Ontop-temporal: A Tool for Ontology-based Query Answering over Temporal Data

Elem Güzel Kalaycı
Free University of Bozen-Bolzano, IT
kalayci@inf.unibz.it

Guohui Xiao*
Free University of Bozen-Bolzano, IT
xiao@inf.unibz.it

Vladislav Ryzhikov
Birkbeck, University of London, UK
vlad@dcs.bbk.ac.uk

Tahir Emre Kalayci
Virtual Vehicle Research Center, AT
emre.kalayci@v2c2.at

Diego Calvanese
Free University of Bozen-Bolzano, IT
calvanese@inf.unibz.it

ABSTRACT

We present *Ontop-temporal*, an extension of the ontology-based data access system Ontop for query answering with temporal data and ontologies. Ontop is a system to answer SPARQL queries over various data stores, using standard R2RML mappings and an OWL 2 QL domain ontology to produce high-level conceptual views over the raw data. The Ontop-temporal extension is designed to handle timestamped log data, by additionally using *(i)* mappings supporting validity time specification, and *(ii)* rules based on metric temporal logic to define temporalised concepts. In this demo we present how Ontop-temporal can be used to facilitate the access to the MIMIC-III critical care unit dataset containing log data on hospital admissions, procedures, and diagnoses. We use the ICD9CM diagnoses ontology and temporal rules formalising the selection of patients for clinical trials taken from the clinicaltrials.gov database. We demonstrate how high-level queries can be answered by Ontop-temporal to identify patients eligible for the trials.

KEYWORDS

Ontology-based data access, metric temporal logic, MIMIC-III

ACM Reference Format:
Elem Güzel Kalaycı, Guohui Xiao, Vladislav Ryzhikov, Tahir Emre Kalayci, and Diego Calvanese. 2018. Ontop-temporal: A Tool for Ontology-based Query Answering over Temporal Data. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269230

1 INTRODUCTION

In real-world applications of various domains it is often necessary to manage temporal information for large amounts of data possibly spread over multiple data sources, also taking into account complex forms of knowledge about the domain of interest. In such setting, temporal reasoning over both data and knowledge has been actively investigated in recent years, e.g., to support turbine diagnosis and meteorological data analysis [4], and clinical decision making (see e.g., [1, 2, 9, 16]).

Ontology-based data access (OBDA) is a popular paradigm to provide a high-level conceptual view of the underlying data [18]. In this paper, we consider the temporal extension of OBDA and present the *Ontop-temporal* system implementing this paradigm. *Ontop-temporal* can be used as a query answering tool in application domains that deal with temporal data.

In the demo, using standard biomedical ontologies, we provide an ontological view over clinical data stored in relational databases, and such view can be queried using the W3C standard query language SPARQL [10]. OBDA systems, such as *Ontop* [6] and morph-RDB [14], have been used to access clinical data, e.g., to identify patients with Type 2 Diabetes Mellitus in Electronic Health Records [15], to support the HL7 Reference Information Model (RIM) [13], and to access the Observational Health Data Science and Informatics (OHDSI) data repositories[1] via the FHIR ontology [11]. To deal with temporal information, we consider a temporal extension of OBDA that supports an expressive rule language [4, 5] for specifying temporal patterns over clinical data. The demo uses the MIMIC-III dataset [12] and the ICD9CM ontology[2]. We present two use cases on *(1)* diagnosis and identifying patients at risk [1, 17], and *(2)* selection of patients for clinical trials [2, 3].

2 OUR TEMPORAL OBDA SOLUTION

In this section we introduce our temporal OBDA framework and its implementation in the *Ontop-temporal* system.

2.1 RDF and OBDA Framework

The vocabulary of RDF [8] consists of three disjoint sets of symbols: IRIs $\mathbf{I}$, blank nodes $\mathbf{B}$, and literals $\mathbf{L}$. An RDF term is an element in $\mathbf{C} = \mathbf{I} \cup \mathbf{B} \cup \mathbf{L}$, an (RDF) *triple* (s, p, o) is an element in $\mathbf{C} \times \mathbf{I} \times \mathbf{C}$, an (RDF) *graph* is a set of triples, and an (RDF) *named graph* is a pair (g, G), where $g \in \mathbf{I}$ and G is an RDF graph.

