INTERNATIONAL JOURNAL OF MEDICAL INFORMATICS 78S (2009) S77–S85

journal homepage: www.intl.elsevierhealth.com/journals/ijmi

# Knowledge-data integration for temporal reasoning in a clinical trial system

*Martin J. O'Connor*[a,*], *Ravi D. Shankar*[a], *David B. Parrish*[b], *Amar K. Das*[a]

[a] *Stanford Center for Biomedical Informatics Research, Stanford University, 251 Campus Drive, MSOB X275, Stanford, CA 94305, USA*
[b] *The Immune Tolerance Network, Pittsburgh, PA, USA*

ARTICLE INFO

*Article history:*
Received 26 February 2008
Received in revised form
16 July 2008
Accepted 23 July 2008

*Keywords:*
Clinical trials
Temporal constraints
Knowledge-based systems
Semantic Web
Ontology
SWRL
OWL

ABSTRACT

Managing time-stamped data is essential to clinical research activities and often requires the use of considerable domain knowledge. Adequately representing and integrating temporal data and domain knowledge is difficult with the database technologies used in most clinical research systems. There is often a disconnect between the database representation of research data and corresponding domain knowledge of clinical research concepts. In this paper, we present a set of methodologies for undertaking ontology-based specification of temporal information, and discuss their application to the verification of protocol-specific temporal constraints among clinical trial activities. Our approach allows knowledge-level temporal constraints to be evaluated against operational trial data stored in relational databases. We show how the Semantic Web ontology and rule languages OWL and SWRL, respectively, can support tools for research data management that automatically integrate low-level representations of relational data with high-level domain concepts used in study design.



## 1. Introduction

Databases are an essential, widely used technology for biomedical research projects. In projects ranging from clinical studies to genomics research, databases are used to maintain, integrate and share data. To support study management and focused analyses, database queries must be written to extract subsets of data into specialized tools for those tasks. These data-processing steps are often customized to a particular analysis and study plan, so the database methods are difficult to reuse across research projects. A more serious shortcoming of databases is that their representation of data, such as in a relational model, does not adequately support the representation of important biomedical domain concepts, such as hierarchies and complex representations. Thus, the link between domain knowledge and data representation is often implicit. The inability to fully and explicitly model within databases temporal information about a study design (such as longitudinal patient observations or time-course experiments) can, in particular, limit the understanding of dynamic and causal phenomena that are central to scientific knowledge.

As a result, there is a critical need to provide general study management methods that can integrate *temporal domain knowledge* used in clinical research with the *temporal database schema* used to collect data. To address this challenge, we have developed methodologies and tools that permit both design-time encoding and run-time validation of domain-specific

* *Corresponding author*. Tel.: +1 650 723 7684.
E-mail addresses: martin.oconnor@stanford.edu (M.J. O'Connor), ravi.shankar@stanford.edu (R.D. Shankar), dparrish@immunetolerance.org (D.B. Parrish), amar.das@stanford.edu (A.K. Das).
1386-5056/$ – see front matter © 2008 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.ijmedinf.2008.07.013

temporal relationships. In particular, we address the problem of expressing temporal constraints for participant and sample tracking in clinical trial protocols, such as those being undertaken by the Immune Tolerance Network [1], an international research collaboration focused on developing new therapeutics in immune-mediated disorders. Our methodology bridges the gap between clinical-trial specification and clinical-trial implementation, which can improve compliance monitoring and data analysis within research environments. In this paper, we discuss the challenges of modeling clinical trial constraints, present our approach to temporal and ontology modeling of such constraints, and provide methods to allow data-knowledge integration to validate constraints.

## 2. Knowledge and database disconnect

Clinical trials are carefully planned research studies to systematically evaluate the safety and efficacy of new or unproven approaches in the prevention and treatment of medical conditions. A clinical trial study, like other types of clinical research, is generally outlined in a *protocol document* that contains items such as study objectives, study design, participant eligibility criteria, enrollment schedule, and a study plan. This document also specifies a schedule of clinical trial activities, or events, such as tests, procedures, and medications, that are to be performed.

Temporal constraints among the timing and sequence of these activities are fundamental to the descriptions of protocol entities. A temporal constraint is defined as an interval-based temporal annotation on a domain entity in terms of its relationship with other entities. Investigators may specify constraints for the purposes of general planning, gauging progression, monitoring patient safety, and managing personnel and clinical resources. Simple constraints may specify that "clinical assessments are required twice a week until day 28 or discharge from hospital"; "the first dose will be infused over a minimum of 12 h; and "visit 10 for the participant occurs 3 weeks, plus or minus 2 days, from the day of transplant." More complex constraint could state that "participants will be enrolled at least two days apart; participant is ineligible if they had a vaccination with a live virus within 6 weeks before enrollment." Following such constraints rigorously for each enrolled subject at each study site and by all study personnel is essential for a systematic undertaking of a trial, the safety of the participants, and the scientific validity of the findings that are generated.

As a result, software to support the management of a clinical trial must have knowledge of these constraints. The correct encoding and ongoing monitoring of temporal constraints is crucial to the successful execution of a clinical trial. However, when encoding these temporal constraints, developers may face two types of disconnect between their specification constraints and their execution.

