On an Ontological Modeling Language by a Non-Formal Example

© Manuk G. Manukyan
Yerevan State University, Yerevan, Armenia
mgm@ysu.am

Abstract. Principles of an ontological modeling language construction are considered. The proposed modeling language is based on the OPENMath formalism, which is oriented to semantical representation of mathematical objects. We are basing on the concept of so-called content dictionaries of the OPENMath to represent knowledge. An ontology is constructed for Allen's interval temporal logic to show the ontological modeling possibilities of the proposed language. To support this ontology we are using the OPENMath content dictionaries, as well as developing new content dictionaries. Mapping rules from the considered ontology into Datalog are offered.

Keywords: ontology, content dictionary, knowledge base, relative event, temporal relation, XML, OPENMath, Datalog language.

CEUR-WS.org/Vol-2277/paper11.pdf

1 Introduction

The fourth paradigm of science generates necessity to multidisciplinary research in order to support data analysis and its management in various data intensive sciences (such as astronomy and astrophysics, genomics, human brain research, Earth sciences, etc.) [6, 8, 18, 26]. In connection with the appearance of this paradigm the issues of ontological modeling of the subject domains become actual. Ontologies offer means to represent high level concepts, their properties, and their interrelationships. In another words, ontologies are used for formal representation of knowledge of the subject domains. Such representations are used for reasoning about entities of the subject domains, as well as for the domains description.

In this paper we are trying to develop an XML-based ontological modeling language by strengthening the XML language by means of the OPENMath concept [10]. OPENMath is a standard to represent mathematical concepts with their semantics on the Web. Usage of OPENMath concept allows to extend the XML language with computational and ontological constructs. The abilities of ontological modeling of the proposed language will be illustrated by a non-formal example. Namely, we will construct ontology to support Allen's interval temporal logic [2].

We have certain experience of OPENMath usage in our research. Particularly, we proposed a minor extension of OPENMath formalism [19] and used it as kernel of a canonical data model, which also has been developed by us for heterogeneous databases integration [20-23].

The paper is organized as follows: A review of the investigations to ontological modeling is presented in Section 2. Formal bases of the proposed modeling language and Allen's algebra are considered in Section 3. In Section 4 the principles of an ontological modeling language construction are discussed by means of a non-formal example. The mapping rules from XML-based ontology into Datalog are offered in Section 5. The conclusion is provided in Section 6.

2 Related Work

Investigations to support ontology-based information management are intensively developing (for instance, [1, 7, 12, 14, 25]). In [12] an overview of ontology-based data access is provided: a specific paradigm for semantic data integration. An approach to big data integration based on a NoSQL database and modular ontologies is proposed in [1]. A conceptual approach to solve the astronomical problems is offered in [25]. An ontology for Allen's temporal logic is proposed in [7] to support relative time in databases. The problems to support ontological queries are studied in [13, 14]. Namely, two important aspects of this problems: query rewriting and query optimization are discussed.

A good survey of the languages for efficient support of access to the databases satisfying the ontological dependencies can be found in [17]. Particularly, in this paper it is noted that ontological languages and systems are frequently used for representation and support of conceptual schemas over (relational) databases. Such approach to support a concept of databases assumes to use axioms of conceptual schemas and facilities of ontological inference machines upon interpreting queries to databases.

There are different families of ontological languages: graph languages, frame languages, logical languages and rule languages [17]. One of the advanced representatives of ontological languages is OWL, which is based on the description logic [16].

In the context of fourth paradigm of science it is important to provide a high level computationally complete language for ontological modeling. The necessity to develop such languages is connected with the possibility to define the subject domains of research and formulate solutions to scientific problems over

Proceedings of the XX International Conference "Data Analytics and Management in Data Intensive Domains" (DAMDID/RCDL'2018), Moscow, Russia, October 9-12, 2018

abstract specifications. Such approach allows to abstract from resource structure when executing queries to specific data resources. Existing languages are either not computationally complete, or do not provide a high level interface.