Let $\mathbf{V}$ be a set of variables, then an element in $(\mathbf{C} \cup \mathbf{V}) \times \mathbf{I} \times (\mathbf{C} \cup \mathbf{V})$ is a *triple pattern*, and a *basic graph pattern* (BGP) is a set of triple patterns. A *filter expression* F is a Boolean combination of expressions $x \circ y$, where $\{x, y\} \subseteq \mathbf{C} \cup \mathbf{V}$ and $\circ \in \{<, >, =\}$.

*Corresponding author

[1] https://www.ohdsi.org/
[2] https://bioportal.bioontology.org/ontologies/ICD9CM

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269230

In standard OBDA [18], an *OBDA specification* $\mathcal{P} = (\mathcal{O}, \mathcal{M}, \mathcal{S})$ consists of an ontology $\mathcal{O}$, a (relational) data source schema $\mathcal{S}$, and a mapping $\mathcal{M}$ from $\mathcal{S}$ to $\mathcal{O}$. The mapping consists of *mapping assertions* of the form $\varphi(\vec{x}) \rightsquigarrow \texttt{sql}(\vec{x})$, where $\texttt{sql}(\vec{x})$ is an SQL query over $\mathcal{S}$ returning columns $\vec{x}$, and $\varphi(\vec{x})$ is a template for a set of triples with $\vec{x}$ as placeholders. Then, an *OBDA instance* is a pair $(\mathcal{P}, \mathcal{D})$, where $\mathcal{D}$ is a relational database instance conforming to schema $\mathcal{S}$.

The semantics of $(\mathcal{P}, \mathcal{D})$ is given by the set of (first-order) models of the OWL knowledge base $(\mathcal{O}, \mathcal{M}(\mathcal{D}))$, where $\mathcal{M}(\mathcal{D})$ is the (virtual) RDF graph generated by applying $\mathcal{M}$ over $\mathcal{D}$.

In OBDA, queries are posed over the ontology $\mathcal{O}$, and are answered by accessing the data in $\mathcal{D}$ through the mapping $\mathcal{M}$. Specifically, given a SPARQL query q, an OBDA system computes its *certain answers*, which are those answers to q that hold in all models of the OBDA instance $(\mathcal{P}, \mathcal{D})$. The computation is done without actually materializing $\mathcal{M}(\mathcal{D})$ (this is why it is *virtual*), but by rewriting q into a query over $\mathcal{S}$ that is then answered by the relational DBMS. We refer to [18] for more details.

2.2 Temporal OBDA

In our approach to temporal OBDA, we extend the RDF data model used for the conceptual level with *(temporal) intervals*. Such an interval τ has the form $\langle t_1, t_2 \rangle$, where '$\langle$' stands for either '[' or '(', respectively specifying that τ is left inclusive or not, and similarly for '$\rangle$'. A *temporal RDF* (t-RDF) *triple* (or simply *t-triple*) is of the form $(s, p, o)@\tau$, stating that the triple (s, p, o) is *valid* in the interval τ. A *t-RDF graph* is a set of t-triples. To stay compatible with the existing RDF/SPARQL infrastructure, we represent a t-RDF graph $\mathcal{G}$ using a set of standard named graphs. Specifically, for each interval τ appearing in $\mathcal{G}$, we use one named graph (g_τ, G_τ), where G_τ contains all RDF triples T such that $T@\tau \in \mathcal{G}$. Moreover, we associate to the IRI g_τ the temporal properties that characterize the interval τ, using the W3C standard Time ontology [7].

