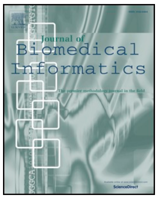
Journal of Biomedical Informatics 100 (2019) 103314

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin



A method for the graphical modeling of relative temporal constraints

Sebastian Mate[a,*], Thomas Bürkle[b], Lorenz A. Kapsner[a], Dennis Toddenroth[c], Marvin O. Kampf[a], Martin Sedlmayr[d], Ixchel Castellanos[e], Hans-Ulrich Prokosch[a,c], Stefan Kraus[c]

[a] Medical Centre for Information and Communication Technology, Universitätsklinikum Erlangen, Erlangen, Germany
[b] Bern University of Applied Sciences, Biel, Switzerland
[c] Chair of Medical Informatics, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
[d] Institute for Medical Informatics and Biometry, Carl Gustav Carus Faculty of Medicine, Technische Universität Dresden, Dresden, Germany
[e] Department of Anesthesiology, Universitätsklinikum Erlangen, Erlangen, Germany

ARTICLE INFO

Keywords:
Temporal queries
Phenotyping
Phenotyping algorithms
Patient cohort identification
Data integration
Data retrieval

ABSTRACT

Searching for patient cohorts in electronic patient data often requires the definition of temporal constraints between the selection criteria. However, beyond a certain degree of temporal complexity, the non-graphical, form-based approaches implemented in current translational research platforms may be limited when modeling such constraints. In our opinion, there is a need for an easily accessible and implementable, fully graphical method for creating temporal queries. We aim to respond to this challenge with a new graphical notation. Based on Allen's time interval algebra, it allows for modeling temporal queries by arranging simple horizontal bars depicting symbolic time intervals. To make our approach applicable to complex temporal patterns, we apply two extensions: with *duration intervals,* we enable the inference about relative temporal distances between patient events, and with *time interval modifiers*, we support counting and excluding patient events, as well as constraining numeric values. We describe how to generate database queries from this notation. We provide a prototypical implementation, consisting of a temporal query modeling frontend and an experimental backend that connects to an i2b2 system. We evaluate our modeling approach on the MIMIC-III database to demonstrate that it can be used for modeling typical temporal phenotyping queries.

1. Introduction

The identification of patient cohorts in electronic patient data, e.g. for clinical studies, can be a difficult task. Several authors (e.g. [1,2]) have reported challenges in determining whether a patient is eligible to be included into a research cohort or not. Finding such cohorts in electronic health record (EHR) data is generally referred to as *EHR-based phenotyping* because researchers aim to identify subjects by phenotypic traits [3–5]. According to Denny [6] and Ross [7], these include demographics, diagnoses, treatments and interventions, laboratory measurements, vital signs, radiological or pathological findings, drugs, and other information recorded in medical documentation. As described by Rea et al. [1], phenotyping plays an important role in the support of clinical studies, in the collection of quality metrics, in outcomes research, in observational studies, in decision support, and in many other tasks.

The set of rules for identifying a specific patient cohort is also called a *phenotyping algorithm* [3]. It consists of several inclusion and exclusion criteria coupled by Boolean logic. Mo et al. [2] mentioned that phenotyping algorithms may include operations beyond Boolean logic, such as numerical comparisons, arithmetic calculations, aggregation operations, negation, and temporal relationships between events. Ross et al. [7] found that 85% of the inclusion criteria of clinical trials had significant semantic complexity and 40% relied on temporal data. Similarly, Conway et al. [8] noticed complex temporal logic in most algorithms of the *Electronic Medical Records and Genomics* (eMERGE) network. Consequently, software designed to support EHR-based phenotyping should also enable the retrieval of such temporal relationships.

In recent years, various research-driven projects, such as i2b2 [9], developed translational research platforms for the selection of patient cohorts. However, as we will describe in more detail in the background section, these systems may offer only limited functionality for temporal queries. We assume that the root cause for this is the lack of an easily accessible and implementable, generic and fully graphical method for temporal queries, which can be integrated into such systems.

In this paper, we describe a straightforward, yet expressive and fully graphical notation for manually modeling temporal queries, such as

---

[*] Corresponding author.
E-mail address: sebastian.mate@uk-erlangen.de (S. Mate).

https://doi.org/10.1016/j.jbi.2019.103314
Received 16 March 2019; Received in revised form 13 August 2019; Accepted 14 October 2019

“List all patients who developed diabetes *during* pregnancy”. We model temporal patterns by arranging horizontal bars on a drawing canvas in accordance with Allen’s time interval algebra [10]. We extend this basic approach using *duration intervals* to allow for modeling complex relative temporal constraints. To address further requirements of temporal phenotyping algorithms, such as counting or excluding patient events, or constraining numeric values, we propose *time interval modifiers*. We then describe the basic steps necessary to translate this notation into executable database queries.

We present a prototypical software implementation, consisting of two open-source tools. The first one, *AllenGUI*, is a direct implementation of our query notation and can be used by a researcher to compose temporal query patterns. The second one, *AllenSPARQL*, is an experimental data integration and query execution backend that connects to an i2b2 system and executes the temporal queries. We evaluate our approach by replicating the temporal queries from Nigrin and Kohane published in [11] on the MIMIC-III [12] database. Finally, we discuss the advantages and limitations of our approach and compare it with similar research.

2. Background

2.1. Temporal patterns in biomedical data

Medicine and time are inextricably linked, as biology, pathogenesis, diagnosis, and treatment follow complex time-oriented processes. For example, it is well known that biology adheres to temporal rhythms [13] and that infectious diseases progress according to patterns in time and space [14]. Diagnosis and treatment are usually performed in compliance with clinical guidelines [15] that define sequences of action for the guidance of physicians. Similarly, cause and effect play a pivotal role in medicine. For example, it is known that the treatment of an ischemic stroke with thrombolysis may lead to cerebral hemorrhage [16].

Biomedical research aims to uncover such cause-effect mechanisms, and patient data plays an indispensable role in this task. The treatment of a patient generates huge amounts of time-oriented electronic data. As described by Moskovitch and Shahar [17], these can be present in the form of simple time-stamps (e.g., *“Hemoglobin value of 9.3 g/100 cc, at 9:05 am, on July 17th, 1998”*) or time intervals (e.g., *“Administration of a medication for 4 days”*). Time intervals can be part of the original data or abstractions of it, e.g. *“3 weeks of severe anemia”*. This abstraction could have been derived via a temporal pattern, such as *“Hemoglobin < 10 g/dL, measured each day for three weeks”*. These temporal patterns are also called *time intervals related patterns* (TIRPs) in the literature [17].

The identification of yet unknown TIRPs, hidden in the original patient data, could support the discovery of previously unknown mechanisms of cause and effect. Therefore, on the one hand, a number of research activities are devoted to the formal description of temporal patterns (e.g. [18–23]) and their automatic recognition in structured data or clinical texts (e.g. [17,24–32]). On the other hand, clinical study protocols and phenotyping algorithms already provide a rich source of known temporal patterns [7,8]. We would like to emphasize that our work is not concerned with the automatic recognition of temporal patterns, but with the graphical modeling of already known patterns and their transformation into executable database queries.

2.2. Querying temporal databases

Unfortunately, temporal relationships in data can only be queried inefficiently with common database query languages. As described by Weng et al. in [33], languages such as SQL support only simple time or date data types, but lack special functionality for sophisticated temporal reasoning. Starting in the 1980s, this has led to the development of temporal extensions, such as *TSQL* [34], *TQuel* [35], the *Historical Relational Data Model* [36], or *TSQL2* [37]. However, such early query languages were too abstract and formal to be used by clinical users [38], had a high learning curve due to their unique syntaxes, concepts, and limitations [39,40], or required rich temporal descriptions that were not common in clinical databases [41].

Consequently, many efforts were made to encapsulate temporal query languages into user-friendly clinical applications, as demonstrated e.g. with the integration of the *AMAS* query language [38] into the *ArchiMed* system [42]. Other examples are the *Chronus* system [43], which implements a temporal language called *TimeLine SQL*, or its successor, *Chronus II* [44], which uses a subset of TSQL2. The *DXtractor* system [11,45] uses textual “atomic” queries to compose expressive temporal patterns.

More modern systems include functionality for the interactive visualization, navigation, and temporal abstraction. As outlined above, this is the process of abstracting simple timestamp data into high-level concepts, such as “high fever for a week” [46]. Popular examples of such systems include *KNAVE* [47], *KNAVE-II* [48], and *VISITORS* [49,50]. KNAVE is built upon a temporal middle layer called *Tzolkin* [51], which utilizes the above-mentioned Chronus system, and *RÉSUMÉ*, a knowledge-based temporal abstraction framework [52]. The later KNAVE-II uses a temporal abstraction framework called *IDAN* [53], the *ALMA* temporal service, and a temporal query language called *TAR* [54]. The KNAVE-II interface was mainly geared towards health care providers aiming to view data of individual patients at the point of care [55]. As such, its successor, VISITORS, was extended to support the visualization of groups of time-oriented records [49]. Because such systems focused on the visualization and interactive exploration of time-stamped data, the direct formulation of temporal patterns played a subordinate role. For example, the *LifeLines2* system [56] allows a researcher to query for event sequences, but it does not support the definition of temporal relationships among events [57].

Rind et al. [58] found that systems, which encapsulate query languages with graphical user interfaces, are easier to learn, but often constrain the expressiveness of query languages in order to keep the user interface manageable. Consequently, systems that provide advanced temporal expressiveness utilize a text-based or form-based approach. However, their notations and workflows may lead to queries that are difficult to understand. Nigrin and Kohane [11] noted that their text-based DXTractor system requires complex steps to formulate queries and that “it is often not immediately obvious how to generate […] results in DXtractor, primarily because of the many steps required” [11]. Form-based approaches are used e.g. in *ChronoMiner* [59] and the above-mentioned *VISITORS* system [49]. Similarly, *PatternFinder* [60] uses “form-fill-in elements that support a rich set of pattern queries” and dialog boxes to model temporal relationships [61]. According to [57] and [58], the system can construct complex temporal patterns, but is also complex to use.

According to our understanding, there are only a few fully graphical approaches which offer high temporal expressiveness and which specifically deal with the manual modeling of temporal query patterns. One of these is the *QueryMarvel* temporal language [57] as implemented in *VizPattern* [62]. It uses a comic strip metaphor to model temporal constraints. Another graphical approach has been developed by Chittaro et al. [63]. It uses a paint strip metaphor to graphically model temporal patterns on a drawing surface and even enables expressing temporal indeterminacy. A later work, based on [63] and presented in [64], introduced support for queries with different temporal granularities and other features. As we will discuss at the end of this paper, these systems offer a high degree of temporal expressiveness, but at the same time we assume that they are complex implementations that cannot easily be adopted.

2.3. Translational research platforms

The widespread adoption of electronic health records (EHRs) in the last decade has led to the development of several translational research platforms. Examples include *Informatics for Integrating Biology and the Bedside* (i2b2) [9,65], *Electronic Health Records for Clinical Research* (EHR4CR) [66–69], *TRANSFoRm* [70–72] and *Observational Health Data Sciences and Informatics* (OHDSI) [73]. Other similar platforms were reviewed by Xu et al. in [74].

In contrast to the temporal querying systems from above, these translational research platforms aim to enable researchers to query clinical data without the need for advanced programming skills. Used across institutions in research networks, they are a critical component in translational research, as demonstrated e.g. in [66,67,75–86].

Multiple publications (e.g. [18,87–93]) have highlighted the significance of querying time-oriented data, and most of the above-mentioned research platforms offer at least some temporal functionality. The i2b2 system, for example, stores time ranges associated with patient facts using the *Start_Date* and *End_Date* columns in its *Observation_Fact* table [9]. Likewise, the OHDSI OMOP database schema captures temporal information using timestamps. Abstractions that were derived from the raw patient data can be stored in separate tables and are referred to as “eras” [28,29,94]. In terms of modeling and executing temporal queries, the i2b2 webclient uses a form-based approach to define temporal relationships between patient events [95]. The *ATLAS* tool from OHDSI [96] uses a similar approach. The EHR4CR system uses a clinician-readable and computational, text-based notation called ECLECTIC [66,97].

The literature describes difficulties that occur when representing temporal queries in such translational platforms [98], which can possibly be attributed to the way how temporal queries are modeled in these systems. Even the i2b2 developers noted in their documentation that the section of the user interface for modeling temporal criteria is “a bit more complicated” than the one for the non-temporal criteria [95]. According to [62], “it is often hard for users to represent [temporal] queries using forms.” It would therefore seem reasonable to investigate how such translational research platforms could be extended with easy-to-use, yet powerful temporal search functions.

2.4. Allen’s time interval algebra

A groundbreaking work in the field of temporal reasoning is *Allen’s time interval algebra* [10], which describes temporal relationships between time intervals in terms of their chronological sequence. In this context, a time interval is a time range formed by two timestamps, one for the start and one for the end. The temporal relationship between two time intervals is expressed by *Allen relations*, of which there are 13 in total. If two intervals are linked via an Allen relation, we speak of an *Allen statement*. Fig. 1 shows the 14 possible Allen statements between two time intervals, which can be generated using the 13 Allen relations. Allen’s relations could be described as consisting of seven *basic* relations, of which six have an inverse (note that *equals* is symmetric). Allen relations can be formally described by the chronological sequence of the start and end times of both time intervals, as shown in Fig. 1.