3 Formal Bases

In this section we will briefly consider the OPENMath concept. Namely, formalism and constructions on which that concept is based. Thereafter we will discuss Allen's interval temporal logic, which is used to construct an ontology on which the ontological modeling possibilities of the proposed language will be shown.

3.1 The OPENMath Concept

OPENMath is a standard for representation of the mathematical objects, allowing them to be exchanged between computer programs, stored in databases, or published on the Web. The considered formalism is oriented to represent semantic information and is not intended to be used directly for presentation. Any mathematical concept or fact is an example of mathematical object. OpenMath objects are such a representation of mathematical objects which assumes an XML interpretation.

Formally, an OpenMath object is a labeled tree whose leaves are the *basic* OpenMath objects. The compound objects are defined in terms of *binding* and *application* of λ-calculus [15]. The type system is built on the basis of types that are defined by themselves and certain recursive rules, whereby the compound types are built from simpler types. To build compound types the following type constructors are used:

- *Attribution*. If *v* is a basic object variable and *t* is a typed object, then a*ttribution* (*v*, *type t*) is a typed object. It denotes a variable with type *t*.
- *Abstraction*. If *v* is a basic object variable and *t*, *A* are typed objects, then *binding* (λ, *attribution* (*v*, *type t*), A) is a typed object.
- *Application*. If *F* and *A* are typed objects, then *application* (*F*, *A*) is a typed object.

Semantic Level. OPENMath is implemented as an XML application. Its syntax is defined by syntactical rules of XML, its grammar is partially defined by its own DTD. Only syntactical validity of OPENMath objects representation can be provided on the DTD level. To check semantics, in addition to general rules inherited by XML applications, the considered application defines new syntactical rules. This is achieved by means of introduction of *signature files* concept, in which these rules are defined. Signature files contain the signatures of basic concepts defined in some content dictionary and are used to check the semantic validity of their representations. A content dictionary is the most important component of OPENMath concept preservation of mathematical information. In other words, content dictionaries are used to assign formal and informal semantics to all symbols (concepts) used in OPENMath objects. A content dictionary is a collection of related symbols, encoded in XML format and fixing the "meaning" of concepts independently of the application.

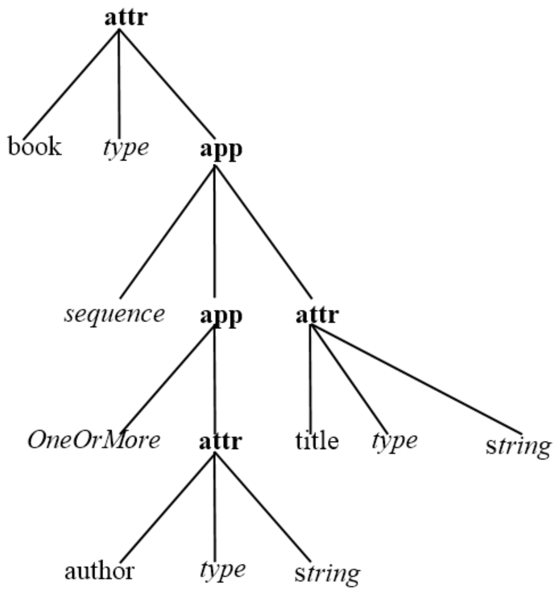

Fig. 1 An example of compound object.

3.2 Allen’s Interval Temporal Logic

The formal framework for our example to ontological modeling is based on Allen's interval temporal logic (for more details see [2-5]) that enables expression of all possible relations between intervals while ensuring computational effectiveness. The basic concepts of the considered formalism are one primitive object, the time interval, and one primitive binary relation: *meets*. A time interval intuitively is the time associated with some event occurring or some property holding in the world. Intuitively, two time intervals t_1 and t_2 meet if and only if t_1 precedes t_2, yet there is no time between t_1 and t_2, and t_1 and t_2 do not overlap. Every other possible relation between two time intervals can be defined in terms of *meets*. As argued in [2] the considered temporal model has several important advantages. In particular, this model allows to represent relative events (for instance, "John married after graduating from school", where it is not known when John married or when he graduated from school). In other words, the considered formalism allows to fix in DB the basic temporal relations of events with possibility to infer implicitly given temporal relations between events.