(1) A knowledge specification disconnect: Constraints are typically expressed as unstructured free text throughout a study design document. Their interpretation is heavily dependent on the context of the research protocol being encoded. Even core terms – such as a definition of participant visit – can be poorly specified. Is a visit a single encounter between a participant and a provider, or can it span several encounters? If it can span several encounters, what is the exact definition of the visit end? Producing a precise definition for a constraint can thus be difficult. The unstructured process of encoding constraints can result in gaps in the final specifications.
(2) A database specification disconnect: Constraints are based on data that are collected during a study's execution and stored in relational databases. The schema design of such databases often reflects the operational requirements of the study managers, whose activities were not defined within the research protocol document, rather than data-analytic needs of the study investigators. Constraints thus must be validated against data that is modeled differently from their initial specification, with a consequent loss of precision. Often, the constraints may only be checked after data has been entered into a database and not when the data is collected, allowing study non-compliance.

As a result of these disconnects, the database schema for a clinical research study and the queries to support protocol-specified constraints may not reflect the knowledge used by study investigators. The validity and quality of the trial data being collected can thus become seriously compromised, which may not be noticed until the final stage of analysis.

To overcome these two types of disconnect, we have developed a novel knowledge-based approach that can be used to specify temporal constraints when authoring a clinical trial protocol and to monitor those constraints in an operational trial database. Based on established standards for knowledge representation and temporal-relational methods, we have implemented an ontology-based architecture for the deployment and execution of its software components.

## 3. Temporal model development

Our efforts on temporal modeling in clinical trials are based on our past informatics work on protocol-based care and the particular needs of the Immune Tolerance Network (ITN [1]), which develops new therapeutics for immune-mediated disorders. In collaboration with ITN, we have created a clinical trial modeling approach that uses ontologies to formally encode the domain knowledge needed to manage multi-site clinical trial studies and to support discovery of common tolerance mechanisms across these trials [2].

A clinical trial protocol at ITN and other research groups typically divides a protocol into *phases*, such as, for example, treatment phase and follow-up phase, and specifies the temporal sequence of the phases. It also includes a schedule of *activities* (Fig. 1) that enumerates a sequence of protocol visits that are planned at each phase, and, for each visit, specifies the time window when the visit should happen. The protocol describes the list of activities (e.g., assessments, procedures and tests) that are planned at that visit. Some activities, such as, for example, medication administration, need not be confined to visits and can be planned to occur in a time window within a protocol phase. An activity can also have sub activities that impose additional temporal constraints. For example,

an assessment activity can include collection and processing of biological specimens with associated temporal constraints.

To determine the temporal expressivity needed by clinical trial model, we undertook a survey of the types of temporal constraints encoded in 32 protocol documents used at ITN. We used the results of this survey to develop a temporal constraint model to represent these constraints [28]. We then developed a temporal-relational model to describe how temporal data can be consistently represented both within our constraint model and at the level of a relational database. These models were then incorporated into the clinical trial model.

### 3.1. *Temporal constraint model*

The core entities of the temporal constraint model are as follows.

#### 3.1.1. *Duration*

Duration is used to specify how long an activity lasts. For example, 'Visit 17 must occur at least 1 *week* but no later than 4 *weeks* after the end of 2003 ragweed season'.

#### 3.1.2. *Varying duration*

Varying Duration is defined as duration with a variance qualification. For example, 'Visit 1 should occur *2 weeks plus or minus 3 days* after transplant'.

#### 3.1.3. *Anchor*

Anchor defines an unbound time point that can be used as a reference point to define the start of another event relative to the anchor. For example, 'Visit 17 must occur at least 1 week but no later than 4 weeks after *the end of 2003 ragweed season*'. During the execution of the protocol, an anchor is bound to the absolute time of the anchor as recorded in the clinical trial database.

#### 3.1.4. *Anchored duration*

Anchored duration relates two activities with a temporal offset. For example, '*Administer Rapamune* 1 week from *visit 0* daily for 84 days'.

#### 3.1.5. *Start and end expression*

Start and end expression constrains the start and the end of an activity and is expressed as offsets before or after one or more reference events. For example, 'Screening visit evaluations must occur between *30 days prior to visit 2* and *45 days prior to visit 4*'.

#### 3.1.6. *Cyclical plan expression*

Cyclical plan expression describes events that are repeated at periodic intervals. The repetition ends typically when a specific number of cycles are reached or until a specific condition is satisfied. There are two types of cyclical plans. The first type has a single anchor point with potentially multiple intervals. For example, 'The vital signs of the participant should be obtained at routine time points *starting at 10 min post infusion, then at 20 min intervals until the participant is dicharged*'. If the participant gets off schedule because the assessment is made at minute 35 instead of minute 30, then the participant gets back on schedule with the next assessment at minute 60. This type of cyclical plan is used generally with assessments and tests where evaluations need to be made at specific intervals after a clinical intervention. The second type of cyclical plan has one or more anchors with a single offset. For example, '*Administer study medication* at *weekly intervals for 3 months*'. The initial anchor is the event of administering the first dose. According to the schedule, the second dose will be 1 week later, and the third 1 week later from the second dose. If the participant gets off schedule because the drug was administered 5 days after first dose and not 7 days, then the participant gets back on schedule with the next dose at 7 days from the last dose. This type of cyclical plan is used typically with drug administration where fixed intervals between dosages need to be maintained for safety and efficacy purposes.