From the perspective of computer science, a group of Allen statements is a directed named graph, with the nodes representing the time intervals and the edges representing the temporal relationships. In a single Allen statement, multiple relations can be grouped to form a composite relation, which allows for expressing temporal indeterminacy. For example, “C {*starts, started by, equals*} D” states that “C *starts* D”, “C *started by* D”, “C *equals* D”, or any combination of these statements is true. Finally, Allen’s constraint propagation technique, which makes use of transitivity rules, can be used to infer implicit knowledge about relationships between time intervals. For example, from “E *before* F” and “F *before* G” one can deduce “E *before* G”.

3. Methods

3.1. Graphical modeling approach

For clinical data analysis precise definitions of chronological event sequences are critically important, e.g. for the analysis of risk factors or therapeutic utility. For example, a researcher may search for patients with a specific sequence of drug administrations prior to an intervention, or patients, where the administration of a specific drug overlaps the administration of another. In these examples, temporal relations, such as *overlaps*, *before*, or *contains* would be useful to formulate temporal queries.

Since Allen’s time interval algebra seems to intuitively describe such temporal relationships, we investigated how we could exploit his formalism for a graphical notation. We intended to investigate whether it is possible to concatenate multiples of his “pictorial examples” (as in Fig. 1) to create complex search patterns.

Following this basic idea, we position horizontal bars on a two-dimensional drawing canvas to symbolize time intervals. The x-axis on the canvas represents the flow of time: the left vertical side of a bar depicts the interval’s start time, the right side its end time. Whenever two time intervals touch each other, be it vertical or horizontal, a temporal relationship in the form of an Allen statement is created. The whole set of these statements, implicitly combined by the Boolean operator “AND”, describes the temporal properties a patient has to fulfill in order to be eligible to be included into the resulting patient cohort. This is equivalent to a TIRP [17]. Note that we do not address modeling Boolean logic (as e.g. in the “logical area” in [64]), because we consider this functionality already well integrated in current phenotyping platforms.

In our approach we distinguish between two types of time intervals: *symbolic time intervals* (as also defined in [17]) and *duration intervals*. The first type deals with the representation of phenotypic concepts, whereas the second type is used to constrain the lengths of time intervals, to model relative temporal distances or temporal gaps between time intervals of the first type.

3.1.1. Symbolic time intervals

With regard to symbolic time intervals, each bar depicts a time interval that itself refers to an event[1] or an episode in the patient’s history. It symbolizes a phenotypic concept, such as a diagnosis, a treatment, a laboratory measurement, etc. By combining such time intervals, simple temporal search patterns can be modeled.

In the upper example of Fig. 2, the two symbolic time intervals “Diabetes” and “Pregnancy” are arranged according to the *during* relation (compare Fig. 1). The modeled Allen statement in this depiction therefore is “Diabetes *during* Pregnancy”. This query aims to identify all patients who developed diabetes during pregnancy. Note that the time scale is not linear; therefore the length of a bar does not specify the duration of the depicted time interval. This becomes possible with the definition of duration intervals as described in the next section.

The Allen relations *before* and *after* require a slightly different graphical notation, because time intervals connected via these relations do not touch. We alleviate this problem by using a horizontal connector line, as already indicated in Fig. 1 and as demonstrated in the second example of Fig. 2. This query searches for all patients who were treated with thrombolysis after an ischemic stroke and who experienced cerebral bleeding after this treatment.

---

[1] We actually convert patient data, which only exist as simple time stamps and not as time intervals, e.g. laboratory measurements and diagnoses, into 1-second time intervals as a preprocessing step. This is a workaround to make the data to be queried fully compatible with Allen’s time interval-based algebra. We tolerate the fact that this is not entirely clean from a formal point of view as it simplifies a later implementation considerably.

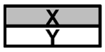
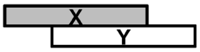
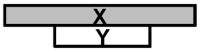
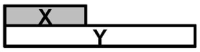
Allen Statements with …		Pictorial Example	Chronological Sequence
Basic Relations	Inverse Relations		
X *before* Y	Y *after* X		$X_{start} < X_{end} < Y_{start} < Y_{end}$
X *equals* Y	Y *equals* X		$X_{start} = Y_{start} < X_{end} = Y_{end}$
X *meets* Y	Y *met by* X		$X_{start} < X_{end} = Y_{start} < Y_{end}$
X *overlaps* Y	Y *overlapped by* X		$X_{start} < Y_{start} < X_{end} < Y_{end}$
X *contains* Y	Y *during* X		$X_{start} < Y_{start} < Y_{end} < X_{end}$
X *starts* Y	Y *started by* X		$X_{start} = Y_{start} < X_{end} < Y_{end}$
X *finishes* Y	Y *finished by* X		$Y_{start} < X_{start} < X_{end} = Y_{end}$

Fig. 1. Allen's time interval algebra. The table shows the 13 possible Allen relations between the two time intervals X and Y, their pictorial notation and the chronological sequence of the start and end times of the intervals. The dashed connector line (at the *before* and *after* depiction) is a modification that we require in Section 3.1.

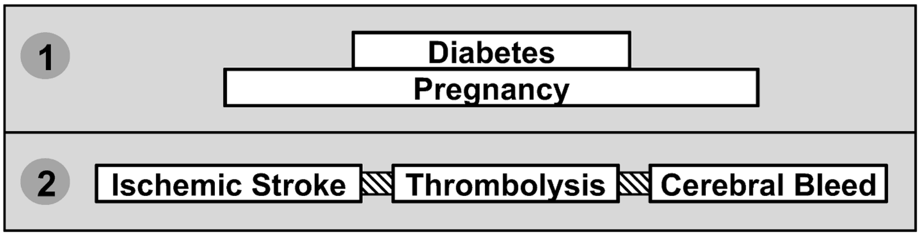


Fig. 2. Two simple temporal queries with symbolic time intervals. The first query aims to identify all patients who developed diabetes during pregnancy. The second query searches for all patients who were treated with thrombolysis after an ischemic stroke and who experienced cerebral bleeding afterwards.

3.1.2. *Duration intervals*

To link more time intervals to create expressive and complex queries, we first need to integrate the temporal construct *duration.* It is similar to the duration data type of the *Arden Syntax*, which "is a length of time that is not anchored to a particular point in time" [99]. Duration is not part of Allen's time interval algebra [100], but support for it can be integrated into our approach via a second time interval type, which we call *duration intervals*. These are similar to the symbolic time intervals described above, but refer to time spans instead of phenotypic concepts. They may be composed using seconds, minutes, hours, days, months and years. For example, valid time spans are "2 Weeks" or "1 Year 3 Months". The distinction on the technical level is that symbolic time intervals have associated start and end times in the patient data, whereas duration intervals are only constructs of the temporal query pattern.

Four examples with duration intervals are shown in Fig. 3. The first query illustrates how duration intervals can be used to constrain the length of a symbolic time interval. In this example, the surgery must have lasted at least 3 h, but not more than 6 h. The second example extends the second query from Fig. 2 by stating that all events must have occurred within 4 days. Without this constraint, the three events could theoretically be years apart. The third query demonstrates the basic approach of modeling explicit temporal gaps. It aims to find patients who showed AIDS symptoms more than 10 years after their HIV diagnose. This is achieved via the dashed connector bar between "10 Years" and "AIDS", modeling the *after* Allen relation. If the two intervals on the right, however, would *meet* as the two on the left, the query would search for patients who developed AIDS *exactly* 10 years after the HIV diagnose. A more complex example, which models relative temporal constraints, is given in the fourth pattern. It aims to identify all patients who had a chemotherapy finishing at least six but at most ten weeks before surgery. The chemotherapy must have started at least ten weeks before surgery, and patients must have been given the medication mitomycin in the six weeks before surgery. Note that "4 Weeks" and "6 Weeks" together form the ten-weeks interval.

3.1.3. *Multiple-referencing of time intervals*

If a user aims to add more bars, others may already occupy the space on the drawing area. We solve this problem by allowing multiple references to time intervals. Such a reference is created whenever a bar uses a label that has already been used for another one. This type of referencing allows the researcher to add more time intervals to a temporal query, even if there is no space left at the current position on the drawing canvas. An example will be given later in in Fig. 7.

3.1.4. *Time interval modifiers*

As outlined in the introduction, exclusion, event counting, numerical restriction and relative comparisons are other important aspects of phenotyping algorithms. For example, a researcher might be interested in counting the number of drug applications during a chemotherapy treatment. Another researcher might want to constrain a query by laboratory values, or specifically exclude patients who had a certain medical condition. We address these aspects with a concept that we call *time interval modifiers*. These specify a time interval more precisely with additional constraints and thereby allow for excluding or counting events and for constraining numerical values. The modifiers are prefixed and suffixed to the medical concept that is associated with the time interval when building the search pattern. There are three types of modifiers:

- **Occurrence Modifier (prefixed):** Can either be "NO" for exclusion, or a numeric comparison, such as "≥10", to count occurrences.

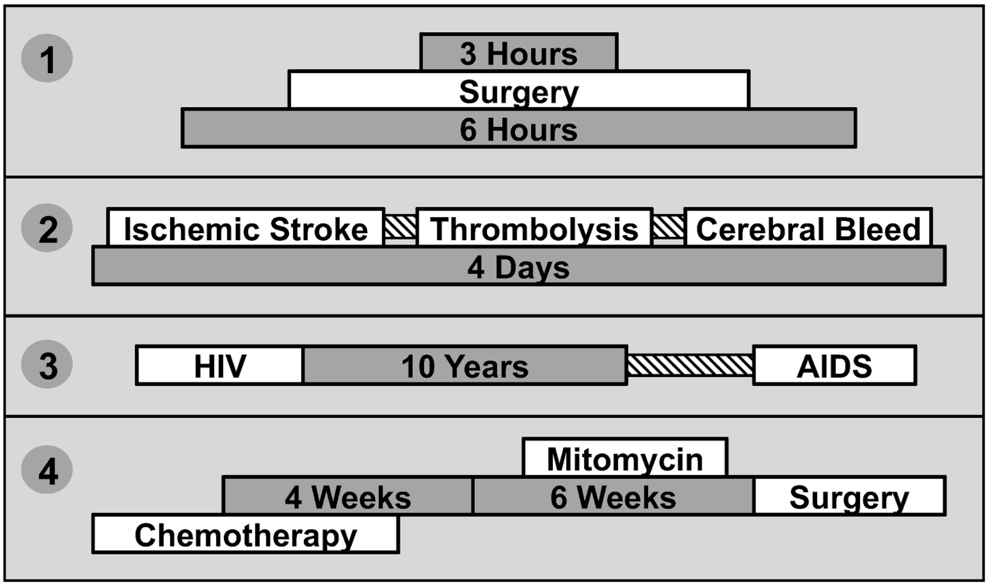


Fig. 3. More complex temporal queries with symbolic time intervals (white) and duration intervals (gray). The first query demonstrates how to use duration intervals for constraining the lengths of symbolic time intervals. The second one restricts the events from the second example from Fig. 2 to less than 4 days. The third one aims to find all patients who developed AIDS more than 10 years after their HIV diagnose. The fourth query demonstrates the use of duration intervals for the specification of relative temporal distances between symbolic time intervals.

- **Value Modifier (suffixed):** A value comparison that constrains the value of a numerical concept, such as a laboratory value.
- **Enumeration Modifier (suffixed):** A number behind a hash (e.g. "#3"), used to distinguish individual instances of time intervals of the same medical concept.

As an example, consider the query pattern in Fig. 4. It aims to identify all patients experiencing lactic acidosis as a delayed side effect of metformin, even though the latter was not administered in the 48 h before surgery [101]. As an indicator of acute acidosis, the right part of the query searches for two or more pH measurements <7.35 during surgery [102]. As no metformin must have been taken in the 48 h before surgery, we exclude this medication with the "NO" occurrence modifier during the interval "48 Hours #2". However, to ensure that the patients are actually metformin patients, we require them to have received the medication in the two days before ("48 Hours #1"). The two enumeration modifiers "#1" and "#2" indicate that the two depictions are referring to two different 48-h duration intervals.

To finish the description of our graphical notation, we aim to emphasize that it is capable of handling any number of time intervals and all types of Allen relations. It is unambiguous because the chronological sequence of all start and end times (as in Fig. 1, "Chronological Sequence") is preserved in each set of fully interconnected time intervals on the drawing canvas.

3.2. Translation into database queries

We will now illustrate how to create database queries from our graphical notation. Note that the database schema used below is overly simple for the sake of clarity. It assumes that all attributes are stored in separate columns of a single table. To be generic and independent of the database schema, most research systems (e.g., i2b2) model their data according to the Entity-Attribute-Value (EAV) schema [103]. For this reason, a real implementation of our approach has to consider the peculiarities of a given database schema.

3.2.1. The query graph

By parsing our graphical notation, one can derive a *query graph,* which can serve as a formal temporal query definition that can be translated into a database query. If one examines all pairs of directly touching bars, the Allen relations can be obtained easily by referring to Allen's depictions in Fig. 1. These Allen statements form a directed named graph, as shown in Fig. 5 with the *solid edges* for the fourth example of Fig. 3. Allen's constraint propagation technique can then be used to derive the additional Allen statements between all time intervals that do not touch directly, as shown with the *dashed edges*. For example, it was deduced that the chemotherapy is before the surgery (Allen statement 7). The result is a graph with $\frac{n(n-1)}{2}$ edges (= Allen statements), where *n* denotes the number of nodes (= time intervals).