A *temporal OBDA specification* extends a classical OBDA specification $(\mathcal{O}, \mathcal{M}, \mathcal{S})$ with a temporal mapping $\mathcal{M}_t$ and a temporal rule component $\mathcal{R}_t$. The *temporal mapping* $\mathcal{M}_t$ consists of a set of *temporal mapping assertions*, each of which is an expression $\varphi(\vec{x})@\langle t_1, t_2 \rangle \rightsquigarrow \texttt{sql}(\vec{x}, t_1, t_2)$, where sql now returns also temporal information, and $\varphi(\vec{x})@\langle t_1, t_2 \rangle$ is a template for a set of t-triples. Intuitively, for each answer tuple $(\vec{a}, b, e)$ to $\texttt{sql}(\vec{x}, t_1, t_2)$, this mapping assertion generates a group of t-triples $\varphi(\vec{a}, b, e)$ sharing the same interval $\langle b, e \rangle$.

We base our temporal rule language on DatalogMTL [4]. The *temporal rule component* $\mathcal{R}_t$ consists of a set of temporal rules. Each *temporal rule* has the form $\alpha \quad \leftarrow \quad \beta_1, \ldots, \beta_n$, where the *head* α is a BGP, and each body atoms β_i is either a filter F or an expression A constructed according to the grammar $A ::= B \mid \top \mid \boxplus_\rho A \mid \boxminus_\rho A \mid \diamondplus_\rho A \mid \diamondminus_\rho A \mid A\, \mathcal{S}_\rho\, A \mid A\, \mathcal{U}_\rho\, A$, where B is a basic graph pattern to which possibly a filter is applied, and ρ denotes a *range*, i.e., an interval with non-negative endpoints. Intuitively, a temporal rule may derive new t-triples from t-triples that have been retrieved from the data or already derived and that satisfy the temporal patterns specified in the rule body. For the formal semantics of the rule language, we refer to [4].

To avoid unwanted interaction between ontological and temporal reasoning, we assume that *(i)* we have two disjoint alphabets Σ_s of static predicates and Σ_t of temporal predicates, *(ii)* the (static) mapping $\mathcal{M}$ populates only predicates in Σ_s, *(iii)* the temporal mapping $\mathcal{M}_t$ populates only predicates in Σ_t, *(iv)* the ontology $\mathcal{O}$ is specified only over predicates in Σ_s, and *(v)* the rules $\mathcal{R}_t$ have in their head only predicates in Σ_t. In this way, the semantics of the temporal OBDA framework can be specified by first retrieving the (static and temporal) data from $\mathcal{D}$, then deriving the (static) consequences through O, and finally carrying out temporal inference through $\mathcal{R}_t$. The formal definition is beyond the scope of this paper.

2.3 The *Ontop-temporal* System

We have implemented the temporal OBDA framework in the system *Ontop-temporal* by extending the OBDA system *Ontop* [6]. The system accepts a temporal OBDA specification as input consisting of *(1)* an *OWL 2 QL* ontology, *(2)* a static mapping, *(3)* a connection to a relational database for extracting the schema information, *(4)* a temporal mapping, and *(5)* a temporal rule component. Since for the latter two components no concrete syntax has been specified so far, we have designed two languages. The temporal mapping language extends the *Ontop* mapping language with a new interval component. The temporal rule language is inspired by SWRL and SPARQL, and we respectively denote the temporal operators $\boxplus$, $\boxminus$, $\diamondplus$, $\diamondminus$, $\mathcal{S}$, and $\mathcal{U}$ as `ALWAYS IN FUTURE`, `ALWAYS IN PAST`, `SOMETIME IN FUTURE`, `SOMETIME IN PAST`, `SINCE`, and `UNTIL`. Concrete examples of temporal mapping and rules are provided in the next section.

The main functionality of the system is SPARQL query answering over a temporal OBDA instance. The system rewrites an input SPARQL query, using the temporal OBDA specification, into a complex SQL query that can be executed over the source database. The algorithm is based on the mapping saturation algorithm used in *Ontop*, and has been extended to deal with the temporal mapping and rule components.

In order to make the system more accessible, we have also extended the *Ontop Protégé* plugin to offer a graphical user interface to edit temporal mapping and rules, and to execute SPARQL queries.