#### 3.1.7. *Conditional expression*

Conditional expression allows associating different temporal constraints with a single activity based on a condition. There are three patterns of conditional expressions: *if-then*, *if-then-else* and *until-then*. An example of the if-then pattern is, '*On days that both IT and omalizumab are administered*, omalizumab will be injected 60 min after the IT.' The temporal constraint between the administrations of two drugs is dependent on the condition that the two drugs are administered on the same day. An example of the *until-then* pattern is '*Monitor cyclosporine levels 3 times per week while in-patient*, then *weekly as out-patient*.'

### 3.2. *Temporal-relational model*

Many clinical trial groups store operational data from ongoing or archived studies in relational databases. While the relational model provides a well-defined data model and query language, it has poor support for storing and querying complex temporal information. For example, if a database row contains some temporal information, there is no indication as to the relationship between the timestamp and the non-temporal data in the row. Does the timestamp refer to the point at which the information was recorded, or to the point at which it was known? Other shortcomings include the lack of a standard way to indicate a timestamp's granularity, no support for automatic coalescing or merging of temporally overlapping data, and no standard means of writing queries with relative times or that refer to the current time [3].

Several proposed extensions to the relational model address these shortcomings. Most have focused on *valid-time* model [4–6]. In this model, a piece of information – which is often referred to as a *fact* – can be associated with instants or intervals denoting the times that they are held to be true. When this model is used in a relational system, temporal information is typically attached to all rows in a temporal table, effectively adding a third dimension to two-dimensional relational tables. In these tables, every tuple holds temporal information denoting the information's valid-time. Conceptually, this representation means that every tuple is held to be true or valid during the time or times associated with this tuple.

We used the valid-time model to standardize the representation of time-stamped clinical data both in our temporal constraint models and in operational trial databases. This consistent representation of temporal information allows

**Summary of Assessments for Subjects**

| Visits | Screening Pre-Dx Review Panel SC1 | Screening Post-Dx Review Panel SC2 | Baseline -1 | Post-mobilization & Pre-conditioning PM | Day 0 (Transplant) 0 | Day +1 to +28 1[a] | Week 4 (Day 28) 2 | Week 8 (Day 56) 3 | Month 6 4 | Month 12 5 | Month 24 6 | Month 36 7 | Month 48 8 | Month 60 9[b] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Informed Consent** | | | | | | | | | | | | | | |
| Signed Screening Informed Consent | X | | | | | | | | | | | | | |
| Signed Treatment Informed Consent | | | X | | | | | | | | | | | |
| **Disease Assessments** | | | | | | | | | | | | | | |
| Confirmation of Diagnosis | X | | | | | | | | | | | | | |
| Disease History | X | | | | | | | | X | X | X | X | X | X |
| QoL Questionnaire | | | X | | | | | | X | X | X | X | X | X |
| **Medical History and Physical Exam** | | | | | | | | | | | | | | |
| Medical History | X | | | | | | | | | | | | | |
| Physical Exam and Health Assessments[d] | X | | X | X | X | X | X | X | X | X | X | X | X | X |
| Post-Mobilization or Post-Transplant Acute Toxicity Assessment | | | | X | | | X | X | X | | | | | |
| **Clinical Procedures & Assessments** | | | | | | | | | | | | | | |
| CBC with diff and platelets | | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Serum chemistries[e] | | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Pregnancy Test (female recipients)[f] | | X | X | X | | | | | | | | | | |

[a]Clinical assessments are required twice a week until Day 28 or discharge from hospital (see MOP).
[b]The Month 60 Visit is the study primary endpoint evaluation visit and the Study Completion Visit. Subjects who meet the primary endpoint (Section 3.2.1) should undergo a complete end of study evaluation at the time of meeting the primary endpoint (see Section 6.3.7.2), and will, in addition, continue to be followed on the schedule listed in Section 6.3.7.1. Study subjects withdrawn from the trial for any reason prior to the 60-month evaluation should undergo a complete end of study evaluation if possible (see Section 6.3.7.2).

**Fig. 1 – Schedule of events. Example visit and assessment specification from a protocol document.**

standard temporal operators top be applied consistently and greatly simplifies the temporal reasoning task. We have developed several SQL extensions to support these operators [4–5].

### 3.3. *Clinical trial model*

In earlier work we described the development a clinical trial model to represent the specification and implementation of a clinical trial [2]. This model has three components that relate to the management of temporal data in a trial.

#### 3.3.1. *Protocol schema*

This schema divides the temporal span of a study into phases, such as treatment phase and follow-up phase, and specifies the temporal sequence of the phases. It also includes information on the arms of a protocol.

#### 3.3.2. *Schedule of events*

This component enumerates the sequence of protocol visits that are planned at each phase, and, for each visit, specifies the time window when the visit should happen. It also lists the protocol events that are planned for each visit.

#### 3.3.3. *Specimen flow*

The specimen flow component describes the workflow associated with the processing of trial specimens. Such specimens are typically shipped from the collection sites to bio-repository sites and from there to the core laboratories where they are assayed. The temporal dimension of specimen shipping can be complex because specimens are not generally shipped as soon as they are collected but instead are grouped together in batches and are dispatched based a variety of criteria, including age, batch size, and laboratory availability.