A graphical query pattern can be translated into a database query by processing the Allen statements from the inferred query graph. For an SQL query, these Allen statements are used to generate the filter criteria in the WHERE clause.

3.2.2. Processing symbolic time intervals

For Allen statements about symbolic time intervals, the inequations that define the chronological sequence of the points in time at the start or at the end of time intervals can be used to directly build the WHERE clause, which then filters the patients via the valid times [104] in the patient data according to the temporal pattern. For example, for the Allen statement "Pregnancy *contains* Diabetes" from Fig. 2, the chronological sequence is defined as $X_{start} < Y_{start} < Y_{end} < X_{end}$. Assuming

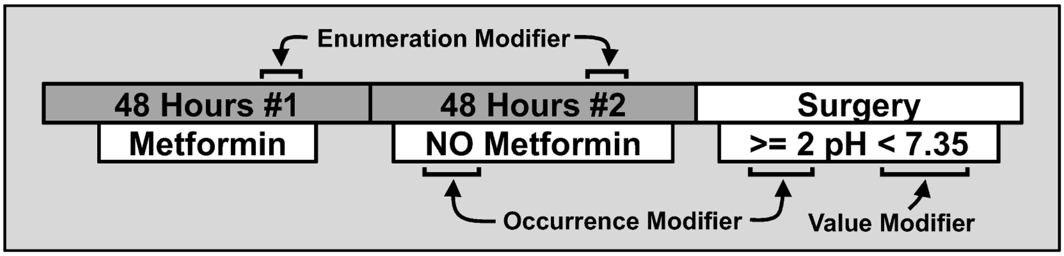
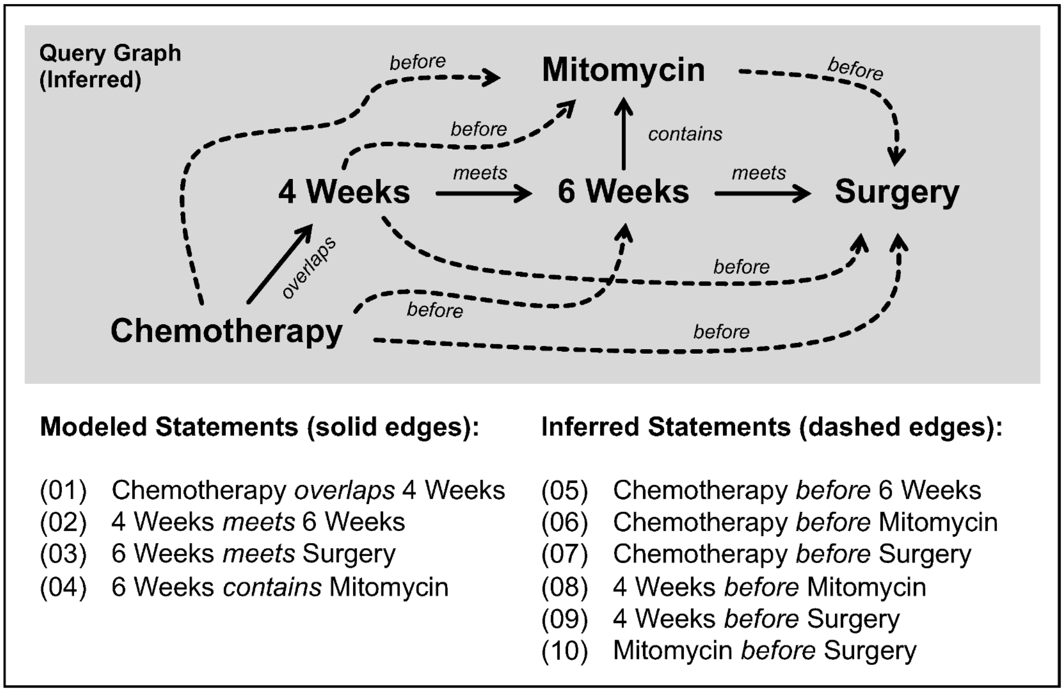


Fig. 4. Time interval modifiers. A temporal pattern with enumeration, occurrence and value modifiers.



Modeled Statements (solid edges):

(01) Chemotherapy *overlaps* 4 Weeks
(02) 4 Weeks *meets* 6 Weeks
(03) 6 Weeks *meets* Surgery
(04) 6 Weeks *contains* Mitomycin

Inferred Statements (dashed edges):

(05) Chemotherapy *before* 6 Weeks
(06) Chemotherapy *before* Mitomycin
(07) Chemotherapy *before* Surgery
(08) 4 Weeks *before* Mitomycin
(09) 4 Weeks *before* Surgery
(10) Mitomycin *before* Surgery

Fig. 5. The query graph of the fourth query from Fig. 3. Inferred statements from Allen's constraint propagation are shown with dashed edges.

that "X = Pregnancy" and "Y = Diabetes", these constraints can be directly translated into an SQL WHERE clause:

```
SELECT Patient …
  WHERE Pregnancy_start < Diabetes_start
    AND Diabetes_start < Diabetes_end
    AND Diabetes_end < Pregnancy_end
```

3.2.3. Processing duration intervals

Duration intervals cannot be processed in the same way as symbolic time intervals. Considering the fourth example from Fig. 3 and following the above approach, one might assume that it is necessary to know the start and end times of the 4 and 6 weeks duration intervals in order to identify the patients. However, duration intervals do not even exist in the database; they only exist as part of the query pattern. Consequently, start and end times of duration intervals cannot be queried in the patient data. For example, a query that includes an SQL WHERE clause such as "`WHERE Chemotherapy_start < 4_Weeks_start`" will inevitably fail, just like any other query that refers to duration intervals. The solution is to eliminate these references via arithmetic operations, which determine the relative temporal distances between all start and end times of the symbolic time intervals. This is the amount of time by which the times of the "outer" symbolic time intervals are pushed apart or held together by the "inner" duration intervals.

In a first step, only these Allen statements that involve duration intervals are collected from the inferred query graph to generate a system of inequations. This system fully describes the chronological sequence of all start and end times. For the example from Fig. 5, such a system is shown in Table 1 in the right column. Additionally, the actual durations of the duration intervals have to be defined (last row). To enable the arithmetic operations and to allow for querying the relative temporal distances in the database, we converted all durations into seconds. Similarly, we assume that the time stamps of the patient data in the database are also represented in seconds relative to a set date (as in Unix time).

Table 1
Expressing relative temporal distances with inequations. The definitions of Allen's relations (see Fig. 1) and the known durations (last row) can be used to define a system of inequations.

Statement No. (Fig. 5)	Allen Statement & Durations	System of Inequations
1	Chemotherapy *overlaps* 4 Weeks	Chemotherapy_start < 4_Weeks_start 4_Weeks_start < Chemotherapy_end Chemotherapy_end < 4_Weeks_end
2	4 Weeks *meets* 6 Weeks	4_Weeks_start < 4_Weeks_end 4_Weeks_end = 6_Weeks_start 6_Weeks_start < 6_Weeks_end
3	6 Weeks *meets* Surgery	6_Weeks_start < 6_Weeks_end 6_Weeks_end = Surgery_start Surgery_start < Surgery_end
4	6 Weeks *contains* Mitomycin	6_Weeks_start < Mitomycin_start Mitomycin_start < Mitomycin_end Mitomycin_end < 6_Weeks_end
5	Chemotherapy *before* 6 Weeks	Chemotherapy_start < Chemotherapy_end Chemotherapy_end < 6_Weeks_start 6_Weeks_start < 6_Weeks_end
8	4 Weeks *before* Mitomycin	4_Weeks_start < 4_Weeks_end 4_Weeks_end < Mitomycin_start Mitomycin_start < Mitomycin_end
9	4 Weeks *before* Surgery	4_Weeks_start < 4_Weeks_end 4_Weeks_end < Surgery_start Surgery_start < Surgery_end
Durations	4 Weeks = 2419200 Seconds	4_Weeks_end − 4_Weeks_start = 2419200
	6 Weeks = 3628800 Seconds	6_Weeks_end − 6_Weeks_start = 3628800

In a second step, solving this system of inequations for the start and end times of the duration intervals allows for deriving inequations, which model the relative temporal distances between the start and end times of all symbolic time intervals. When solving the system from Table 1 for the four variables 4_Weeks_start, 4_Weeks_end, 6_Weeks_start and 6_Weeks_end, the following result can be obtained:

(1) 4_Weeks_end = Surgery_start − 3628800
(2) 4_Weeks_start = Surgery_start − 6048000
(3) 6_Weeks_end = Surgery_start
(4) 6_Weeks_start = Surgery_start − 3628800
(5) Surgery_start - Chemotherapy_end > 3628800

(6) Surgery_start - Chemotherapy_end < 6048000
(7) Surgery_start - Chemotherapy_start > 6048000
(8) Surgery_start - Mitomycin_start > 3628800
(9) Surgery_start - Mitomycin_end > 0

The first four equations (1–4) describe the four variables according to which the system was solved. They are not further relevant in the following. However, the remaining five inequations (5–9) refer to the relative temporal distances between “Chemotherapy”, “Surgery” and “Mitomycin”; this is the time by which the start and end times of these outer symbolic time intervals are pushed apart or held together by the inner duration intervals. Note that the original 4-week and 6-week duration intervals no longer appear in the form of variables. As such, the inequations 5–9 can now be translated directly into a database query:

```
SELECT Patient …
  WHERE Surgery_start - Chemotherapy_end > 3628800
    AND Surgery_start - Chemotherapy_end < 6048000
    AND Surgery_start - Chemotherapy_start > 6048000
    AND Surgery_start - Mitomycin_start > 3628800
    AND Surgery_start - Mitomycin_end > 0
```

To complete the query for the example from Fig. 5, one has to also process the constraints for the statements 6, 7 and 10 of Fig. 5 as described in Section 3.2.2, and include the derived SQL code into the above query.

3.2.4. Processing time interval modifiers

As described in Section 3.1.4, we use time interval modifiers to specify the properties of a symbolic time interval. Time interval modifiers can be used to constrain a medical concept by its numeric value, describe exclusion, count occurrences, or distinguish instances via enumeration.

Integrating support for value modifiers is straightforward. In addition to the temporal constraints in the WHERE clause of an SQL query, a numeric restriction on the value has to be created. For the example in Fig. 4, this could be as simple as adding “`AND pH_value < 7.35`” to the WHERE clause of the SQL query.

Translating occurrence modifiers into database queries is more complicated. “NO” indicates that events or time intervals must not have occurred in the patient’s history. A proper way of implementing this is to first ignore all excluded time intervals (those with “NO” modifiers) in an initial search and then, by using *n* subsequent searches (with *n* equal to the number of excluded time intervals), to subtract patients that *do* have one of the excluded events in their data. For the example in Fig. 4 this would mean that the “NO Metformin” time interval is not considered in the query graph. This initial query may however contain patients who received metformin in the second 48-hour duration interval. To exclude them, a second query must be generated that searches for specifically these patients. The patients identified in this second query are then removed from those of the first query (Fig. 6).

If occurrence modifiers that describe a numerical comparison are used, the generation of more complex SQL statements becomes necessary. For the example in Fig. 4, the software would have to create an SQL subquery that searches for at least two pH measurements having taken place during surgery.

Support for enumeration modifiers occurs automatically when the software routine, which derives the inferred query graph via Allen’s constraint propagation technique, relies only on the names of the intervals. With enumeration modifiers added, duration or symbolic intervals are automatically distinguished. Conversely, it is only necessary to use enumeration modifiers if two intervals that have exactly the same name should be distinguished. For this reason, the two metformin intervals in Fig. 4 were not equipped with the modifiers.

4. Results

4.1. Prototype implementation

To test our approach in practice, we implemented a software prototype consisting of a graphical frontend for modeling the temporal patterns and a backend for query execution. Therefore the implementation of the frontend should replicate the methods described in Section 3.1, the backend the techniques described in Section 3.2.

4.1.1. Frontend application (AllenGUI)

The first program implemented is *AllenGUI*, a graphical temporal query modeler (see Fig. 7). The tool provides a drawing canvas on which temporal relationships can be composed graphically. Using a computer mouse, the user can draw, resize, move and rename horizontal bars depicting intervals. AllenGUI renders bars associated with symbolic time intervals, duration intervals, and interval modifiers using different colors. Whenever the user changes the temporal pattern, AllenGUI creates an inferred query graph (as shown in the “Allen Statements” window on the right), which is stored in a file for the later processing by the backend.

