuangdao, China. Her research interests include spatiotemporal databases and social networks.

NAN LI was born in 1996. She is currently pursuing the master's degree with Northeastern University at Qinhuangdao, China. Her research interests include fuzzy spatiotemporal data modeling and spatiotemporal data integration.

LUYI BAI received the Ph.D. degree from Northeastern University, China. He is currently an Associate Professor with Northeastern University at Qinhuangdao, China. He has published articles in several journals such as *Integrated Computer-Aided Engineering*, *Fuzzy Sets and Systems*, *Applied Intelligence*, and *Applied Artificial Intelligence*. His current research interests include uncertain databases, fuzzy spatiotemporal XML data management, and knowledge graph. He is also a member of CCF.

YUNQING GONG was born in China, in 1998. She is currently pursuing the bachelor's degree with the School of Computer and Communication Engineering, Northeastern University at Qinhuangdao, China. Her main research interest includes fuzzy spatiotemporal data modeling.

YIZONG XING was born in China, in 1998. He is currently pursuing the bachelor's degree with the School of Computer and Communication Engineering, Northeastern University at Qinhuangdao, China. His main research interest includes fuzzy spatiotemporal data management.