## 4. Temporal ontology development

Using Semantic Web knowledge representation languages, we then developed ontologies to encode these temporal models.

### 4.1. *Knowledge representation languages*

The Semantic Web is a shared research plan that aims to provide explicit semantic meaning to data and knowledge on the World Wide Web [9]. The Web Ontology Language (OWL [10]) has been designed as the language of the Semantic Web. OWL can be used to build ontologies that provide high-level descriptions of Web content. These ontologies are created by building hierarchies of *classes* describing concepts in a domain and relating the classes to each other using *properties*. OWL can also represent data as instances of OWL classes – referred to as *individuals* – and it provides mechanisms for reasoning with the data and manipulating it.

The Semantic Web Rule Language (SWRL [11]) was developed to add rules to OWL. SWRL allows users to write Horn-like rules that can be expressed in terms of OWL concepts and that can reason about OWL individuals. SWRL provides deduc-

tive reasoning capabilities that can infer new knowledge from an OWL ontology. One of SWRL's most powerful features is its ability to support user-defined methods or *built-ins* that can be used in rules. A number of core built-ins for common mathematical and string operations are defined by SWRL. For example, the built-in `greaterThan` can be used to determine if one number is greater than another. For example, a simple SWRL rules to classify trial participants aged 18 or older as adults can be written:

```
Participant(?p) ^ hasAge(?p, ?age) ^
swrlb:greaterThan(?age, 17) -> Adult(?p)
```

SWRL is a rule language, not a query language. However, many ontology-based applications require the ability to extract information from ontologies in addition to reasoning with the information in those ontologies. To support this knowledge extraction, a query language called SQWRL (Semantic Query-Enhanced Web Rule Language [29]) was developed to extend SWRL to support querying of OWL ontologies.

SQWRL is implemented as a built-in library using the standard SWRL built-in mechanism. It is syntactically and semantically compatible with standard SWRL. The SQWRL built-in library contains SQL-influenced built-ins that can be used in a rule to construct retrieval specifications for information stored in an OWL ontology. For example, the following SQWRL query retrieves all participants in an ontology whose age is less than 25, together with their ages:

```
Participant(?p) ^ hasAge(?p,?a) ^
swrlb:lessThan(?a,25) -> sqwrl:select(?p,?a)
```

We used OWL and SWRL to develop the ontologies in our system. These ontologies were authored using Protégé-OWL [12], a software tool that supports the specification and maintenance of terminologies, ontologies and knowledge-bases in OWL; a plug-in called the SWRLTab [13] was used to encode and execute the SWRL rules in the ontology. Following ontology development, individual protocols were encoded using Protégé-OWL's knowledge-acquisition facilities. SQWRL was used to perform all ontology querying in the system.

### 4.2. *Valid-time ontology*

We have adapted Shahar et al.'s [7–8] temporal knowledge model to provide a valid-time representation of clinical trial data within OWL ontologies. In this model, all facts have temporal extent and are associated with instants or intervals denoting the times that they are held to be true. The core concept in the model is the *extended proposition* class that represents information that extends over time. There are two types of extended propositions in the model: (1) *extended primitive propositions* that represent data derived directly from secondary storage and (2) *extended abstract propositions* that are abstracted from other propositions.

We developed a temporal ontology in OWL to encode the valid-time temporal model. The core class modeling an entity that can extend over time in the OWL ontology is represented by an OWL class called `ExtendedProposition`. This class is associated with a property called `hasValidTimes` that holds the time or times during which the associated information is held to be true. This property is modeled by an abstract class called `Time`, which has subclasses `Instant` and `Period`. These classes represent instants and intervals, respectively. The class `Instant` is associated with the property `hasTime`, and the `Period` class, is associated with the properties `hasBeginning` and `hasFinish`. Periods and instances also have granularities associated with them. Temporal durations are modeled using a `Duration` class that holds a count and a granularity. The two types of extended propositions in the temporal model are represented in the temporal ontology by `ExtendedPrimitiveProposition` and `ExtendedAbstractProposition` classes, respectively.

These classes can be used to consistently represent temporal information in ontologies. For example, a set of visits in a protocol tracking application can be represented by defining a class called `Visit` that subclasses the extended proposition class. It inherits the `hasValidTime` property from that class, which holds its visit times. Similarly, an extended primitive proposition can be used to represent a drug regimen, with a value of type string to hold the drug name and a set of periods in the valid time property to hold drug delivery times. These extended propositions can then be associated with a class using OWL properties. Named points in time – anchor points – can be modeled as subclasses of the `Instant` class.

We integrated this temporal ontology into the clinical trial ontologies to ensure a consistent representation of temporal information and temporal constraints across all protocol activities.

### 4.3. *Temporal constraint specification*

Once all temporal information is represented consistently using the valid-time ontology, SWRL rules can be written in terms of this ontology. However, the core SWRL language has limited temporal reasoning capabilities. A few temporal predicates called *built-ins* are included in the set of standard predicates, but they have limited expressive power. Fortunately, SWRL's built-in facility provides an extension mechanism to add user-defined predicates. We used this mechanism to define a set of temporal predicates to operate on temporal values. These predicates support the standard Allen temporal operators [14].