The complex temporal query in Fig. 7 searches for patients who have a history of statins during six months before surgery and received propofol and amiodarone approximately at the time of surgery. This temporal uncertainty is achieved with the one-day duration interval between propofol, amiodarone, and surgery. Note that the query pattern does not specify whether propofol was administered prior to amiodarone because these two symbolic time intervals do not touch each other. The pattern models the statements “Surgery *during* 1 Day”, “Propofol *during* 1 Day”, and “Amiodarone *during* 1 Day”. The lower part of the pattern references the “Surgery” interval from above. It specifies the medical history after the surgery, where it aims to detect rising creatinine kinase values (CK) within

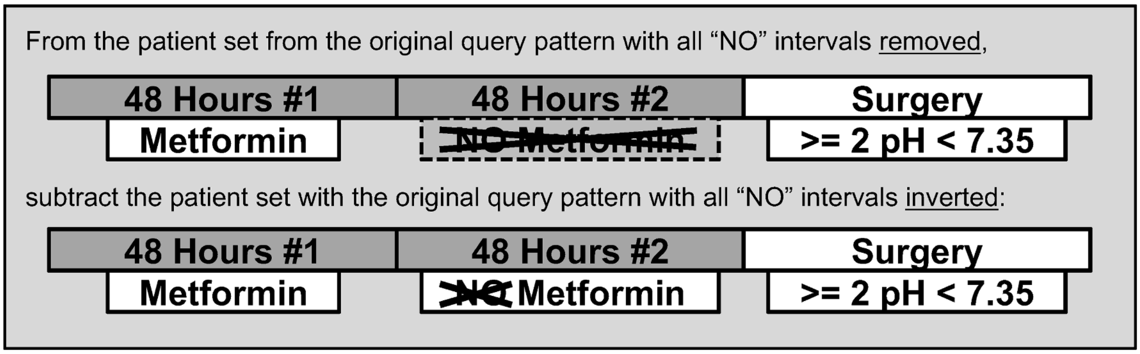
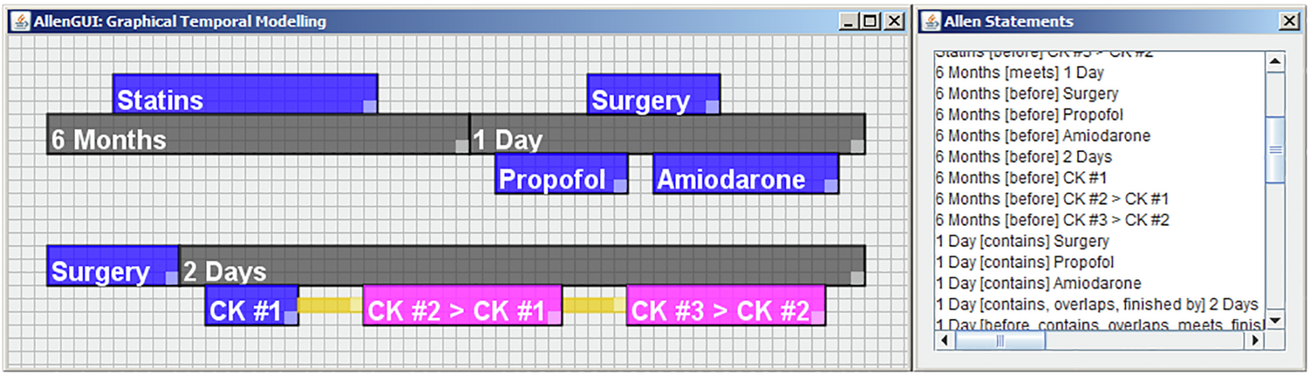


Fig. 6. Exemplary translation of the “NO” occurrence modifier. This fig. explains how to exclude the patients that received metformin in the second 48-h duration interval from Fig. 4.



Statins [before] CK #3 > CK #2
6 Months [meets] 1 Day
6 Months [before] Surgery
6 Months [before] Propofol
6 Months [before] Amiodarone
6 Months [before] 2 Days
6 Months [before] CK #1
6 Months [before] CK #2 > CK #1
6 Months [before] CK #3 > CK #2
1 Day [contains] Surgery
1 Day [contains] Propofol
1 Day [contains] Amiodarone
1 Day [contains, overlaps, finished by] 2 Days
1 Day [before, contains, overlaps, meets, finis

Fig. 7. User interface of AllenGUI. The AllenGUI tool is used for the graphical modeling of temporal constraints.

two days. This is done via the three CK intervals, which make use of time interval modifiers (#1, #2, #3) and relative value comparisons. Via the orange connector lines, the temporal pattern expresses that CK #1 must be before CK #2, which in turn must be before CK #3. The relative value comparisons in the two CK intervals on the right state that the value of CK #2 has to be higher than that of CK #1, and that the value of CK #3 has to be higher than that of CK #2.

AllenGUI is implemented in Java and uses an AWT and Swing-based user interface. We decided to integrate Allen's constraint propagation technique into the frontend to enable direct user-feedback; for example, the user gets notified when the tool detects an inconsistent temporal pattern. For this we relied on an existing open-source software library [105]. To implement the horizontal connector lines for the *before* and *after* relationships, we utilized a trick: Technically, these are normal (but "flat") symbolic time intervals that have no visual label and which form *meets* relationships with their neighboring intervals. Via Allen's constraint propagation technique, these are inferred automatically into the *before* and *after* relationships as intended by the user.

4.1.2. Backend application (AllenSPARQL)

Furthermore, we implemented a Java-based backend called *AllenSPARQL*. Similar to generating SQL queries from a query graph that can be executed on relational databases (and as outlined in Section 3.2), AllenSPARQL produces SPARQL queries [106] that it executes on a triple store.

We decided to connect AllenSPARQL to the i2b2 system, a widely used clinical research platform with a well-understood database schema, assuming that this facilitates the adoption and replication of our method. A researcher can select an already executed i2b2 query. As a data preparation step, AllenSPARQL then converts the relational i2b2 patient data from this i2b2 query into the RDF format [107] and transfers the data using D2RQ [108] into an Apache Jena Fuseki [109] server. AllenSPARQL also implements a simple unsupervised temporal abstraction method based on the concatenation of adjacent time intervals below a threshold value (described further in [110]), which can be configured and applied during the data preparation. The data are then ready for being filtered via the temporal patterns generated by AllenGUI.

Whenever AllenSPARQL detects a change in the query graph produced by AllenGUI, it translates the temporal pattern into a SPARQL query and executes it on the Fuseki server. To perform the calculations necessary for handling duration intervals (as described in Section 3.2.3), we use the open source mathematical software SageMath [111]. After query execution, AllenSPARQL lists and counts all identified patients. Generated SPARQL queries are also prepared for the retrieval of additional data, such as the start and end times of time intervals, or the values of numeric data elements. After uncommenting the respective variables in the SPARQL code, the query can be manually executed on Fuseki's web interface, which then retrieves and renders the patient data in a tabular format.

We based our decision to use Semantic Web technologies [112] on the fact that these are built on the foundation of graph structures. As such, they are compatible with the query graph described in Section 3.2.1, which simplifies the translation into database queries. However, as we aimed to focus on the graphical notation, a more detailed description of the backend would go beyond the scope of the paper.

4.2. Evaluation

We have carried out an evaluation consisting of two parts. The first part aimed to ensure technical functionality using a minimal, known set of artificial test data. In the second part, we used a larger amount of real patient data to examine whether the implementation was acceptable in terms of query runtimes. More importantly, we used medical queries published in the literature to analyze whether our graphical notation is capable of modeling typical temporal patterns from a conceptual point of view.

4.2.1. Technical i2b2 test dataset

As a first step, we created a minimal technical i2b2 test dataset. It contains 13 fictional patients (one for each Allen relation) with data for the two concepts "A" and "B". The temporal data consists of short time intervals in a predefined pattern to test the data preprocessing (such as the simple temporal abstraction) in AllenSPARQL and the translation of query graphs into SPARQL code. The dataset also contains known numeric values to allow for testing value modifiers. Using the dataset enabled us to analyze whether a temporal query returned the expected results. For example, when querying "A after B", our "last" Allen relation, the system should identify the "last" patient with ID 13. In this way, the functionality could be tested in a systematic way, which was particularly important for the implementation of the prototype. The test dataset is described in more detail in [110].

4.2.2. Queries on the MIMIC-III dataset

To further assess the capability of our approach for modeling and executing typical medical temporal queries on real data, we replicated the temporal queries from Nigrin and Kohane [11] on the *Medical Information Mart for Intensive Care* (MIMIC-III) [12] database, version 1.4. This openly accessible database contains de-identified data from about 40,000 intensive care patients admitted to the Beth Israel Deaconess Medical Center in Boston. It includes ICU data (bedside monitoring, charts), laboratory and microbiology tests, provider order entry, billing-related data (ICD9, DRG, CPT), demographics, as well as free text notes and reports [12].

Our evaluation environment comprised an i2b2 version 1.7 instance with an Oracle database (11 g Enterprise Edition Release 11.2), both installed on servers inside the University Hospital Erlangen. We exported a subset of MIMIC-III data into these systems. AllenGUI and AllenSPARQL were executed on a typical Windows 7 office computer,

Table 2
The temporal queries that we adopted from Nigrin and Kohane [11] and executed on the MIMIC-III dataset. Query execution times are shown as mean and standard deviation across three measurements.

Query	Description	i2b2 Query	Execution on our system
1	Diabetic patients (type 1 diabetes) with at least two occurrences of glycohemoglobin values greater than 10 percent.	"Diabetes Type I" and "HbA1c"	i2b2 patients: 122 Triples: 7482 AllenSPARQL patients: 6 Query time: 99 ± 25 ms
2	Diabetic patients who have had at least one elevated glycohemoglobin value that is not coincident with the time of their diagnosis (one month).	"Diabetes Type I" and "HbA1c"	i2b2 patients: 122 Triples: 7482 AllenSPARQL patients: 2 Query time: 288 ± 88 ms
3	Patients with congenital hypothyroidism who were not seen in the endocrine clinic within one month after an elevated thyrotropin (TSH) laboratory result (i.e., a value greater than or equal to 10 µU/ml).	"Congenital Hypothyroidism" and "TSH" and "ICU" *Note: "endocrinic clinic visit" was replaced with ICU stay.*	i2b2 patients: 11 Triples: 1007 AllenSPARQL patients: 8 Query time: 33 ± 4 ms
4	Patients with Tanner Stage 2 breast characterization uninterrupted by a Tanner Stage 1 characterization.	"SAPS" *Note: "Tanner stages" were replaced with SAPS scores.*	i2b2 patients: 44,476 Triples: 4,481,087 AllenSPARQL patients: 86 Query time: 67.8 ± 1.6 s
5	All patients with three ICU stays with increasing SAPS scores.	"SAPS"	i2b2 patients: 44,476 Triples: 4,481,087 AllenSPARQL patients: 2,606 Query time: 26.9 ± 0.9 min
6	All patients with three ICU stays within 3 months, with increasing SAPS scores.	"SAPS"	i2b2 patients: 44,476 Triples: 4,481,087 AllenSPARQL patients: 1,062 Query time: 4.2 ± 0.05 min
7	Patients with elevated glycohemoglobin value (>12 percent), followed by onset of microalbuminuria (µAlb > 20 mg/g), followed by a glycohemoglobin value greater than 12 percent, but with persistence of microalbuminuria.	"uAlb" and "HbA1c"	i2b2 patients: 988 Triples: 15,523 AllenSPARQL patients: 3 Query time: 208 ± 8 ms
8	Patients with HbA1c greater than 12% that is coincident within one month with microalbuminuria (µAlb > 20 mg/g), followed by an HbA1c value less than 12% that is still coincident with microalbuminuria.	"uAlb" and "HbA1c"	i2b2 patients: 988 Triples: 15,523 AllenSPARQL patients: 0 Query time: 236 ± 40 ms

equipped with a first-generation Intel Core i5 CPU, 8 GB of RAM and a solid state drive. This machine was not performance-optimized and ran various programs in the background, such as PostgreSQL with the MIMIC-III database. Furthermore, D2RQ and Apache Jena Fuseki were installed on this computer to handle the i2b2 data in RDF.

Table 2 lists the temporal queries from [11], which we have remodeled so that they can be executed with our approach. The required data elements for each query were first queried in the i2b2 system using the MIMIC-III data, as shown in column "i2b2 Query". The resulting i2b2 patient set was then post-processed with AllenGUI and AllenSPARQL using the respective temporal pattern from Fig. 8. The final patient counts returned by AllenSPARQL are denoted as "AllenSPARQL patients". Where it seemed appropriate, we also carried out additional queries (5, 6 and 8). It is important to understand that not all data elements from these initial i2b2 queries had to be part of the temporal pattern. For example, in query 1 the i2b2 patient set contained data only from diabetics. As the diabetes concept has no further temporal relationship with the HbA1c concept, it was not necessary to reuse this concept in the temporal pattern.

We were able to successfully reproduce all temporal patterns by Nigrin and Kohane with our approach. During the modeling process, however, we noticed some aspects that we would like to elaborate on in the following.

For query pattern 1 it is worth noting that this query can be modeled directly in i2b2 (without temporal logic) when constraining the HbA1c to be greater than 10 and by setting the occurrence of the HbA1c concept to greater than 1. Such an i2b2 query also returns the same six patients as in our approach.

As described by Nigrin and Kohane, query 1 can be further refined from a medical perspective by excluding all patients who had an HbA1c measurement coincident with their diabetes diagnose, e.g., within 1 month. We implemented this query as shown in pattern 2 by temporally linking the (excluded) HbA1c to the diagnosis via the one-month duration interval. It is interesting to note that when inferring across this temporal pattern, the left HbA1c and the diabetes intervals are linked via the generic Allen relation. This is a composite Allen relation comprising all thirteen Allen relations:

NO HbA1c #1 > 10 {*before, after, during, contains, overlaps, overlapped by, meets, met by, starts, started by, finishes, finished by, equals*} Diabetes Type I

In the context of Allen's formalism, this means that no conclusions can be drawn regarding the direct relationship between the two symbolic time intervals (as all thirteen Allen relations between the intervals are possible). However, our support for duration intervals allows us to deduce additional timing constraints between both intervals. By resolving a system of inequations as described in Section 3.2.3, further constraints are derived, which then ensure that the two symbolic time intervals occur within one month.