Temporal Relations. Let *t* be a time interval, then $t-$ and $t+$ denote the lesser and the greater endpoints of *t* correspondingly. Let t_1 and t_2 be time intervals. The following 13 pairwise disjoint basic temporal relations are considered:

$$(t_1\ \mathit{equals}\ t_2) \Leftrightarrow (t_1- = t_2-) \wedge (t_1+ = t_2+)$$

$$(t_1\ \mathit{before}\ t_2) \Leftrightarrow (t_2\ \mathit{after}\ t_1) \Leftrightarrow t_1+ < t_2-$$

$$(t_1\ \mathit{meets}\ t_2) \Leftrightarrow (t_2\ \mathit{metBy}\ t_1) \Leftrightarrow t_1+ = t_2-$$

$$(t_1\ \mathit{overlaps}\ t_2) \Leftrightarrow (t_2\ \mathit{overlappedBy}\ t_1) \Leftrightarrow (t_1- < t_2-) \wedge (t_2- < t_1+) \wedge (t_1+ < t_2+)$$

$$(t_1\ \mathit{during}\ t_2) \Leftrightarrow (t_2\ \mathit{contains}\ t_1) \Leftrightarrow (t_2- < t_1-) \wedge (t_1+ < t_2+)$$

$$(t_1\ \mathit{starts}\ t_2) \Leftrightarrow (t_2\ \mathit{startedBy}\ t_1 \Leftrightarrow (t_1- = t_2-) \wedge (t_1+ < t_2+)$$

$$(t_1\ \mathit{finishes}\ t_2) \Leftrightarrow (t_2\ \mathit{finishedBy}\ t_1) \Leftrightarrow (t_2- < t_1-) \wedge (t_1+ = t_2+)$$

Here, *after* is the inverse of *before*, *metBy* is the inverse of *meets*, *overlapedBy* is the inverse of *overlaps*, *contains* is the inverse of *during*, *startedBy* is the inverse of *starts*, *finishedBy* is the inverse of *finishes*, *equals* is symmetric and transitive.

4 Ontological Modeling Language

The weakness of XML is the absence of data types concept in conventional sense. To eliminate this shortcoming and to support ontological dependencies on the XML level, we expand the XML by means of the OPENMath concept. The considered ontological modeling language coincides with XML which was strengthened by OPENMath concept. OPENMath is an extensible formalism. Its extensibility is achieved by defining new content dictionaries. We propose a minor extension of OPENMath to support the built-in data types concept of the XML Schema [27]. Namely, to model the constants of built-in data types of the XML Schema the corresponding basic objects were introduced. In the context of the considered language we consider three kinds of mechanisms to formalize subject domains:

- content dictionaries to define basic concepts of subject domains;
- signature files to define signatures of basic concepts to check the semantic validity of their representations;
- files of reasoning to formalize knowledge of subject domains. Defining a concept in terms of known ones we introduce a new concept (knowledge) within the considered subject domain. Thus, these files are collections of reasoning rules, which are defining the new concepts in terms of known ones in the considered subject domain.

A content dictionary which contains representation of basic concepts of the subject domain contains two types of information: one which is common to all content dictionaries, and one which is restricted to a particular basic concept definition. Definition of a new basic concept includes name and description of the basic concept, and also some optional information about this concept. Specific information pertaining to the basic concept like the signature and the defining of a concept in terms of known ones is defined in additional files associated with content dictionaries. Content dictionaries contain just one part of the information that can be associated with a basic concept in order to stepwise define its meaning and its functionality. Signature files and files of reasoning are used to formalize the different aspects of subject domains. Namely, to formalize the basic concepts formats, and to define reasoning rules to formalize knowledge of subject domains.