Using these built-in operators in conjunction with the temporal ontology permits expression of temporal rules and queries. For example, in modeling visits in a protocol as extended propositions and the start of treatment of a participant as an anchor point, a SQWRL query can be written to find participants for which a second visit in a particular protocol did not occur within two weeks of the start of treatment anchor:

```
Participant(?p) ^ hasVisit(?p, ?v) ^
Visit2(?v2) ^ temporal:hasStart(?v2, ?startV2) ^
hasAnchor(?p, ?a) ^ StartOfTreatment(?a) ^
temporal:hasTime(?a, ?sot) ^
durationGreaterThan(?sot, ?startV2, 2, weeks)
-> sqwrl:select(?p)
```

SWRL and SQWRL thus allow the knowledge-level encoding of complex temporal rules and queries using concepts from the temporal ontology. These rules can freely mix temporal concepts with underlying ontology concepts representing domain entities.

In principle, SWRL rules could be used to express all constraints within the protocol tracking application. However, while relatively concise, SWRL rules are not suitable for non-specialists. Thus, we are developing a template-based knowledge acquisition interface, called TrialWiz, to allow users to enter constraints. This interface was developed using Protege-OWL's interface extension mechanism. Instead of defining constraints as low-level SWRL rules, users can select a suitable constraints template for a template set and fill in the required fields defined by the template. This template set contains definitions for common constraints and can be extended by system developers. User-selected constraint templates are then mapped automatically to SWRL rules at run time. Expert users are still able to directly write SWRL rules to express their constraints.

## 5. Knowledge-data integration method

In principle, developers could take biomedical data in an existing relational database, develop an ontology to describe those data, and then convert the data into a knowledge-based form for all future processing. However, using ontology-based software to store operational data-such as, for example, in clinical trial systems, is not currently feasible. In addition to scalability issues, his approach suffers from several other shortcomings. First, there is an issue of inconsistencies due to data duplication. There are questions about how frequently to update the knowledge base to reflect changes in associated relational database. Knowledge-driven applications requiring up-to-date information require frequent synchronization, which may be cumbersome and problematic. And, of course, if knowledge-driven updates are to be supported, the synchronization issues arise in the reverse direction.

For large data sets, a mapping solution is needed. We have developed a customized mapping tool that supports a dynamic demand-driven relational-to-OWL mapping [15]. This tool maps relation data described in terms of the temporal ontology to OWL individuals. Essentially, it creates extended propositions from time-stamped relational data.

Two OWL ontologies are used to drive this mapping: (1) a *schema ontology*, which is a knowledge-level description of a relational schema; (2) a *mapping ontology*, which describes how relational data are mapped to extended propositions. The schema ontology describes the structure of one or more databases that will be mapped. It contains descriptions of the tables in the database, such as the names of types of columns in those tables. The mapping ontology uses this schema ontology to describe the relational or temporal-relational tables to be mapped. Every extended proposition in the temporal model has an optional input and output storage descriptor. The descriptor uses the schema ontology to point to data that is stored in a database. The mapping software uses this descriptor to perform run-time transformations of the data between rows in a relational database and OWL individuals.

### 5.1. *Bridge architecture*

We have developed a bridge architecture to support the integration of relational databases and reasoning methods into a knowledge-driven system. Fig. 2 shows a schematic of the architecture and its five main components: (1) an OWL knowledge base; (2) a relational database; (3) the mapping component; (4) a method, such as a rule engine; and (5) the bridge itself. A bridge is a customized method that provides a specific computational task through the integration of one or more existing knowledge sources (such as an OWL knowledge base); data sources (such as a relational database); and data-processing mechanisms (such as a rule engine). The bridge resolves low-level differences of how these software components interact with each other through the communication of data and knowledge. A deployed bridge may work with several databases, methods, and, potentially, knowledge bases. Each bridge is driven from its associated knowledge base. The knowledge base contains a number of ontologies that are used in deploying the bridge: (1) a *method ontology*, which describes at a high level the analytic method or methods being used; (2) a mapping ontology (described above), which is used to map relational data; and (3) a *domain ontology* that describes the underlying application domain.

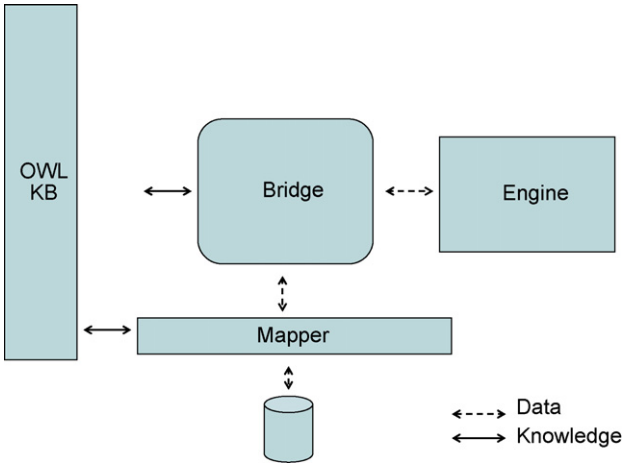


**Fig. 2 – Bridge architecture. Schematic showing how a bridge is deployed to isolate an existing method from details of both an OWL knowledge base and an existing relational database, which is accessed through a mapper.**

Providing efficient data and knowledge access techniques is a central goal of our bridge's design and implementation [15]. For example, when translating OWL knowledge into an intermediate form, instead of transferring all knowledge in a knowledge base, only potentially relevant knowledge is represented. The bridge examines each SWRL rule and only represents OWL classes, properties and individuals that are referenced by those rules. Such references can be indirect, so the bridge must traverse the interrelationships between all OWL concepts mentioned in SWRL rules to ensure completeness. This step significantly reduces the amount of knowledge that needs to be reasoned with by a rule engine and can ensure significant performance benefits.