Since the MIMIC-III database does not contain all data elements used by Nigrin and Kohane, we had to substitute some of them, while

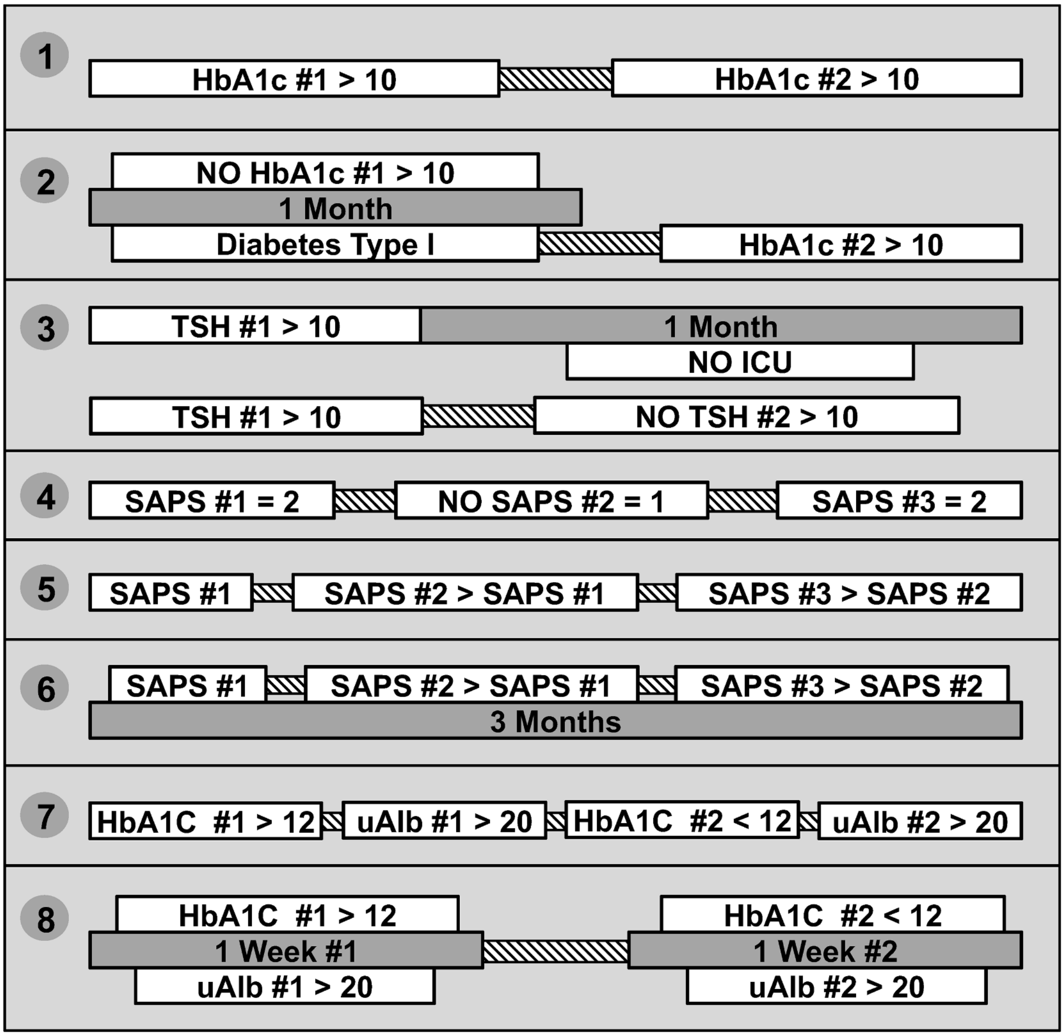


Fig. 8. The graphical temporal patterns used to model the queries in Table 2.

keeping the original temporal patterns. This was the case in query 3, where we replaced the endocrine clinic visit with an ICU stay. Similarly, because the MIMIC-III database does not include Tanner stages, we replaced this concept with SAPS scores in query 4. As such a query does not make sense from a medical perspective, we modified the pattern in query 5 to search for patients with increasing SAPS scores for each ICU stay. Running such a query on a live reporting system could be useful to predict complications for patients with clinical deteriorations.

Query 7 is another example of such a sequential search pattern. One problem is that the before and after Allen relations do not model true sequences, but rely on the transitive properties of these relations. For query 7, this means that there could be any number of microalbumin measurements in the patient data between the two microalbumin measurements described in the query pattern. Query pattern 8 attempts to model the research question in a different way by linking the HbA1c to microalbumin via a one-week duration interval (similar to query 2).

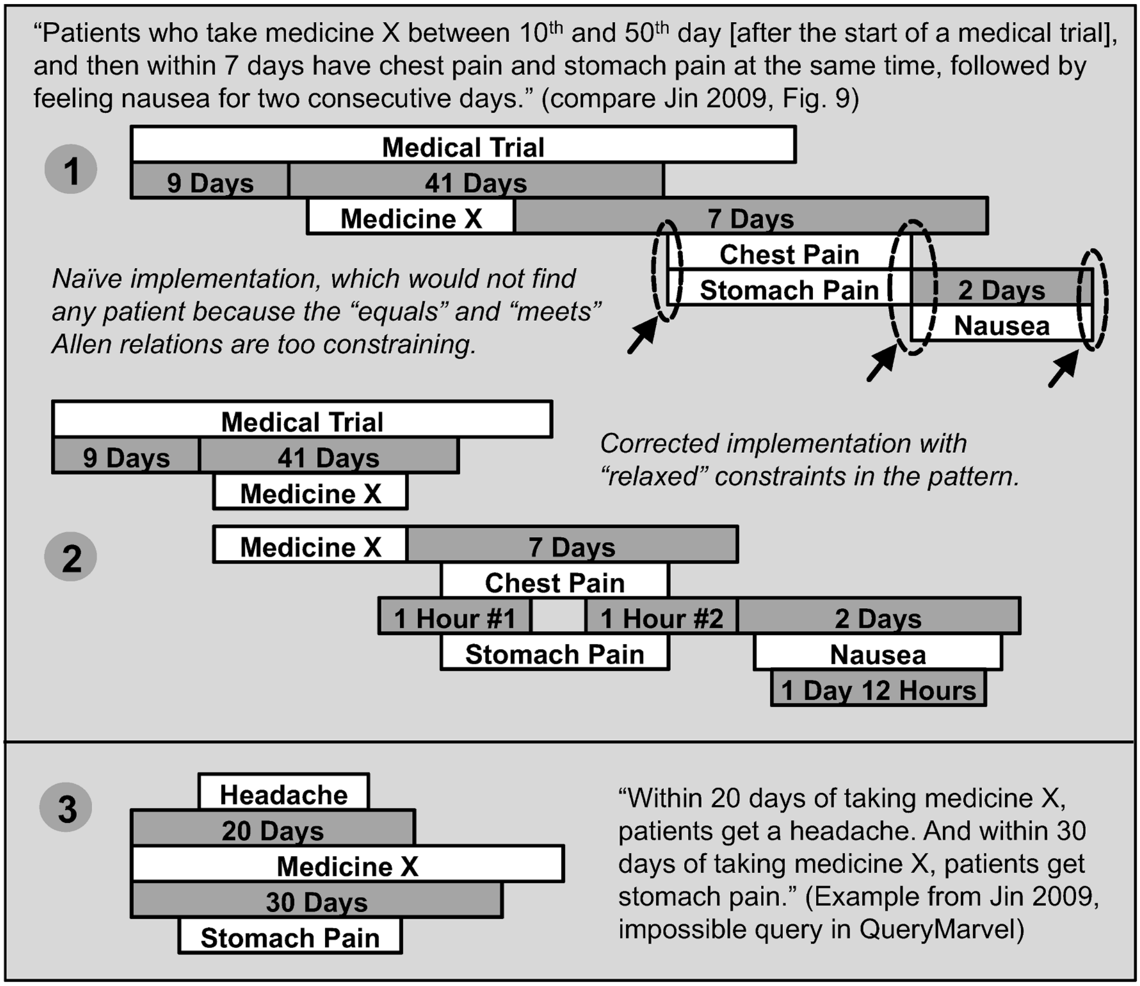
As outlined in Table 2, the query execution times varied depending on the RDF dataset size and query pattern. In most cases, and whenever we made changes to the query pattern in AllenGUI, the system responded to new queries without significant delay. Query performance was poor only in query 5. This is because the SPARQL engine has to identify all combinations of subsequent SAPS scores while comparing their values. Processing such “sequences” is a computationally intensive task. The reason for this is the huge search space, a consequence of the sequential order of clinical facts on the timeline, in combination with the unspecific use of the Allen relations before and after in the query pattern. However, the execution time of the query can be considerably reduced when introducing additional temporal constraints, as we show in query 6. If one demands that the three SAPS values occur within three months, the search space and the execution time are reduced considerably, with the latter to about 4 min.

Finally, to analyze whether our implementation returned correct results, we reviewed the generated SPARQL code for soundness, but also checked the returned data records of the executed queries by modifying the SPARQL code to also retrieve patient data (as described in Section 4.1.2).

5. Discussion and outlook

In 2015, Xu et al. [74] discovered that almost all cohort identification systems use Boolean logic to couple inclusion and exclusion criteria. That is, Boolean logic is the set standard for this task. At the same time, the authors observed that the integration level of temporal query functionality is “not very high”. They assume that this is due to the complexity of dynamic temporal phenotyping algorithms or due to a lack of tools for modeling them. However, as there is a vast amount of research dealing with time-related retrieval, exploration and visualization of data (see e.g. the review by Rind et al. in [58]), the question arises why the methods described therein are not yet widely used in current translational research systems.

As described by Combi and Oliboni in [64], it is not always simple to distinguish between visualization and visual exploration on the one



"Patients who take medicine X between 10th and 50th day [after the start of a medical trial], and then within 7 days have chest pain and stomach pain at the same time, followed by feeling nausea for two consecutive days." (compare Jin 2009, Fig. 9)

Naïve implementation, which would not find any patient because the "equals" and "meets" Allen relations are too constraining.

Corrected implementation with "relaxed" constraints in the pattern.

"Within 20 days of taking medicine X, patients get a headache. And within 30 days of taking medicine X, patients get stomach pain." (Example from Jin 2009, impossible query in QueryMarvel)

Fig. 9. Comparison with QueryMarvel. The fig. shows how one would model queries from [57] with our graphical notation. While the upper example is complex to model with our approach, the lower one (pattern 3) is impossible with QueryMarvel.

hand and query of temporal clinical information on the other hand, since both are intertwined. As such, we assume that many of the systems described by Rind in [58] are specialized standalone research applications, and their functionality for temporal queries may not always be transferable to other implementations.

5.1. Comparison with other approaches

As described in the background section, most approaches to model temporal queries are either text-based or form-based. There are only a few approaches which we consider truly graphical, such as the *QueryMavel* temporal language [57] or the approach by Chittaro, Combi and Oliboni [63,64]. Due to the similarities in modeling, we briefly compare these two approaches with ours.

QueryMavel uses a comic strip metaphor to model temporal constraints. Certain temporal patterns can be modeled more intuitively in QueryMarvel than with our method, as demonstrated with the patterns 1 and 2 in Fig. 9. A good understanding of Allen's formalism is required to successfully construct queries in our notation. As such, the naïve adoption of one of the temporal QueryMarvel patterns from [57], shown in pattern 1, would not return the desired results. This is due to the very restrictive nature of the *equals* and *meets* relations (circles), which assume a precision of timing in patient data that is unlikely to match clinical reality. This effect can be mitigated via the more complex pattern 2 or the other methods discussed below. In addition, the unintentionally modeled statement "41 Days *overlaps* 7 Days" in pattern 1 was incorrect; this was solved in pattern 2 via the double referencing of "Medicine X". From this point of view, QueryMarvel is easier to use, but on the other hand it does not support parallel timelines [57] or patterns that can be created easily with our method (see e.g. pattern 3).

The approach by Chittaro, Combi and Oliboni [63,64] uses a paint strip metaphor to graphically model time intervals on a drawing surface. These can be (technically) squeezed and stretched to enable the definition of composite Allen relations that are *conceptual neighborhoods* [113]. This approach enables expressing temporal indeterminacy. In our notation, one would have to resort to disjunctions, as described in [63] and as shown in pattern 1a in Fig. 10. However, via a workaround it is possible to model such disjunctions by relying on Allen's constraint propagation technique: As shown in pattern 1b, using an intermediate duration interval allows for inferring the composite temporal relation "A {*starts, started by, equals*} B". The drawback is that this modifies the original query by specifying that A and B must have a duration of at least one second. Patterns 2 and 3 in Fig. 10 show adaptations of similar queries from [64] using our approach. We will discuss possible alternatives for modeling indeterminacy below.