4.1 An Ontology for Allen's Interval Temporal Logic

In [3] an algebra of binary temporal relations on time intervals is proposed for representing qualitative temporal information (i.e., using natural language expressions such as *before*, *after*, *during*), and also the problem of reasoning about such information is considered. With the aim to construct an ontology for Allen's interval temporal logic, we are basing on the definition of Allen's algebra which is proposed in [24]. To express indefinite information, unions of the basic temporal relations are used, which are written as set of basic temporal relations leading to 2^{13} binary *temporal relations*, including the *null* relation $\varnothing$ (also denoted by $\perp$). Let X, Y, Z be time intervals and R, S, T be set of basic temporal relations. Among the considered algebra operands are binary temporal relations, and the operations unary *inverse* (~), binary *intersection* ($\cap$), and binary *composition* ($\circ$), which are defined as follows:

$$\forall X, Y : X\ R^{\sim}\ Y \leftrightarrow Y\ R\ X$$

$$\forall X, Y : X\ (R \cap S)\ Y \leftrightarrow X\ R\ Y \wedge X\ S\ Y$$

$$\forall X, Y : X\ (R \circ S)\ Y \leftrightarrow \exists Z\ (X\ R\ Z \wedge Z\ S\ Y)$$

It follows that the inverse of $R = \{B_1, B_2, \ldots, B_n\}$ can be expressed by the set of basic temporal relations $R^{\sim} = \{B^{\sim}_1, B^{\sim}_2, \ldots, B^{\sim}_n\}$. Further, the intersection of two relations ($R \cap S$) can be expressed as the set-theoretic intersection of the sets of basic relations that are used to describe the temporal relations, i.e.,

$$(R \cap S) = \{B \in \mathbf{B} \mid B \in R \wedge B \in S\}$$

Here, **B** is the set of thirteen basic temporal relations. Finally, the composition of two temporal relations is the union of the component-wise composition of basic temporal relations:

$$R \circ S = \bigcup \{B \circ B' \mid B \in R \wedge B' \in S\}$$

In fact, inferring implied relations and detecting inconsistencies in a set of asserted relations is an NP -

hard problem, but tractable sets (i.e. solvable by polynomial-time algorithms) are known to exist [24]. In other words, tractable subsets of this set that are closed under composition produced a relation also in this subset. Thus, the compositions of triples of relations can be computed from compositions of pairs of relations:

$$R \circ S \circ T \equiv ((R \circ S) \circ T)$$

Inferring implied relations is based on the composition operation. Namely, when a temporal relation *R* holds between time intervals *X* and *Y* and a temporal relation *S* holds between time intervals *Y* and *Z*, then the result of the composition operation of these relations ($R \circ S$) is a possible temporal relation (s), which holds between time intervals *X* and *Z*. Let us note, that composition operation is based on the composition table, which is defined in [3]. Finally, the construction of an ontology for Allen's interval temporal logic is reduced to modeling the considered algebra by the proposed ontological language.

Content Dictionary for Basic Temporal Relations. The considered algebra is based on the basic temporal relations. To formalize the basic temporal relations we developed a new content dictionary named "TempRel" (temporal relation), which contains formal definitions of these relations. Below is the definition of one of them:

```
<CD>
 <CDName> TempRel </CDName>
 <CDUses>
  <CDName> logic1 <CDName>
  <CDName> quant1 <CDName>
 <CDUses>
 <Description>
  This CD defines  symbols  for
  temporal  relations
 </Description>
 <CDDefinition>
  <Name> before </Name>
  <Description>
   A binary relation
  </Description>
  <CMP> before(i,j)≡
        ∃k(meets(i,k)∧ meets(k,j))
  </CMP>
  <FMP>
   <OMOBJ>
    <OMA>
     <OMS name = “equivalent”
          cd = “logic1”/>
    <OMA>
     <OMS name = “before”
          cd = “TempRel”/>
     <OMV name = “i”/>
     <OMV name = “j”/>
    </OMA>
    <OMBIND>
     <OMS name = "exists"
          cd = "quant1"/>
     <OMBVAR>
      <OMV name = "k">
     </OMBVAR>
     <OMA>
      <OMS name = “and”
           cd = “logic1”/>
      <OMA>
       <OMS name = "meets"
            cd = "TempRel"/>
       <OMV name = "i"/>
       <OMV name = "k"/>
      </OMA>
      <OMA>
       <OMS name = "meets"
            cd = "TempRel"/>
       <OMV name = "k"/>
       <OMV name = "j"/>
      </OMA>
     </OMA>
    </OMBIND>
   </OMA>
  <OMOBJ>
 </FMP>
</CDDefinition>

...

</CD>
```