3 THE DEMO

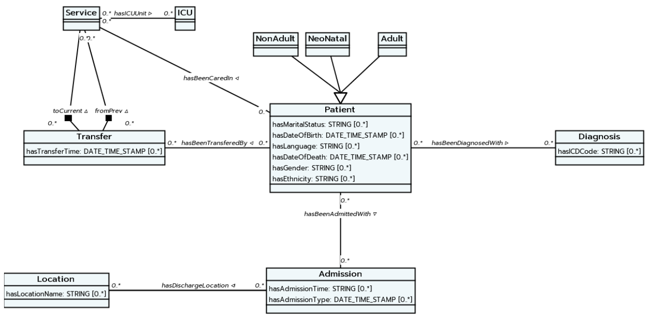
Our demo shows how *Ontop-temporal* helps facilitating clinical decision support. The material to reproduce the demo is online[3].

3.1 Data Preparation

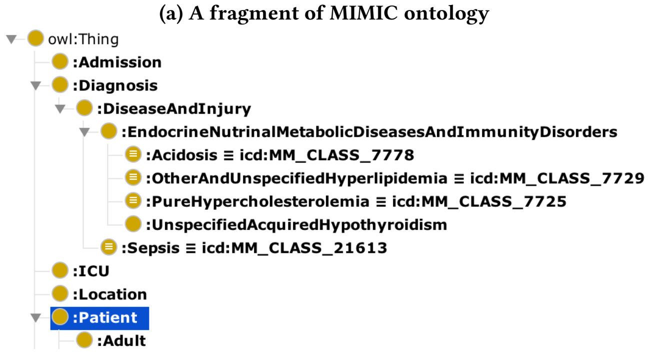
Dataset. The demo is based on the MIMIC-III dataset, a large database from a real hospital with records of 58,000 hospital admissions for 38,645 adults and 7,875 neonates. The raw size of the MIMIC-III dataset is 60 GB on disk in a PostgreSQL database. It comes with a set of sample SQL queries[4] focused on the extraction of temporal data [12]. MIMIC-III has been used to test systems and prototypes that support biomedical temporal reasoning [17].

Ontology. We define a small ontology that models intensive care unit (ICU) stays of patients. The main concepts are shown in Figure 1a. For the taxonomy of Diseases and Injuries we imported the ICD9CM (International Classification of Diseases, Version 9 – Clinical Modification) ontology. Figure 1b shows the equivalences

[3] https://github.com/ontop/ontop-examples/tree/master/cikm-2018-temporal-obda
[4] https://github.com/MIT-LCP/mimic-code



(a) A fragment of MIMIC ontology



(b) Importing ICD9CM ontology

Figure 1: Ontology in our use case

```
mappingId  urn:Map-Adult
target     :Patient/{subject_id} a :Adult; :hasPatientID {subject_id}^^xsd:int.
source     SELECT "patients"."subject_id" FROM "icustays", "patients"
           WHERE "icustays"."subject_id" = "patients"."subject_id"
           AND ("icustays"."intime" - interval '18' YEAR) > "patients"."dob"

mappingId urn:Map-Sepsis
target    :Diagnosis/{icd9_code} a :Sepsis .
source    SELECT "d_icd_diagnoses"."icd9_code" FROM "public"."d_icd_diagnoses"
           WHERE "d_icd_diagnoses"."icd9_code" = '99591'
```

Figure 2: Example static mapping assertions

between the concepts in our ontology (with prefix "`:`") and those in ICD9CM (with prefix "`icd:`").

Static Mapping. We created 10 static mapping assertions to populate the static concepts using the tables of the MIMIC-dataset. Figure 2 shows two example assertions for the `Adults` concept with their `hasPatientID` role and the `Sepsis` diagnosis.