The original approach as described in [63] was updated in [64], where the authors, besides support for relative temporal distances and other features, introduced support for queries involving different temporal granularities [114]. This is a feature not considered in our approach. For example, it is not possible to model that symbolic intervals must start in the same calendar year (pattern 2 in Fig. 10). Instead, we define relative temporal relationships via duration intervals, which, in our opinion, seems to be more adapted to medical research, as health problems are not tied to calendar units. However, such support for

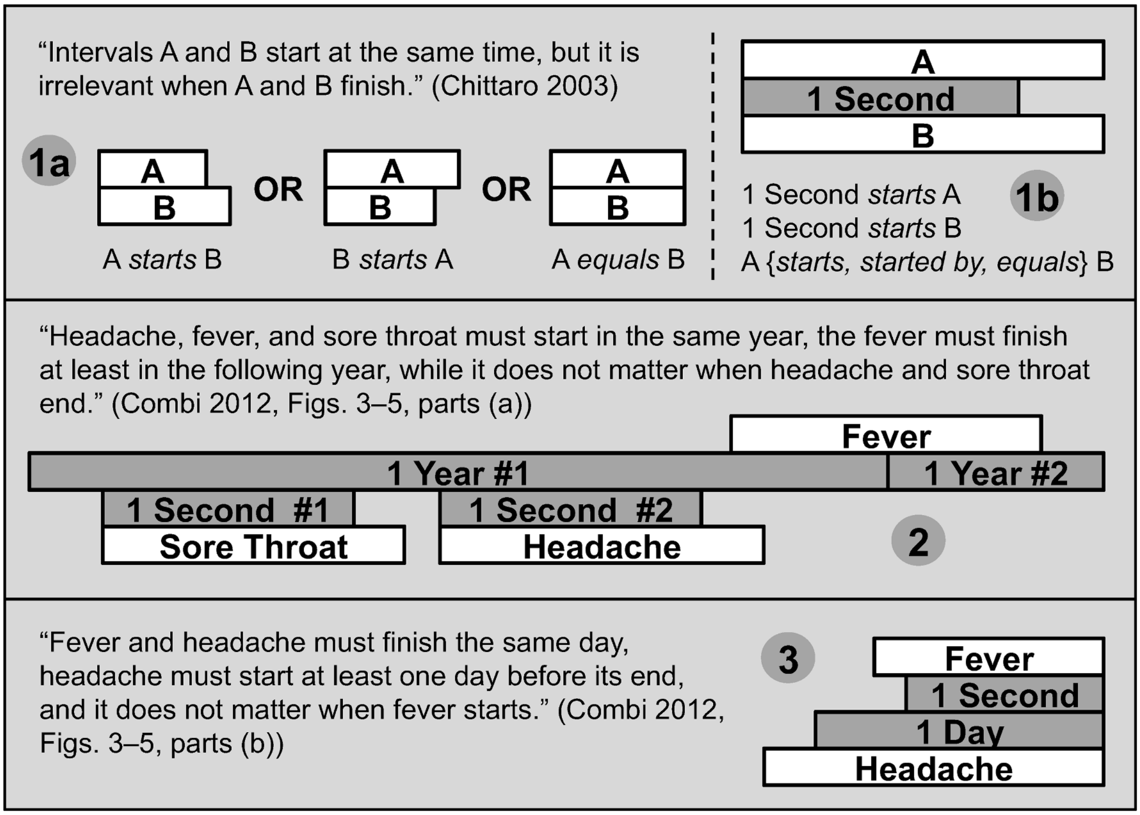


Fig. 10. Comparison with the approach from Chittaro, Combi and Oliboni. The fig. shows how one would model queries from [63] and [64] with our graphical notation.

different temporal granularities could be of use for administrative aspects, such as quality control or financial analyses.

QueryMarvel and the approach by Chittaro, Combi and Oliboni have in common that they require rather complex user interface implementations, because of the metaphors they use. In contrast, our approach only uses horizontal bars. Furthermore, for our Allen-based notation we did not have to develop a new algorithm to check the query pattern for basic logical consistency and could rely on an existing implementation [105] of the original algorithm proposed by Allen in [10].

5.2. Is Allen's time interval algebra too expressive?

Other researchers (e.g. [115–117]) have discussed that the high level of Allen's temporal expressiveness can be a problem in practice, for which the phenomena described in Fig. 9 and Fig. 10 are good examples.

In pattern 1 of Fig. 9, the *equals* relations between chest pain and stomach pain would only identify patients where the start and end times of the two time intervals are the same. This issue can be partially mitigated via more complex patterns, as the one in pattern 2. A second solution, which we also implemented in AllenSPARQL, is the relaxation of the temporal constrains of Allen's relations via a configurable epsilon value. Added to the definition of chronological sequences (as in Fig. 1), this epsilon value can be used to capture temporal uncertainties in the query pattern or in noisy patient data. This idea is further described e.g. in [17,118,119]. A third solution (for this specific example) would be to round all time stamps in the patient data to days. The query would then operate on the temporal granularity of days.

Similarly, the issues discussed in the context of Fig. 10 are other effects stemming from the high specificity of Allen's temporal relations. A possible solution might be the adoption of the "broken paint strip" metaphor from [64] for modeling open symbolic time intervals, as sketched in Fig. 11. When translating this updated graphical notation into a query graph, we would have to create composite Allen relations whenever we encounter such a broken interval end. For pattern 3 of Fig. 11, the Allen relation between "Fever" and "1 Day" would be "Fever {*finished by, equals, finishes*} 1 Day". Another option worth investigating is using Freksa's semi-intervals [113] to model the open intervals.

To reduce the expressiveness of Allen's time interval algebra, one could try to combine the detailed Allen relations (*meets*, *during*, *starts*, *finishes*, and their inverses) into coarser ones, as Allen has already described in his original work [10]. For example, the relations *before* and *meets* could be merged into *bef* and *during*, *starts* and *finishes* into *dur*. Nevertheless, complex Allen relations, such as *meets*, *overlaps*, *during*, *starts* and *finishes*, are essential for supporting relative temporal distances, as demonstrated throughout our work. We would therefore discourage the use of a reduced Allen formalism, as it would significantly limit temporal expressiveness.

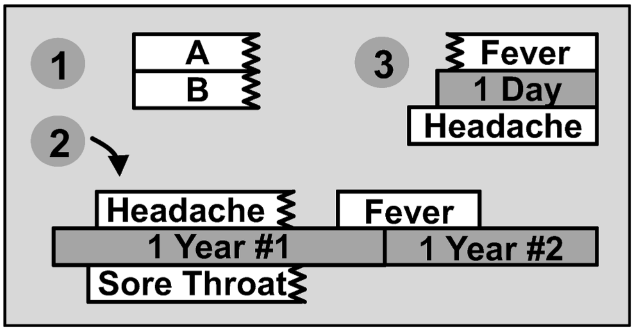


Fig. 11. A possible implementation of "open symbolic time intervals". A (graphical only) adoption of the "broken paint strip" metaphor from [64] could be useful for modeling temporal indeterminacy. The patterns shown correspond to those in Fig. 10.

5.3. Limitations and future improvements

The practical use of our approach presupposes a good understanding of Allen's time interval algebra. The graphical representation of Allen relations might be misinterpreted by some users, which may cause errors to creep into queries. Our initial observations from pilot usage indicate that the logical precedence of time interval modifiers may not always be comprehensible to users, especially when using a combination of occurrence and value modifiers (as illustrated in Section 3.1.4 with the pH interval). A possible solution could be to replace the currently used standard relational operators with those from the Arden Syntax, such as "AT MOST", "AT LEAST", etc. [120]. This would also lead to further consistency with the "NO" occurrence modifier.

When reinterpreting the search patterns from Nigrin and Kohane and implementing the AllenGUI queries, we observed that search patterns could sometimes be modeled in different ways (compare e.g. queries 7 and 8 in Table 2). This is caused on the one hand by linguistic inaccuracies of the research question, and on the other hand by the high expressiveness of Allen's time interval algebra. Such different queries usually also return different results. Nevertheless, it is not always possible to tell which temporal pattern best answers a medical question. These uncertainties remind of those described in Hripcsak et al. [121], albeit less pronounced in our case. Temporal queries, as presented in our work, should only be used for preselecting patients. Subsequently, close collaboration with physicians and an additional accurate assessment of the results (possibly with direct access to the original patient records) is required.

Three further constructs would be valuable in our approach and are planned to be integrated in future revisions. First, it is not yet possible to directly express the Boolean operator "OR", both for the logical and the temporal part of a query. A solution for the first could be to allow for expressing disjunctions in single symbolic time intervals, as in "Propofol OR Amiodarone". A practical solution for temporal disjunctions could be to adopt the broken paint strip metaphor from [64], as discussed above. Second, while our approach has some support for modeling trends of numerical values over time, as shown in the lower part of Fig. 7 for the CK values, or as in the temporal pattern 5 of Fig. 8, it does not allow for modeling true ordered sequences. This feature, which has been implemented e.g. in Eureka! [122], VizPattern [62], and many other visual data exploration tools, could be potentially integrated into our approach by extending the 13 Allen relations with two new relations, *precedes* and *succeeds*. Third, there is currently no support for absolute time intervals. Such a feature could be implemented via intervals that encode absolute start and end times, as in "2017-01-01 – 2017-03-19".

Finally, we will reconsider the use of our RDF-based backend, although it has worked well as a proof-of-concept. However, in [123] and [124] we recently demonstrated that query formalisms from one system can be converted into another one, and demonstrated this between i2b2 and Samply [125]. We aim to continue with this research by investigating to what extent temporal constraints, created with AllenGUI, can be translated and injected into other query formalisms (such as an i2b2 XML query definition [126]). We assume that it will not always be possible to reproduce the full expressiveness of our notation in these systems, but we expect that many typical temporal patterns can be translated. Processing temporal query patterns where this does not work will be an interesting aspect of this future research. Another option would be the translation of the temporal patterns into executable code, for example based on the Arden Syntax [127,128], or implemented in a programming language that natively provides support for Allen's relations, such as the *Clinical Quality Language* [129,130].

6. Conclusions

The efficient processing of temporal information is of utmost importance in medicine. We have presented a straightforward but expressive approach to graphically model queries with complex temporal constraints. Via the extensions, *duration intervals* and *time interval modifiers*, we were able to overcome many challenges associated with EHR-based temporal phenotyping. We have shown that our graphical notation is suitable for modeling typical medical temporal queries. Other approaches may be more powerful in certain aspects, but are more complex to implement and may not always be generic enough to be transferable to other environments. The results of our work suggest that similar temporal expressiveness can be provided with a minimalistic graphical notation. We plan to refine our graphical approach in the near future and aim to conduct a detailed usability assessment.

Ethics approval and consent to participate

The experiments described in this work were performed using anonymized patient data from the freely available MIMIC-III database. For the entire rest of this work, only artificial data were used. The authors therefore declare that this study is in accordance with the Helsinki Declaration.

Consent for publication

Not applicable.

Funding

This work was funded in part by the German Federal Ministry of Education and Research (BMBF) within the Medical Informatics Initiative (MIRACUM Consortium) under the funding number 01ZZ1801A.

Data availability statement

The source code (licensed under the GNU General Public License v3.0) is available on GitHub (https://github.com/sebmate/AllenGUI and https://github.com/sebmate/AllenSPARQL). Also supplied are the development i2b2 dataset (with the 13 artificial patients) and the SQL scripts to populate the i2b2 system with the MIMIC-III data. Please note that the software is of prototypical character and provided "as is", without any warranties.

Declaration of Competing Interest

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

Acknowledgements

We thank Jörn Franke for developing and providing the "alleninintervalrelationships" package (used in AllenGUI). We also thank the Mathematics Department of the Autonomous University of Madrid, SageMath, Inc., and Andrey Novoseltsev for providing the SageMathCell services running at https://sagecell.sagemath.org (used by AllenSPARQL). We would also like to express our sincere thanks to all researchers involved in compiling and providing the MIMIC-III database.

Authors' contributions

SM developed the approach, implemented the software and performed the evaluation. He wrote the manuscript with feedback from all other authors. In particular, SK, TB, LAK and MOK contributed to the clarity and understandability of the manuscript. LAK, DT, TB and IC contributed medical expertise for the examples in this work. MS and HUP provided and managed the infrastructure for this research.

Authors' information

The present work was performed in (partial) fulfillment of the requirements for obtaining the degree "Dr. rer. biol. hum." from the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) (SM).

References

[1] S. Rea, J. Pathak, G. Savova, T.A. Oniki, L. Westberg, C.E. Beebe, et al., Building a robust, scalable and standards-driven infrastructure for secondary use of EHR data: the SHARPn project, J. Biomed. Inform. 45 (4) (2012) 763–771.

[2] H. Mo, W.K. Thompson, L.V. Rasmussen, J.A. Pacheco, G. Jiang, R. Kiefer, et al., Desiderata for computable representations of electronic health records-driven phenotype algorithms, J. Am. Med. Inform. Assoc. 22 (6) (2015) 1220–1230.

[3] C.L. Overby, C. Weng, K. Haerian, A. Perotte, C. Friedman, G. Hripcsak, Evaluation considerations for EHR-based phenotyping algorithms: a case study for drug-induced liver injury, AMIA Summits Transl. Sci. Proc. 2013 (2013) 130–134.