Fig. **2** The temporal relation Content Dictionary File

Here, we used the OPENMath content dictionaries "logic1" and "quant1". In the "logic1" content dictionary the operations of Boolean algebra are defined, and in the content dictionary "quant1" the universal and existential quantifiers are defined. The above used XML elements have obvious interpretations. Only note that the element "CMP" contains the commented mathematical property of the considered basic temporal relation, and the element "FMP" contains the OPENMath representation of this property.

As is mentioned above, to check semantic validity of the basic concepts representations we associate extra information with content dictionaries, namely signature files. A signature file contains the definitions of all the basic concept signatures of the considered content dictionary. Here we use Small Type System [9] to formalize the basic concept signatures. Below is the definition of the signature of the basic temporal relation *before*:

```
<CDSignatures type = "sts"
              cd = "TempRel”>
 ...

 <Signature name = "before">
  <OMOBJ>
   <OMA>
    <OMS name = "mapsto" cd = "sts"/>
    <OMV name = "string"/>
```

```
    <OMV name = "string"/>
    <OMS name = "boolean"
         cd = "logic1"/>
   </OMA>
  </OMOBJ>
 </Signature>

 ...

</CDSignatures>
```

Fig. 3 The temporal relation STS Signature File

Here, Signature introduces a symbol *before* and the *mapsto* symbol is used to construct non-dependent function spaces. The first *n*-1 children denote the types of the arguments, the last one denotes the return type.

Reasoning rules. For modeling Allen's algebra, we developed an XML DTD, the instance of which is an XML file containing the reasoning rules of the considered subject domain. Reasoning rules can be embedded into the ontology based on the content dictionary "logic1" of OPENMath. As we noted above the reasoning rules to support ontology for Allen's interval temporal logic are based on the algebra operations and presented by means of element *rdf* (Rule Definition Formalism). This element contains reasoning rules, each of which defines one of the algebra operations and has two required attributes: *name* and *type*. The value of the attribute *name* is the name of the content dictionary on which the reasoning rules are based. The value of the attribute *type* is the name of the signature file, in which the formats of basic temporal relations are defined. A reasoning rule is defined by means of the *rule* element, which is based on the OPENMath *application object* and has one required attribute *name*. The value of this attribute is the name of the reasoning rule, which coincides with the name of the corresponding algebra operation. Below, DTD for modeling Allen's algebra operations is presented:

```
<!-- include dtd for extended
     OPENMath objects -->

<!ELEMENT rdf (rule)*>

<!ELEMENT rule (OMA)>

<!ATTLIST rdf name #REQUIRED
              type #REQUIRED>

<!ATTLIST rule name(inverse|
           intersection|composition)
           "inverse">
```

Fig. 4 DTD for the Reasoning rules XML Encoding

In case when compositions of relations *R* and *S* generate a single relation *T*, then they are formalized using the "logic1" content dictionary of OPENMath by means of the rules of the following types:

$R(X, Y) \wedge S(Y, Z) \rightarrow T(X, Z)$

The following is an example of such a composition rule:

$\mathit{before}(X, Y) \wedge \mathit{before}(Y, Z) \rightarrow \mathrm{before}(X, Z)$

Below, the XML encoding of this composition rule is presented:

```
<rdf name = "TemRel" type = "sts">

 ...

<rule name = "composition">
 <OMA>
  <OMS name = "implies"
       cd = "logic1"/>
  <OMA>
   <OMS name = "and" cd = "logic1"/>
   <OMA>
    <OMS name="before"
         cd = "TempRel"/>
    <OMV name = "X"/>
    <OMV name = "Y"/>
   </OMA>
   <OMA>
    <OMS name = "before"
         cd = "TempRel"/>
    <OMV name = "Y"/>
    <OMV name = "Z"/>
   </OMA>
  </OMA>
  <OMA>
   <OMS name = "before"
        cd = "TempRel"/>
   <OMV name = "X"/>
   <OMV name = "Z"/>
   </OMA>
  </OMA>
 </rule>

 ...

</rdf>
```