Another optimization technique relates to data access. Extended propositions used in rules may be held in databases and accessed through the mapper. SWRL rules can operate on these propositions using temporal built-ins. There is a fairly direct mapping from SWRL rules with temporal propositions

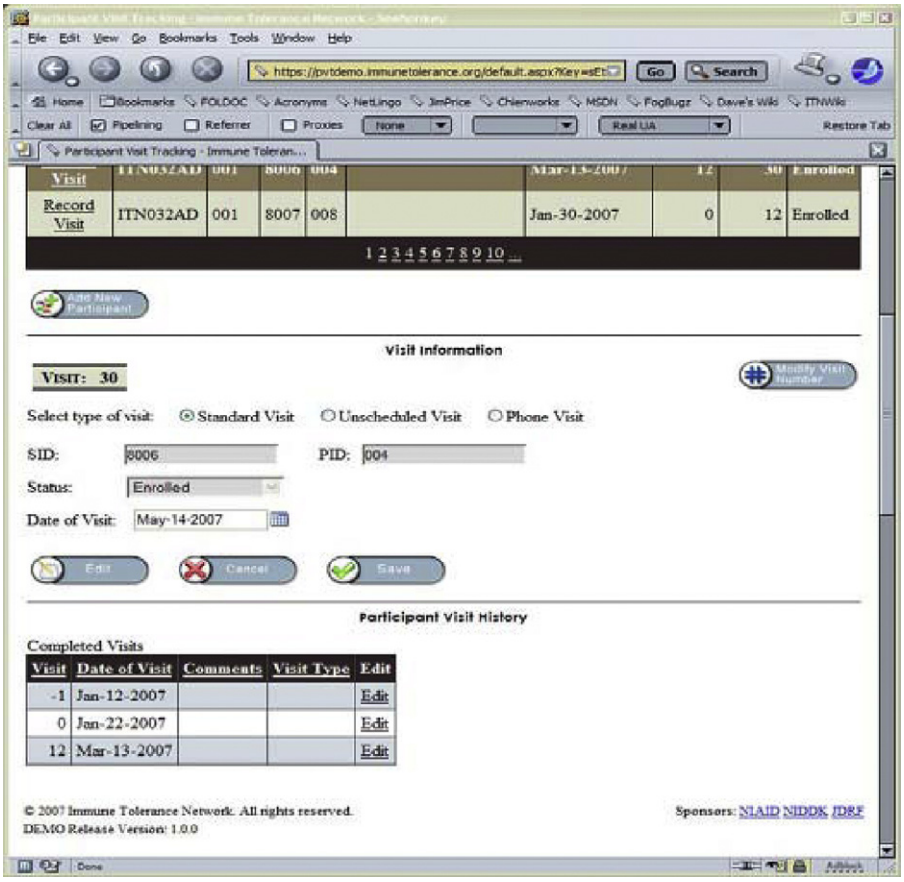


**Fig. 3 – Visit tracking application screenshot. This application allows research nurses, physicians, investigators and trial managers to examine the visit status of clinical trial participants, to determine visits to be scheduled, and to monitor visit status across trials.**

to valid-time queries. This parallel structure can be exploited by the bridge to optimize its data access. The bridge examines each SWRL rule with temporal operators and looks for operators that temporally restrict the range of propositions. For example, if a temporal operator restricts the range of a proposition to dates after a particular time point, only data after than time will be requested from the mapper. Because SWRL rules do not have disjunctions, the analysis required for this optimization process is not computationally expensive.

## 6. Deployment of a visit tracking application

To support the validation of temporal constraints for clinical-trial management at ITN, we have used our system to develop a visit tracking application (see Fig. 3). This application allows research nurses, physicians, investigators and trial managers to examine the visit status of clinical trial participants. Depending on their levels of authorization, users can use the application to manage various aspects of visit tracking in clinical trials being carried out by the ITN. For example, research nurses or physicians at trial sites can use to application at participant visit time to determine the visits previously attended by those participants. The applications will also display follow-up visits and indicate when they should be scheduled. Participants that violate trial constraints are also displayed. Summary statistics are presented for investigators to examine the number of participants at various stages in a clinical trial protocol. Trial managers can display data for multiple trials and cross-trial summary statistics.

The bridge architecture to provide the infrastructure necessary to execute temporal queries generated by the visit tracking application and to execute constraints specified in the schedule of events ontology. As described in Section 4.3, these constraints are ultimately encoded as SWRL rules; queries are expressed in SQWRL. At run-time, the visit tracking application sends these rules and queries to the bridge for evaluation. The bridge uses the mapping layer to retrieve appropriate data from ITN's operational databases so that the rules and queries can be evaluated. It then sends that data, the relevant domain concepts from the clinical trial ontology, and the SWRL rules or SQWRL queries to a rule engine for execution. The rule engine takes these entities, performs inference, and sends the results back to the bridge. The bridge translates these results to their OWL representation and returns them to the invoking visit tracking application. The relational mapping process is completely hidden from ITN applications; from their viewpoint, all information processing is at the knowledge level.