3.2 Use-case 1: Kidney Injury Induced by Sepsis

Here, we aim at identifying patients who had sepsis and potentially suffered from kidney injury as a result. One of the most important indicators of kidney injury is the level of Creatine in the blood of a patient. Based on the SQL views defined in the MIMIC-III dataset for kidney injury diagnosis, it is important to look at the levels of Creatine observed while patients are in the ICU and at the levels measured within 24 hours after the admission. Thus, we make use of the temporal concept `mt:ICUStay`, which indicates the time interval of a patient's stay at the ICU, and of a temporal role `mt:hasCreatinineLevel`, which indicates the Creatine level of a patient (and in this case, the interval consists of one time point). These mapping assertions are shown in Figure 3.

Now, we define the temporal concept `mt:ICUStayFirstDay`, which indicates the period(s) of stay of a patient at the ICU within the first 24 hours after the admission. We also represent Creatine levels

```
mappingId    urn:Map-icustay
target       ms:Patient/{subject_id} a mt:ICUStay .
interval     [{intime}^^xsd:dateTimeStamp, {outtime}^^xsd:dateTimeStamp]
source       SELECT "subject_id", "intime", "outtime"
             FROM "public"."icustays"

mappingId    urn:Map-CreatinineLevel
target       ms:Patient/{subject_id} mt:hasCreatinineLevel {v}.
interval     [ {lab_begin}^^xsd:dateTimeStamp, {lab_end}^^xsd:dateTimeStamp ]
source       SELECT "subject_id", ."valuenum" AS "v",
               "charttime" AS "lab_begin", "charttime" AS "lab_end"
             FROM "public"."labevents"
```

Figure 3: Example temporal mapping assertions

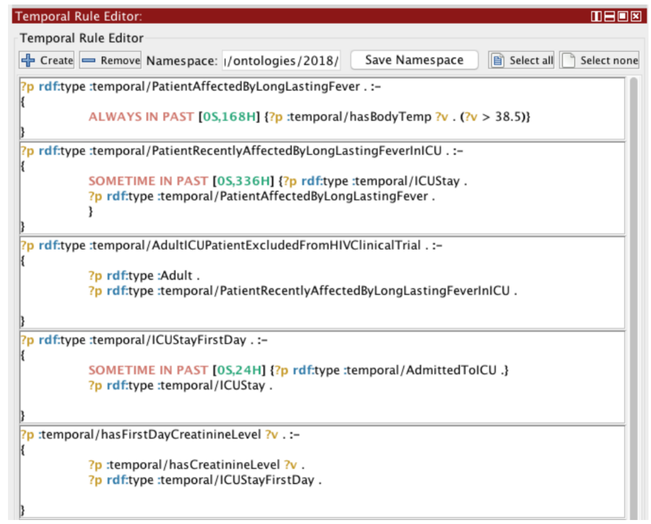


Figure 4: Temporal rules

during this period using the role `mt:hasFirstDayCreatinineLevel`. The temporal rules defining these two temporal predicates are shown in the last two segments of Figure 4. To identify the patients diagnosed with sepsis, we rely on the static concept `Sepsis`.

Finally, we use the SPARQL query depicted in Figure 5 to return Creatine levels for the periods of ICU stay within 24 hours from admission for the patients diagnosed with sepsis. Recall that temporal concepts (e.g., `mt:hasFirstDayCreatinineLevel`) are placed into named graphs (e.g., the one named `?g` in the query in Figure 5), and the information about the interval associated with the graph can be queried using the vocabulary of the W3C Time ontology.

It is known that significantly high levels of Creatine indicate kidney injury, and the levels above 4 are considered significantly high [17]. Thus, the answers returned to our query can help a medical specialist in identifying patients who may have had such injuries as a result of sepsis.

3.3 Use-case 2: Patients Selection for HIV Trials

Here we aim at assisting a medical specialist in selecting patients for trials of medications and procedures. For this use case, we have chosen the HIV medication trial[5]. It is reasonable to conduct this trial among the patients who have a diagnosis in the class `icd:MM_CLASS_4648`, which represents "Endocrine, Nutritional and Metabolic Diseases, and Immunity Disorders", according to the ICD9CM ontology. Then, we can use in our SPARQL queries the