Fig.5 The Reasoning Rules File

In case, when compositions of relations *R* and *S* generates a set of possible basic temporal relations $\{B_1, B_2, \ldots, B_k\}$, then they are formalized using the "logic1" content dictionary of OPENMath `by` means of rules of the following types:

$R(X, Y) \wedge S(Y, Z) \rightarrow (B_1(X, Z) \vee B_2(X, Z) \vee \ldots \vee B_k(X, Z))$

Below is an example of such a composition rule:

$meets(X, Y) \wedge during(Y, Z) \rightarrow (overlaps(X, Z) \vee during(X, Z) \vee starts(X, Z))$

Supporting inverse basic temporal relations involves introducing the following rules in the knowledge base:

$B'(X, Y) \rightarrow B(Y, X)$

$B(X, Y) \rightarrow B'(Y, X)$

Here B is a basic temporal relation, and B' is the inverse relation of B. In this case, we should add to the knowledge base the following reasoning rules for basic temporal relation *before*.

$after(X,Y) \rightarrow before(Y, X)$

$before(X,Y) \rightarrow after(Y, X)$

In addition to the basic temporal relations, we introduce new temporal relations (*before_starts* and *before_ends*) to define the sequence arising of events. Below are the formal definitions of these relations:

$before\_starts(X, Y) \equiv (\exists Z,V)(meets(Z,X) \wedge meets(Z,V) \wedge meets(V, Y))$

$before\_ends(X,Y) \equiv (\exists Z,V)(meets(X, Z) \wedge meets(Z, V) \wedge meets(Y, V))$

In the next section we will consider principles of representation of XML-based ontology in Datalog, in order to convert abstract representations of concepts and their relationships in subject domain to the realization level representations. The choice of Datalog language as a language to support ontology is explained by the fact that this language is one of the best logical formalism for describing knowledge of subject domains.

5 XML-based Ontology Representation in the Datalog

In fact, these two types of predicates are distinguished in Datalog [11]:

- *Extensional* predicates, which are predicates whose relations are stored in a database,
- *Intensional* predicates, whose relations are computed by applying one or more Datalog rules.

The following rules to represent XML-based ontology in Datalog language are proposed:

1. The basic temporal relations are represented by means of extensional predicates. In other words, basic temporal relations are considered as facts, are represented as relations between events, and are stored in an extensional database.
2. The reasoning rules are represented by means of one or more Datalog rules. By means of Datalog rules we are modeling Allen's interval algebra operations. Namely, we model the unary inverse, binary intersection and binary composition operations. These Detalog rules are stored in the intensional database.

Thus, the basic temporal relations are predicates of extensional databases, and the temporal relations are predicates of intensional databases. The following is an example of basic temporal relation: E_1 is an event in which John is married, and E_2 is an event in which John is graduated from school. Then by means of a predicate $after(E_1, E_2)$ we fix in the extensional database the following fact: "John married after graduating from school".

We use Datalog rules to infer implicitly defined information from extensional database. These rules are represented in intensional database by means of a Datalog program. As above mentioned, when compositions of relations R and S generate a single relation T, then we use the following reasoning rule on the level of the ontological modeling language to model such composition operation:

$R(X, Y) \wedge S(Y, Z) \rightarrow T(X, Z)$

The following Datalog representation of the considered reasoning rule is proposed:

$T(X, Z) \leftarrow R(X, Y)\ AND\ S(Y, Z)$

Below is an example of such Datalog rule:

$before(X, Z) \leftarrow before(X, Y)\ AND\ contains(Y, Z)$

In case, when compositions of relations R and S generate a set of possible basic temporal relations $\{B_1, B_2, \ldots, B_k\}$, then it is proposed to use the following reasoning rule on the level of the ontological modeling language to model the considered composition operation:

$R(X, Y) \wedge S(Y, Z) \rightarrow (B_1(X, Z) \vee B_2(X,Z) \vee \ldots \vee B_k(X, Z))$