There are overheads associated with executing queries through a bridge instead of expressing them directly in SQL. The performance penalty is approximately a factory a 10 for the types of queries typical in the visit tracking application. However, since the ITN is responsible for a limited number of trials and most trials have an average of a few dozen participants the added query processing latency is barely perceptible to users. The application of these querying techniques to significantly greater quantities of data will require optimizations. We have developed a number of optimization strategies [15] are actively pursing others so that we can apply these techniques to the analysis of the significant quantities of ITN's clinical trial data.

## 7. Related work

There have been a number of previous systems that use formal models to represent clinical constraints in clinical systems [16–19]. Systems such as EON [20], PROforma [21], and GLIF [22] use models that represent temporal constraints on patient data and on activities found in clinical guidelines. In the area of clinical trials, several recent modeling efforts have addressed the temporal requirements of trial management activities. These systems automate different clinical trial management activities such as eligibility determination, participant tracking, site management, and data analysis. For example, an ontology to represent temporal information and cyclical event patterns in clinical trial protocols has been proposed by Weng et al. [23]. The Trial Bank Project [24] also tackles the problem of representation of the temporal dimension of clinical trials. Trial Bank is a trial registry that uses a protocol ontology to capture information on randomized clinical trials such as intervention, outcomes, and eligibility criteria. The underlying knowledge base can support systematic reviewing and evidence-based practice. Temporal analysis is a central component in this knowledge base.

There is an ongoing effort by CDISC [25], an industry-lead, multidisciplinary organization, to develop and support the electronic acquisition, exchange, submission and archiving of clinical trials data. As part of this effort, CDISC is developing the Trial Design Model (TDM) that identifies standard elements of a clinical trial protocol that can be codified to facilitate the data interchange among systems and stakeholders including regulatory authorities, biopharmaceutical industry, statisticians, and project managers. A parallel effort is the BRIDG [26] project, a partnership of several organizations including CDISC, the HL7 [27] standards body, the National Cancer Institute and the Federal Drug Administration, that consumes the Trial Design Model work to build a comprehensive domain analysis model representing protocol-driven biomedical/clinical research. The BRIDG model is a work in progress to elaborately define functions and behaviors throughout clinical trials, and uses the Unified Modeling Language (UML) for knowledge representation. The model currently lacks formalization of and reasoning with temporal constraints. We are collaborating with the authors of the BRIDG model to incorporate the temporal components of our clinical trial model into their system.

**Summary points**

What was known before the study:

- Managing time-stamped data is essential to clinical research activities and often requires the use of considerable domain knowledge.
- Adequately representing and integrating temporal data and domain knowledge is difficult with the database technologies used in most clinical research systems.
- There is often a disconnect between the database representation of research data and corresponding domain knowledge of clinical research concepts.
- As a result, the gap between the specification of a study protocol and the management of resulting data can often be significant in these systems.

What the study has added to the body of knowledge:

- To help close this gap, we have developed a set of general methodologies for managing temporal information at the knowledge level rather than the database level.
- Our approach allows knowledge-level temporal constraints to be evaluated against operational data stored in relational databases.
- We show how the Semantic Web ontology and rule languages OWL and SWRL, respectively, can support tools for research data management that automatically integrate low-level representations of relational data with high-level domain concepts used in study design.

## 8. Discussion

The gap between the specification of a study protocol and the management of resulting data can often be significant in clinical research systems. To help close this gap, we have developed a set of general methodologies for specifying and executing temporal constraints at the knowledge level rather than the database level. Our approach demonstrates that proposed Semantic Web standards for ontology and rule representation, OWL and SWRL, respectively, can support the knowledge model needed to integrate temporal representations of relational data with the domain-specific semantics needed to reason with them for biomedical and healthcare applications.

In contrast to previous work on constraint specification in clinical trials, our set of methodologies addresses the knowledge and database disconnect that exist in clinical research systems. Our approach requires that all relevant temporal knowledge on a study protocol and its corresponding data representation be encoded within an OWL ontology, which allows the uniform specification of temporal patterns in knowledge-level querying. Our bridge architecture supports robust optimization techniques to ensure that encoded constraints are automatically translated into an executable form at run time and are efficiently validated against study data held in an existing relational database.

## Acknowledgements

This research was supported in part by the Immune Tolerance Network, funded by Grant NO1-AI-15416 from the National Institutes of Health (USA), and by a Pharmaceutical Research and Manufacturers Association Foundation Research Starter Grant. The authors thank Valerie Natale for her editorial comments.