[4] P.L. Peissig, L.V. Rasmussen, R.L. Berg, J.G. Linneman, C.A. McCarty, C. Waudby, et al., Importance of multi-modal approaches to effectively identify cataract cases from electronic health records, J. Am. Med. Inform. Assoc. 19 (2) (2012) 225–234.

[5] M.R. Boland, G. Hripcsak, Y. Shen, W.K. Chung, C. Weng, Defining a comprehensive verotype using electronic health records for personalized medicine, J. Am. Med. Inform. Assoc. 20 (e2) (2013 Dec) e232–e238.

[6] J.C. Denny, Chapter 13: Mining Electronic Health Records in the Genomics Era, in: F. Lewitter, M. Kann, (Eds.) PLoS Comput Biol. 8(12) (2012) e1002823.

[7] J. Ross, S. Tu, S. Carini, I. Sim, Analysis of eligibility criteria complexity in clinical trials, AMIA Summits Transl. Sci. Proc. 1 (2010) (2010) 46–50.

[8] M. Conway, R.L. Berg, D. Carrell, J.C. Denny, A.N. Kho, I.J. Kullo, et al., Analyzing the heterogeneity and complexity of electronic health record oriented phenotyping algorithms, AMIA Annu. Symp. Proc. 2011 (2011) 274–283.

[9] S.N. Murphy, G.M. Weber, M.E. Mendis, V.S. Gainer, H.C. Chueh, S.E. Churchill, et al., Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2), J. Am. Med. Inform. Assoc. 17 (2) (2010) 124–130.

[10] J.F. Allen, Maintaining knowledge about temporal intervals, Commun. ACM. ACM 26 (11) (1983) 832–843.

[11] D.J. Nigrin, I.S. Kohane, Temporal expressiveness in querying a time-stamp-based clinical database, J. Am. Med. Inform. Assoc. 7 (2) (2000) 152–163.

[12] A.E.W. Johnson, T.J. Pollard, L. Shen, L.-W.H. Lehman, M. Feng, M. Ghassemi, et al., MIMIC-III, a Freely accessible critical care database, Sci. Data 3 (2016).

[13] Y. Touitou, E. Haus, Biologic Rhythms in Clinical and Laboratory Medicine, Springer Science & Business Media, 2012, p. 1.

[14] D. Chen, B. Moulin, J. Wu, Analyzing and Modeling Spatial and Temporal Dynamics of Infectious Diseases, John Wiley & Sons, 2014, p. 1.

[15] S. Woolf, H.J. Schünemann, M.P. Eccles, J.M. Grimshaw, P. Shekelle, Developing clinical practice guidelines: types of evidence and outcomes; values and economics, synthesis, grading, and presentation and deriving recommendations, Implement. Sci. 4 (7) (2012) 61.

[16] R. Kummer von, Brain hemorrhage after thrombolysis: good or bad? Stroke 33 (6) (2002) 1446–1447.

[17] R. Moskovitch, Y. Shahar, Fast time intervals mining using the transitivity of temporal relations, Knowl. Inf. Syst. 42 (1) (2013) 21–48.

[18] C. Weng, M.G. Kahn, J.H. Gennari, Temporal knowledge representation for scheduling tasks in clinical trial protocols, Proc. AMIA Symp. 879–83 (2002).

[19] L. Zhou, G.B. Melton, S. Parsons, G. Hripcsak, A temporal constraint structure for extracting temporal information from clinical narrative, J. Biomed. Inform. 39 (4) (2006) 424–439.

[20] C. Tao, W.-Q. Wei, H.R. Solbrig, G. Savova, C.G. Chute, CNTRO: a semantic web ontology for temporal relation inferencing in clinical narratives, AMIA Annu. Symp. Proc. 13 (2010) (2010) 787–791.

[21] S. Velupillai, D.L. Mowery, S. Abdelrahman, L. Christensen, W.W. Chapman, Towards a generalizable time expression model for temporal reasoning in clinical notes, AMIA Annu. Symp. Proc. 2015 (2015) 1252–1259.

[22] M.R. Boland, S.W. Tu, S. Carini, I. Sim, C. Weng, EliXR-TIME: a temporal knowledge representation for clinical research eligibility criteria, AMIA Summits Transl. Sci. Proc. 2012 (2012) 71–80.

[23] W.F. Styler, S. Bethard, S. Finan, M. Palmer, S. Pradhan, P.C. de Groen, et al., Temporal annotation in the clinical domain, Trans. Assoc. Comput. Linguist. 2 (2014) 143–154.

[24] L. Zhou, C. Friedman, S. Parsons, G. Hripcsak, System architecture for temporal information extraction, representation and reasoning in clinical narrative reports, AMIA Annu. Symp. Proc. 869–73 (2005).

[25] A. Shknevsky, Y. Shahar, R. Moskovitch, Consistent discovery of frequent interval-based temporal patterns in chronic patients' data, J. Biomed. Inform. 75 (2017) 83–95.

[26] W. Sun, A. Rumshisky, O. Uzuner, Evaluating temporal relations in clinical text: 2012 i2b2 challenge, J. Am. Med. Inform. Assoc. 20 (5) (2013) 806–813.

[27] L. Zhou, S. Parsons, G. Hripcsak, The evaluation of a temporal reasoning system in processing clinical discharge summaries, J. Am. Med. Inform. Assoc. 15 (1) (2008) 99–106.

[28] R. Moskovitch, C. Walsh, F. Wang, G. Hripcsak, N. Tatonetti, Outcomes prediction via time intervals related patterns, 2015 IEEE International Conference on Data Mining, IEEE, 2015, pp. 919–924.

[29] R. Moskovitch, H. Choi, G. Hripcsak, N. Tatonetti, Prognosis of clinical outcomes with temporal patterns and experiences with one class feature selection, IEEE/ACM Trans. Comput. Biol. Bioinform. 14 (3) (2017) 555–563.

[30] R. Moskovitch, Y. Shahar, Classification of multivariate time series via temporal abstraction and time intervals mining, Knowl. Inf. Syst. (2015).

[31] J. Huang, J. Huan, A. Tropsha, J. Dang, H. Zhang, M. Xiong, Semantics-driven frequent data pattern mining on electronic health records for effective adverse drug event monitoring, BIBM (2013).

[32] Z. Luo, S.B. Johnson, A.M. Lai, C. Weng, Extracting temporal constraints from clinical research eligibility criteria using conditional random fields, AMIA Annu. Symp. Proc. 2011 (2011) 843–852.

[33] C. Weng, S.W. Tu, I. Sim, R.L. Richesson, Formal representation of eligibility criteria: a literature review, J. Biomed. Inform. 43 (3) (2010) 451–467.

[34] S.B. Navathe, R.T.S.Q.L. Ahmed, A language interface for history databases, Temp. Aspects Inf. Syst. (1987).

[35] R.T. Snodgrass, The temporal query language TQuel, ACM Trans. Database Syst. (1987).

[36] J. Clifford, A. Croker, The Historical Relational Data Model (HRDM) and Algebra Based on Lifespans, ICDE, 1987.

[37] R.T. Snodgrass, The TSQL2 Temporal Query Language, Springer Science & Business Media, 1995, p. 1.

[38] W. Dorda, W. Gall, G. Duftschmid, Clinical data retrieval: 25 years of temporal query management of the university of vienna medical school, Methods Inf. Med. 41 (2) (2002) 89–97.

[39] R. Lan, H. Lee, M. Monroe, A. Fong, C. Plaisant, B. Shneiderman, Temporal search and replace: an interactive tool for the analysis of temporal event sequences 15 (2013) 1–8.

[40] K. Wongsuphasawat, C. Plaisant, M. Taieb-Maimon, B. Shneiderman, Querying event sequences by exact match or similarity search: design and empirical evaluation, Interact. Comput. 24 (2) (2012) 55–68.

[41] D.C. Nario, Secondary Use of Electronic Clinical Data: Barriers, Facilitators and a Proposed Solution, University of Washington, 2012, pp. 1–260.

[42] W. Dorda, T. Wrba, G. Duftschmid, P. Sachs, W. Gall, C. Rehnelt, et al., ArchiMed: a medical information and retrieval system, Methods Inf. Med. 38 (1) (1999) 16–24.

[43] A.K. Das, M.A. Musen, A temporal query system for protocol-directed decision support, Methods Inf. Med. 33 (4) (1994) 358–370.

[44] M.J. O'Connor, S.W. Tu, M.A. Musen, The Chronus II temporal database mediator, Proc. AMIA Symp. 567–71 (2002).

[45] D.J. Nigrin, I.S. Kohane, Data mining by clinicians, Proc. AMIA Symp. 957–61 (1998).

[46] R. Bellazzi, C. Larizza, A. Riva, Temporal abstractions for interpreting diabetic patients monitoring data, Intell. Data Anal. (1998).

[47] Y. Shahar, C. Cheng, Knowledge-based visualization of time-oriented clinical data, Proc AMIA Symp. 155–9 (1998).

[48] Y. Shahar, D. Goren-Bar, D. Boaz, G. Tahan, Distributed, intelligent, interactive visualization and exploration of time-oriented clinical data and their abstractions, Artif. Intell. Med. 38 (2) (2006) 115–135.

[49] D. Klimov, Y. Shahar, M. Taieb-Maimon, Intelligent selection and retrieval of multiple time-oriented records, J Intell Inf Syst. 35 (2) (2009 3) 261–300.

[50] D. Klimov, Y. Shahar, M. Taieb-Maimon, Intelligent visualization and exploration of time-oriented data of multiple patients, Artif. Intell. Med. Elsevier B.V. 49 (1) (2010) 11–31.

[51] J.H. Nguyen, Y. Shahar, S.W. Tu, A.K. Das, M.A. Musen, A temporal database mediator for protocol-based decision support, Proceedings of the AMIA Annual Fall Symposium, American Medical Informatics Association, 1997, p. 298.

[52] Y. Shahar, M.A. Musen, Knowledge-based temporal abstraction in clinical domains, Artif. Intell. Med. 8 (3) (1996) 267–298.

[53] D. Boaz, Y. Shahar, A framework for distributed mediation of temporal-abstraction queries to clinical databases, Artif. Intell. Med. 34 (1) (2005) 3–24.

[54] M. Balaban, D. Boaz, Y. Shahar, Applying temporal abstraction in medical, Inf. Syst. (2004) 1–15.

[55] S.B. Martins, Y. Shahar, D. Goren-Bar, M. Galperin, H. Kaizer, L.V. Basso, et al., Evaluation of an architecture for intelligent query and exploration of time-oriented clinical data, Artif. Intell. Med. 43 (1) (2008) 17–34.

[56] T. Wang, C. Plaisant, A. Quinn, R. Stanchak, S.N. Murphy, B. Shneiderman, Aligning temporal data by sentinel events: discovering patterns in electronic health records, Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, 2008, pp. 457–466.

[57] J. Jin, P. Szekely, QueryMarvel: A visual query language for temporal patterns using comic strips, 2009 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), IEEE, 2009, pp. 207–214.

[58] A. Rind, T.D. Wang, W. Aigner, S. Miksch, K. Wongsuphasawat, C. Plaisant, et al., Interactive Information Visualization to Explore and Query Electronic Health Records. Foundations and Trends in Human-Computer Interaction, Now Pub, 2013, p. 100.

[59] R. Raj, M.J. O'Connor, A.K. Das, An ontology-driven method for hierarchical mining of temporal patterns: application to HIV drug resistance research, AMIA Annu. Symp. Proc. 11 (2007) 614–619.

[60] J. Fails, A. Karlson, L. Shahamat, B. Shneiderman, A visual interface for multivariate temporal data: finding patterns of events across multiple histories, IEEE Symposium on Visual Analytics Science and Technology, 2006, pp. 167–174.

[61] C. Plaisant, S. Lam, B. Shneiderman, M. Smith, D. Roseman, G. Marchand, et al., Searching electronic health records for temporal patterns in patient histories: a Case study with microsoft amalga, AMIA Ann. Symp. Proc. 2008 (2008) 601.

[62] Szekely, Interactive Querying of Temporal Data Using a Comic Strip Metaphor, IEEE, 2010, pp. 1–8.

[63] L. Chittaro, C. Combi, Visualizing queries on databases of temporal histories: new metaphors and their evaluation, Data Knowl. Eng. (2003).

[64] C. Combi, B. Oliboni, Visually defining and querying consistent multi-granular clinical temporal abstractions, Artif. Intell. Med. 54 (2) (2012) 75–101.

[65] I.S. Kohane, S.E. Churchill, S.N. Murphy, A translational engine at the national scale: informatics for integrating biology and the bedside, J. Am. Med. Inform. Assoc. 19 (2) (2012) 181–185.

[66] G. De Moor, M. Sundgren, D. Kalra, A. Schmidt, M. Dugas, B. Claerhout, et al., Using electronic health records for clinical research: the case of the EHR4CR project, J. Biomed. Inform. 53 (2015) 162–173.

[67] J. Doods, R. Bache, M.M. McGilchrist, C. Daniel, M. Dugas, F. Fritz, et al., Piloting the EHR4CR feasibility platform across Europe, Methods Inf. Med. 53 (4) (2014).

[68] I. Soto-Rey, R. Bache, M. Dugas, F. Fritz, Query engine optimization for the EHR4CR protocol feasibility scenario, Stud. Health Technol. Inform. 192 (2013) 1080.