[5]https://clinicaltrials.gov/ct2/show/NCT02355184

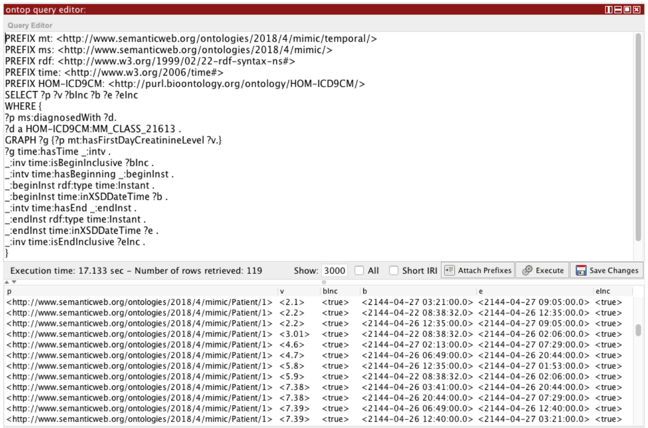


Figure 5: SPARQL query for use-case 1 and its execution

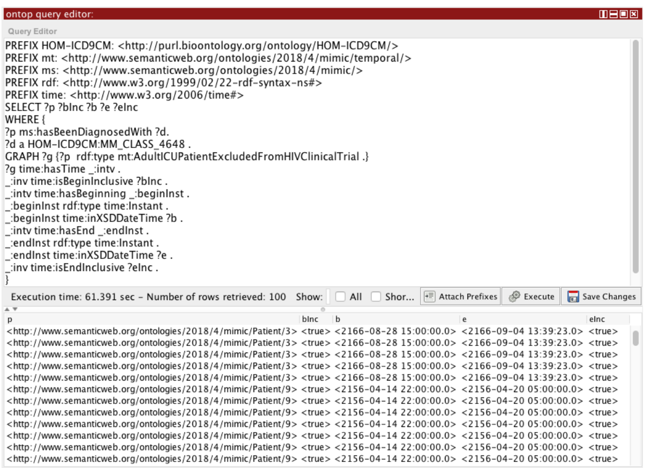


Figure 6: SPARQL query for use-case 2 and its execution

static concept `icd:MM_CLASS_4648`, which we import together with the ICD9CM taxonomy.

Not every patient in our group of interest is eligible, because the HIV medication trial prescribes exceptions. One of such exceptions can be defined by the temporal concept `AdultICUPatientExcludedFromHIVClinicalTrial` (see first three rules of Figure 4). Now, in order to identify the adult patients from the group of interest who are not eligible for the trials, a medical specialist can run the SPARQL query shown in Figure 6, asking for patients having diagnosis of class `icd:MM_CLASS_4648` that are also in `AdultICUPatientExcludedFromHIVClinicalTrial`. With this query, which indirectly accesses the superclass of the other diagnoses, we show that our system is able to combine temporal reasoning with static ontological reasoning.

3.4 Performance

The experiments were run on an AWS server with an Intel Xeon Platinum-8175 processor having 16 logical cores at 2.5 GHz and 64 GB of RAM. During the experiments *Ontop-temporal* translated the SPARQL query into SQL and then executed this SQL translation over the MIMIC-III dataset. For the queries, our system generated SQL of rather complex structure, with 115 and 174 lines respectively for the SPARQL queries used in two use cases. The translation time is negligible. Since the input SPARQL queries are not selective, we run these queries with a limit of 10000. The two SPARQL queries take 251 seconds and 36 seconds respectively.

4 CONCLUSIONS

In this paper, we introduced a tool for practical temporal OBDA, and demonstrated use cases from a clinical domain in order to present the functionalities of *Ontop-temporal*. As future work, we plan to connect *Ontop-temporal* to a big data processing framework (e.g., Spark), which can run the SQL translation in parallel, and evaluate it over large scale heterogeneous time series data. Another direction is to support temporal aggregations at the ontological level. Moreover, we will improve the graphical user interface to visualize the query results with their temporal duration.

Acknowledgements This research is supported by the projects OBATS and QUADRO of the Free University of Bozen-Bolzano.