## REFERENCES

[1] D. Rotrosen, J.B. Matthews, J.A. Bluestone, The immune tolerance network: a new paradigm for developing tolerance-inducing therapies, J. Allergy Clin. Immunol. 110 (1) (2002) 17–23.
[2] R. Shankar, S.B. Martins, M.J. O'Connor, D. Parrish, A.K. Das, Towards semantic interoperability in a clinical trials management system, in: Fifth International Semantic Web Conference, Athens, Georgia, 2006, pp. 901–912.
[3] R.T. Snodgrass, On the semantics of 'now' in databases, ACM Trans. Database Syst. 22 (2) (1997) 171–214.
[4] R.T. Snodgrass (Ed.), The TSQL2 Temporal Query Language, Kluwer Academic Publishers, Boston, 1995.
[5] M.J. O'Connor, S.W. Tu, M.A. Musen, The Chronus II temporal database mediator, in: AMIA Annual Symposium, San Antonio, TX, 2002, pp. 567–571.
[6] A.K. Das, M.A. Musen, Synchronus: a reusable software module for temporal integration, in: AMIA Annual Symposium, San Antonio, Texas, 2002, pp. 195–199.
[7] Y. Shoham, Temporal logics in AI: semantical and ontological considerations, Artif. Intell. 33 (1) (1987) 89–104.
[8] Y. Shahar, H. Chen, D.P. Stites, L. Basso, H. Kaizer, D.M. Wilson, M.A. Musen, Semi-automated entry of clinical temporal-abstraction knowledge, J. Am. Med. Inform. Assoc. 6 (6) (1999) 494–511.
[9] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Sci. Am. 35 (May) (2001) 43–52.
[10] OWL Web Ontology Language Reference, www.w3.org/TR/owl-ref, 2004.
[11] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C, May 21, 2004.
[12] H. Knublauch, R.W. Fergerson, N.F. Noy, M.A. Musen, The Protégé OWL Plug-in: an open development environment for Semantic Web applications, in: Third International Semantic Web Conference (ISWC 2004), Hiroshima, Japan, 2004, pp. 229–243.
[13] M.J. O'Connor, H. Knublauch, S.W. Tu, B. Grosof, M. Dean, W.E. Grosso, M.A. Musen, Supporting rule system interoperability on the Semantic Web with SWRL, in: Fourth International Semantic Web Conference (ISWC 2005), Galway, Ireland, 2005, pp. 974–986.
[14] J.F. Allen, Maintaining knowledge about temporal intervals, Commun. ACM 26 (11) (1993) 832–843.
[15] M.J. O'Connor, R.D. Shankar, S.W. Tu, C. Nyulas, M.A. Musen, A.K. Das, Using Semantic Web technologies for knowledge-driven queries in clinical trials, in: Proceedings of the 11th Conference on Artificial Intelligence in Medicine, Amsterdam, Netherlands, 2007.
[16] C. Weng, M. Kahn, J. Gennari, Temporal knowledge representation for scheduling tasks in clinical trial protocols, in: Proceedings of the AMIA Annual Symposium, San Antonio, Texas, November 2002, pp. 879–883.
[17] A.M. Deshpande, C. Brandt, P.M. Nadkarni, Temporal query of attribute-value patient data: utilizing the constraints of clinical studies, Int. J. Med. Inform. 70 (1) (2003) 59–77.
[18] P. Terenziani, Toward a unifying ontology dealing with both user-defined periodicity and temporal constraints about repeated events, Comput. Intell. 18 (3) (2002) 336–385.
[19] C. Bettini, S. Jajodia, X. Wang, Solving multi-granularity constraint networks, Artif. Intell. 140 (1–2) (2002) 107–152.
[20] M.A. Musen, S.W. Tu, A.K. Das, Y. Shahar, EON: a component-based approach to automation of protocol-directed therapy, J. Am. Med. Inform. Assoc. 3 (6) (1996) 367–388.
[21] J. Fox, N. Johns, A. Rahmanzadeh, R. Thomson, PROforma: a method and language for specifying clinical guidelines and protocols, in: Proceedings of Medical Informatics Europe, Amsterdam, 1996.
[22] A.A. Boxwala, M. Peleg, S.W. Tu, O. Ogunyemi, Q.T. Zeng, D. Wang, V.L. Patel, R.A. Greenes, E.H. Shortliffe, GLIF3: a representation format for sharable computer-interpretable clinical practice, J. Biomed. Inform. 37 (3) (2004) 147–161.
[23] C. Weng, M. Kahn, J.H. Gennari, Temporal knowledge representation for scheduling tasks in clinical trial protocols, in: Proceedings of the AMIA Annual Symposium, San Antonio, TX, 2002, pp. 879–883.
[24] I. Sim, B. Olasov, S. Carini, The Trial Bank system: capturing randomized trials for evidence-based medicine, in: Proceedings of the AMIA Annual Symposium, 2003.
[25] CDISC, http://www.cdisc.org/standards/, 2008.
[26] BRIDG, http://www.bridgproject.org/, 2008.
[27] HL7, http://www.hl7.org/, 2008.
[28] R. Shankar, S.B. Martins, M.J. O'Connor, D. Parrish, A.K. Das, An ontological approach to representing and reasoning with temporal constraints in clinical trial protocols, in: Best Papers of Biomedical Engineering Systems and Technologies. Communications in Computer and Information Science Series, Springer, 2008.
[29] M.J. O'Connor, S.W. Tu, C.I. Nyulas, A.K. Das, M.A. Musen, Querying the Semantic Web with SWRL, in: The International RuleML Symposium on Rule Interchange and Applications (RuleML2007), Orlando, FL, Springer–Verlag, 2007.