In this case, we cannot model the considered reasoning rule by means of one Datalog rule, since in Datalog it is not allowed to use disjunctions of atomic formulas as a head of rule. Therefore, we introduce a new temporal relation, which is represented as disjunctions of relations, and whose compositions must also be defined and asserted into the knowledge base. Let the relation D represent the disjunctions of relations $B_1, B_2, \ldots, B_k$, then the composition of relations R and S can be represented in the intensional database as follows:

$D(X, Z) \leftarrow R(X, Y)\ AND\ S(Y, Z)$

The set of possible disjunctions over all basic temporal relations contains 2^{13} relations, but tractable subsets of this set are closed under composition. Introduction of relations of type D is generated by the problem to support such relations. Namely, the set of Datalog rules defining the result of intersection of relations holding between two intervals is required to be introduced in the knowledge base. In other words, all relations which participate in the relations of type D must be represented in the knowledge base by means of such Datalog rules in which these relations are heads of such rules. Let *DOS* represent the disjunctions of relations *during*, *overlaps* and *starts* (see above considered example), then we should add into knowledge base the following Datalog rule:

$$DOS(X, Z) \leftarrow meets(X, Y)\ AND\ during(Y, Z)$$

In addition, we should define and add Datalog rules into the knowledge base to support each relation, which participate in *DOS*. Below is an example of such Datalog rule:

$$during(X, Z) \leftarrow DOS(X, Z)\ AND\ before\_starts(Z, X)$$

Let us note, that the result of the intersection of relation *DOS* with relation *during* is relation *during*:

$$DOS(X, Y) \wedge during(X, Y) \rightarrow during(X,Y)$$

Finally, the intersection of relation *meets* with relation *during* is an empty relation:

$$meets(X, Y) \wedge during(X, Y) \rightarrow \perp$$

Supporting inverse relations is achieved by adding into the knowledge base the considered Datalog rules below:

$$B'(X, Y) \leftarrow B(Y, X)$$

$$B(X, Y) \leftarrow B'(Y, X)$$

Here, B' and B are basic temporal relations. Below are examples of such Datalog rules:

$$after(X, Y) \leftarrow before(Y, X)$$
$$before(X, Y) \leftarrow after(Y, X)$$

In this Section we considered the mapping rules from the proposed ontology into Datalog. Let us note, that when a composition operation is generated by a set of possible basic temporal relations, then this composition operation cannot be directly represented in the Datalog. In this case a necessity to model such reasoning rule by means of a Datalog program arises. In other cases, a direct representation of the reasoning rule by means of a Datalog rule is provided.

6 Conclusion

In this paper an XML-based ontological modeling language is proposed. The proposed language is a result of extension of the XML language with the OPENMath concept. The choice of OPENMath as the basic formalism is explained by the fact that the considered formalism is oriented to semantic representation of mathematical objects. Moreover, that formalism is extensible and provides a rich mathematical apparatus for formalizing the knowledge of the subject domains. The extensibility is achieved by adding new content dictionaries, in which the concepts and reasoning rules for the considered subject domains are defined. Besides the construction of new content dictionaries there is also a possibility to use content dictionaries of OPENMath in which different divisions of computational mathematics are represented, as well as to use the content dictionaries from different subject domains. It is essential that we use a computationally complete language for formalization and systematization of the subject domains knowledge. Thus, a unified interface is provided for representation and management of knowledge from different subject domains.

Acknowledgments

I am grateful to Professor Leonid Kalinichenko for scientific support and collaboration in the field of databases for many years. He represented a whole epoch in databases and his contribution is indisputable and significant.

References

[1] H. Abbes and F. Gargouri. Big data integration: a mongodb database and modular ontologies-based approach. In 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, vol. 96, pp. 446-455. Elsevier, 2016.

[2] J. F. Allen. An interval-based representation of temporal knowledge. In 7th International Joint Conference on Artificial Intelligence, pp. 221-226. 1981.

[3] J. F. Allen. Maintaining knowledge about temporal intervals. Communications of the ACM, 26(11):832-843, 1983.