REFERENCES

[1] Juan Carlos Augusto. 2005. Temporal Reasoning for Decision Support in Medicine. *Artificial Intelligence in Medicine* 33, 1 (2005), 1–24.

[2] Franz Baader, Stefan Borgwardt, and Walter Forkel. 2018. Patient Selection for Clinical Trials Using Temporalized Ontology-Mediated Query Answering. In *WWW Companion Volume*. ACM, 1069–1074.

[3] Paolo Besana, Marc Cuggia, Oussama Zekri, Annabel Bourde, and Anita Burgun. 2010. Using Semantic Web Technologies for Clinical Trial Recruitment. In *Proc. ISWC (LNCS)*, Vol. 6497. Springer, 34–49.

[4] Sebastian Brandt, Elem Güzel Kalayci, Roman Kontchakov, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. 2017. Ontology-Based Data Access with a Horn Fragment of Metric Temporal Logic. In *Proc. AAAI*. 1070–1076.

[5] Sebastian Brandt, Elem Güzel Kalaycı, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. 2018. Querying Log Data with Metric Temporal Logic. *J. of Artificial Intelligence Research* (2018).

[6] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. 2017. Ontop: Answering SPARQL Queries over Relational Databases. *Semantic Web J.* 8, 3 (2017), 471–487.

[7] Simon Cox and Chris Little. 2017. *Time Ontology in OWL*. W3C Recommendation. W3C. Available at http://www.w3.org/TR/owl-time/.

[8] Richard Cyganiak, David Wood, and Markus Lanthaler. 2014. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation. W3C.

[9] Claudio Daza, Josefa Santa Maria, Ignacio Gomez, Mario Barbe, Javier Trincado, and Daniel Capurro. 2016. Phenotyping Intensive Care Unit Patients Using Temporal Abstractions and Temporal Pattern Matching. In *Proc. of the 7th Int. Conf. on Bioinformatics, Computational Biology, and Health Informatics*. ACM.

[10] Steve Harris and Andy Seaborne. 2013. *SPARQL 1.1 Query Language*. W3C Recommendation. W3C. Available at http://www.w3.org/TR/sparql11-query.

[11] Guoqian Jiang, Guohui Xiao, Richard C. Kiefer, Eric Prud'hommeaux, and Harold R. Solbrig. 2017. Building an FHIR Ontology-based Data Access Framework with the OHDSI Data Repositories. In *Proc. AMIA*. AMIA.

[12] Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, H. Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. MIMIC-III, a Freely Accessible Critical Care Database. *Scientific Data* 3, 160035 (2016).

[13] Freddy Priyatna, Raúl Alonso-Calvo, Sergio Paraiso-Medina, and Óscar Corcho. 2017. Querying Clinical Data in HL7 RIM Based Relational Model with morph-RDB. *J. Biomedical Semantics* 8, 1 (2017), 49:1–49:12.

[14] Freddy Priyatna, Óscar Corcho, and Juan F. Sequeda. 2014. Formalisation and Experiences of R2RML-based SPARQL to SQL Query Translation using morph. In *Proc. WWW*. ACM, 479–490.

[15] Alireza Rahimi, Siaw-Teng Liaw, Jane Taggart, Pradeep Ray, and Hairong Yu. 2014. Validating an Ontology-based Algorithm to Identify Patients with Type 2 Diabetes Mellitus in Electronic Health Records. *Int. J. of Medical Informatics* 83, 10 (2014), 768–778.

[16] Yuval Shahar. 1997. A framework for knowledge-based temporal abstraction. *Artificial Intelligence* 90, 1 (1997), 79–133.

[17] Eitam Sheetrit, Nir Nissim, Denis Klimov, Lior Fuchs, Yuval Elovici, and Yuval Shahar. 2017. *Temporal Pattern Discovery for Accurate Sepsis Diagnosis in ICU Patients*. CoRR Technical Report abs/1709.01720. arXiv.org e-Print archive.

[18] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. 2018. Ontology-Based Data Access: A Survey. In *Proc. IJCAI*. AAAI Press.