[69] R. Bache, S. Miles, A. Taweel, An adaptable architecture for patient cohort identification from diverse data sources, J. Am. Med. Inform. Assoc. 24 (2013).

[70] N. Mastellos, A. Andreasson, K. Huckvale, M. Larsen, V. Curcin, J. Car, et al., A Cluster randomised controlled trial evaluating the effectiveness of eHealth-supported patient recruitment in primary care research: the TRANSFoRm study protocol, Implement. Sci. 10 (2015) 15.

[71] J.-F. Ethier, O. Dameron, V. Curcin, M.M. McGilchrist, R.A. Verheij, T.N. Arvanitis, et al., A unified structural/terminological interoperability framework based on LexEVS: application to TRANSFoRm, J. Am. Med. Inform. Assoc. BMJ Publishing Group Ltd 20 (5) (2013) 986–994.

[72] T.N. Arvanitis, W. Kuchinke, TRANSFoRm query workbench, J. Clin. Bioinforma. 5 (1) (2015) 16.

[73] G. Hripcsak, J.D. Duke, N.H. Shah, C.G. Reich, V. Huser, Observational health data sciences and informatics (OHDSI): opportunities for observational researchers, Stud. Health Technol. Inform. (2015).

[74] J. Xu, L.V. Rasmussen, P.L. Shaw, G. Jiang, R.C. Kiefer, H. Mo, et al., Review and evaluation of electronic health records-driven phenotype algorithm authoring tools for clinical and translational research, J. Am. Med. Inform. Assoc. 29 (2015) ocv070.

[75] C. Haverkamp, T. Ganslandt, P. Horki, M. Boeker, A. Dörfler, S. Schwab, et al., Regional differences in thrombectomy rates: secondary use of billing codes in the MIRACUM (Medical Informatics for Research and Care in University Medicine) consortium, Clin. Neuroradiol. (2018) 1–10.

[76] V.S. Gainer, A. Cagan, V.M. Castro, S. Duey, B. Ghosh, A. Goodson, et al., The biobank portal for partners personalized medicine: a query tool for working with consented biobank samples, genotypes, and phenotypes using i2b2, JPM. 6 (1) (2016) 11.

[77] J.G. Klann, A. Abend, V.A. Raghavan, K.D. Mandl, S.N. Murphy, Data interchange using i2b2, J. Am. Med. Inform. Assoc. 5 (2016) ocv188.

[78] W. Chen, R. Kowatch, S. Lin, M. Splaingard, Y. Huang, Interactive cohort identification of sleep disorder patients using natural language processing and i2b2, ACI 6 (2) (2015) 345–363.

[79] J.G. Klann, M.D. Buck, J.S. Brown, M. Hadley, R. Elmore, G.M. Weber, et al., Query health: standards-based, cross-platform population health surveillance, J. Am. Med. Inform. Assoc. (2014).

[80] E.K. Johnson, S. Broder-Fingert, Use of the i2b2 research query tool to conduct a matched case-control clinical research study: advantages, disadvantages and methodological considerations, BMC Med. (2014).

[81] M.D. Natter, J. Quan, D.M. Ortiz, A. Bousvaros, N.T. Ilowite, C.J. Inman, et al., An i2b2-based, generalizable, open source, self-scaling chronic disease registry, J. Am. Med. Inform. Assoc. 20 (1) (2013) 172–179.

[82] Y. Girardeau, J. Doods, E. Zapletal, G. Chatellier, C. Daniel, A. Burgun, et al., Leveraging the EHR4CR platform to supportpatient inclusion in academic studies: challenges and lessons learned, BMC Med. Res. Methodol. 23 (2017) 1–10.

[83] N. Hong, N. Zhang, H. Wu, S. Lu, Y. Yu, L. Hou, et al., Preliminary exploration of survival analysis using the OHDSI common data model: a case study of intrahepatic cholangiocarcinoma, BMC Med. Inf. Decis. Making 18 (Suppl 5) (2018) 116.

[84] X. Zhang, L. Wang, S. Miao, H. Xu, Y. Yin, Y. Zhu, et al., Analysis of treatment pathways for three chronic diseases using OMOP CDM, J. Med. Syst. 42 (12) (2018) 260.

[85] R. Vashisht, K. Jung, A. Schuler, J.M. Banda, R.W. Park, S. Jin, et al., Association of Hemoglobin A1c levels with use of sulfonylureas, dipeptidyl peptidase 4 inhibitors, and thiazolidinediones in patients with type 2 diabetes treated with metformin: analysis from the observational health data sciences and informatics initiative, JAMA Netw. Open 1 (4) (2018) e181755.

[86] C. Maier, L. Lang, H. Storf, P. Vormstein, R. Bieber, J. Bernarding, et al., Towards implementation of OMOP in a German University Hospital Consortium, ACI. 9 (1) (2018) 54–61.

[87] R. Moskovitch, Y. Shahar, F. Wang, G. Hripcsak, Temporal biomedical data analytics, J. Biomed. Inform. 90 (2019 Feb) 103092.

[88] G. Hripcsak, D.J. Albers, Next-generation phenotyping of electronic health records, J. Am. Med. Inform. Assoc. 20 (1) (2012) 117–121.

[89] J. Pathak, A.N. Kho, J.C. Denny, Electronic health records-driven phenotyping: challenges, recent advances, and perspectives, J. Am. Med. Inform. Assoc. 20 (e2) (2013) e206–e211.

[90] L. Sacchi, A. Dagliati, R. Bellazzi, Analyzing complex patients' temporal histories: new frontiers in temporal data mining, Methods Mol. Biol. 1246 (2015) 89–105.

[91] C. Combi, E. Keravnou-Papailiou, Y. Shahar, Temporal Information Systems in Medicine, Springer Science & Business Media, 2010, p. 1.

[92] L. Sacchi, C. Larizza, C. Combi, R. Bellazzi, Data mining with temporal abstractions: learning rules from time series, Data Min. Knowl. Discov. (2007).

[93] Y. Shahar, Timing is everything - temporal reasoning and temporal data maintenance in medicine, AIMDM (1999).

[94] Standardized Derived Elements · OHDSI/CommonDataModel Wiki · GitHub [Internet]. github.com. [cited 2019 Aug 4], Available from: https://github.com/OHDSI/CommonDataModel/wiki/Standardized-Derived-Elements.

[95] i2b2.org. i2b2 Temporal Query Tool View [Internet]. i2b2.org. [cited 2018 May 15], Available from: https://www.i2b2.org/webclient/help/tqt.html.

[96] OHDSI. GitHub - OHDSI/Atlas: ATLAS is an Open Source Software Tool for Researchers to Conduct Scientific Analyses on Standardized Observational Data [Internet]. github.com. [cited 2018 May 15], Available from: https://github.com/OHDSI/Atlas.

[97] R. Bache, A. Taweel, S. Miles, B.C. Delaney, An eligibility criteria query language for heterogeneous data warehouses, Methods Inf. Med. 53 (4) (2014).

[98] F.A. Meineke, S. Stäubert, M. Löbe, A. Winter, A comprehensive clinical research database based on CDISC ODM and i2b2, Stud. Health Technol. Inform. 205 (2014) 1115–1119.

[99] G. Hripcsak, P. Ludemann, T.A. Pryor, O.B. Wigertz, P.D. Clayton, Rationale for the arden syntax, Comput. Biomed. Res. 27 (4) (1994) 291–324.

[100] E.T. Keravnou, Modelling medical concepts as time-objects, AIME (1995).

[101] P. Aldam, N. Levy, G.M. Hall, Perioperative management of diabetic patients: new controversies, Br. J. Anaesth. 113 (6) (2014 Dec) 906–909.

[102] R. Vreven, M. De Kock, Metformin lactic acidosis and anaesthesia: myth or reality? Acta Anaesthesiol. Belg. 56 (3) (2005) 297–302.

[103] P.M. Nadkarni, L. Marenco, R. Chen, E. Skoufos, G.M. Shepherd, P. Miller, Organization of heterogeneous scientific data using the EAV/CR representation, J. Am. Med. Inform. Assoc. 6 (6) (1999) 478–493.

[104] R.T. Snodgrass, I. Ahn, A taxonomy of time in databases, SIGMOD Conference, (1985).

[105] J. Franke, Allen temporal relationships: Implementation of Allen's Temporal Interval Relationships and the Path Consistency Algorithm in Java [Internet]. github.com. [cited 2017 Sep 29], Available from: https://github.com/jornfranke/allentemporalrelationships.

[106] E. Prud'hommeaux, A. Seaborne, SPARQL Query Language for RDF. W3C Recommendation, 2009.

[107] G. Klyne, J.J. Carroll, B. McBride, Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation, 2004.

[108] C. Bizer, A. Seaborne, D2RQ - treating Non-RDF databases as virtual RDF graphs, in: Proceedings of the 3rd International Semantic Web Conference (ISWC2004), 2004.

[109] Apache Software Foundation, Apache Jena Fuseki [Internet]. jena.apache.org. [cited 2017 Sep 29]. Available from: https://jena.apache.org/documentation/fuseki2/.

[110] S. Mate, Quick Start Guide for AllenGUI & AllenSPARQL [Internet]. github.com. 2019 [cited 2019 Aug 12]. Available from: https://github.com/sebmate/AllenSPARQL/raw/master/Quick-Start-Guide.pdf.

[111] The Sage Developers. SageMath, the Sage Mathematics Software System (Version 8.0) [Internet], 2017, Available from: http://www.sagemath.org.

[112] J. Hebeler, M. Fisher, R. Blace, A. Perez-Lopez, Semantic Web Programming, Wiley, 2009, p. 651.

[113] C. Freksa, Temporal reasoning based on semi-intervals, Artif. Intell. (1992).

[114] C. Combi, M. Franceschet, A. Peron, Representing and reasoning about temporal granularities, J. Log Comput. (2004).

[115] P. Papapetrou, G. Kollios, S. Sclaroff, D. Gunopulos, Discovering Frequent Arrangements of Temporal Intervals, IEEE, 2005, p. 8.

[116] F. Mörchen, A. Ultsch, Efficient mining of understandable patterns from multivariate interval time series, Data Min. Knowl. Disc. Springer, US 15 (2007) 181–215.

[117] S.-Y. Wu, Y.-L. Chen, Mining nonambiguous temporal patterns for interval-based events, IEEE Trans. Knowl. Data Eng. 19 (6) (2007) 742–758.

[118] R. Moskovitch, F. Polubriaginof, A. Weiss, P. Ryan, N. Tatonetti, Procedure prediction from symbolic electronic health records via time intervals analytics, J. Biomed. Inform. 17 (75) (2017) 70–82.

[119] P. Papapetrou, G. Kollios, S. Sclaroff, D. Gunopulos, Mining frequent arrangements of temporal intervals, Knowl. Inf. Syst. (2009).

[120] HL7, The Arden Syntax for Medical Logic Systems Version 2.8, 2012, pp. 1–197.

[121] G. Hripcsak, N. Elhadad, Y.-H. Chen, L. Zhou, F.P. Morrison, Using empiric semantic correlation to interpret temporal assertions in clinical texts, J. Am. Med. Inform. Assoc. 16 (2) (2009) 220–227.

[122] A.R. Post, T. Kurc, R. Willard, H. Rathod, M. Mansour, A.K. Pai, et al., Temporal abstraction-based clinical phenotyping with eureka!, AMIA Annu. Symp. Proc. 2013 (2013) 1160–1169.

[123] S. Mate, P. Vormstein, D. Kadioglu, R.W. Majeed, M. Lablans, H.-U. Prokosch, et al., On-the-fly query translation between i2b2 and samply in the german biobank node (GBN) prototypes, Stud. Health Technol. Inform. 243 (2017) 42–46.

[124] S. Mate, D. Kadioglu, R.W. Majeed, M.R. Stöhr, M. Folz, P. Vormstein, et al., Proof-of-concept integration of heterogeneous biobank IT infrastructures into a hybrid biobanking network, Stud. Health Technol. Inform. 243 (2017) 100–104.

[125] medinfo mainz - Bitbucket [Internet]. bitbucket.org. [cited 2019 Jun], Available from: https://bitbucket.org/medinfo_mainz/.

[126] Partners HealthCare, i2b2 Software Documentation. i2b2 Cell Messaging, Data Repository (CRC) Cell [Internet]. i2b2.org, 2016 [cited 2019 Jan], Available from:

https://www.i2b2.org/software/files/PDF/current/CRC_Messaging.pdf.

[127] S. Kraus, Generalizing the arden syntax to a common clinical application language, Stud. Health Technol. Inform. 247 (2018) 675–679.

[128] S. Mate, I. Castellanos, T. Ganslandt, H.-U. Prokosch, S. Kraus, Standards-based procedural phenotyping: the arden syntax on i2b2, Stud. Health Technol. Inform. 243 (2017) 37–41.

[129] E. Odigie, R. Lacson, A. Raja, D. Osterbur, I. Ip, L. Schneider, et al., Fast healthcare interoperability resources, clinical quality language, and systematized nomenclature of medicine-clinical terms in representing clinical evidence logic statements for the use of imaging procedures: descriptive study, JMIR Med Inform. 7 (2) (2019) e13590.

[130] HL7, Clinical Quality Language (CQL) [Internet]. cql.hl7.org. [cited 2019 Aug 6], Available from: https://cql.hl7.org/.