[4] J. F. Allen. Towards a general theory of action and time. Artificial Intelligence, 23(2):123-154, 1984.

[5] J. F. Allen and P. J. Hayes. A common-sense theory of time. In 9th International Joint Conference on Artificial Intelligence, pp.528-531, 1985.

[6] K. Amunts. The eu's human brain project (hbp) agship - accelerating brain science discovery and collaboration. CEUR-WS, 2022:148-149, 2017.

[7] S. Batsakis, G. Antoniou, and I. Tachmazidis. Integrated representation of temporal intervals and durations for the semantic web. In New Trends in Database and Information Systems II, pp. 147-158,

[8] S. Ceri, A. Canakoglu, A. Kaitoua, M. Masseroli, and P. Pinoli. Data-driven genomic computing: Making sense of signals from the genome. CEUR-WS, 2022:1-2, 2017.

[9] J. H. Davenport. S small openmath type system. ACM SIGSAM Bulletin, 34(2):16-21, 2000.

[10] M. Drawar. Openmath: An overview. ACM SIGSAM Bulletin, 34(2):2-5, 2000.

[11] H. Garcia-Molina, J. Ullman, and J. Widom. Database Systems: The Complete Book. Prentice Hall, 2009.

[12] G. D. Giacomo, D. Lembo, M. Lenzerini, and A. Pogg. Using ontologies for semantic data integration. In A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years, pp. 187-202. Springer, 2017.

[13] G. Gottlob, G. Orsi, and A. Pieris. Ontological queries rewriting and optimization. In ICDE, pp. 2-13, 2011.

[14] G. Gottlob, G. Orsi, and A. Pieris. Ontological query answering via rewriting. LNCS, 6909:1-18, 2011.

[15] J. R. Hindley and J. P. Seldin. Introduction to Combinators and λ-Calculus, Cambridge University Press,1986.

[16] I. Horrocks, P. F. Patel-Schneider, and D. L. McGuinness. Owl: A description logic based ontology language. LNCS, 3709:5-8, 2005.

[17] L. A. Kalinichenko. Effective support of databases with ontological dependencies: Relational languages instead of description logics. Programming and Computer Software, 38(6):315-326, 2012.

[18] G. Longo, M. Brescia, and S. Cavuoti. The astronomical data deluge: the template case of photometric redshifts. CEUR-WS, 2022:27-29, 2017.

[19] M. G. Manukyan. Extensible data model. In Advances in Databases and Information Systems, pp. 42-57, 2008.

[20] M. G. Manukyan. Canonical model: Construction principles. In iiWAS2014, pp. 320-329, ACM, 2014.

[21] M. G. Manukyan. On an approach to data integration: Concept, formal foundations and data model. CEUR-WS, 2022:206-213, 2017.

[22] M. G. Manukyan and G. R. Gevorgyan. An approach to information integration based on the AMN formalism. In First Workshop on Programming the Semantic Web, pp. 1-13, USA, 2012.

[23] M. G. Manukyan and G. R. Gevorgyan. Canonical data model for data warehouse. Communications in Computer and Information Science, 637:72-79, 2016.

[24] B. Nebel and H. J. Burckert. Reasoning about temporal relations: A maximal tractable subclass of allen?s interval algebra. Journal of the ACM, 42(1):43-66, 1995.

[25] N. A. Skvortsov, E. A. Avvakumova, D. O. Bryukhov, A. E.Vovchenko, A. A. Volnova, O. B. Dluzhnevskaya, P. V. Kaygorodov, L. A. Kalinichenko, A. Y. Knyazev, D. A. Kovaleva, O. Y. Malkov., A. S. Pozanenko, and S. A. Stupnikov. Conceptual approach to astronomical problems (in Russian). Astrophysical Bulletin, 71(1):122-133, 2016.

[26] Z. Szallasi. Development of genomic based diagnostics in various application domains. CEUR-WS, 2022:3-4, 2017.

[27] H. Thompson, D. Beech, and et al. XML Schema Part1: Structures. http://www.w3.org, 2001..