

Towards Semantic Interoperability in a Clinical Trials Management System

Ravi D. Shankar¹, Susana B. Martins¹, Martin J. O'Connor¹, David B. Parrish²,
and Amar K. Das¹

¹ Stanford Medical Informatics, Stanford University School of Medicine,
Stanford, CA 94305

ravi.shankar@stanford.edu

² The Immune Tolerance Network, Pittsburgh, PA

Abstract. Clinical trials are studies in human patients to evaluate the safety and effectiveness of new therapies. Managing a clinical trial from its inception to completion typically involves multiple disparate applications facilitating activities such as trial design specification, clinical sites management, participants tracking, and trial data analysis. There remains however a strong impetus to integrate these diverse applications – each supporting different but related functions of clinical trial management – at syntactic and semantic levels so as to improve clarity, consistency and correctness in specifying clinical trials, and in acquiring and analyzing clinical data. The situation becomes especially critical with the need to manage multiple clinical trials at various sites, and to facilitate meta-analyses on trials. This paper introduces a knowledge-based framework that we are building to support a suite of clinical trial management applications. Our initiative uses semantic technologies to provide a consistent basis for the applications to interoperate. We are adapting this approach to the Immune Tolerance Network (ITN), an international research consortium developing new therapeutics in immune-mediated disorders.

1 Introduction

Clinical trials are carefully-controlled research studies in human patients to systematically evaluate the safety and efficacy of new or unproven approaches in the prevention and treatment of medical conditions. The lifecycle management of a complex clinical trial typically involves multiple applications facilitating activities such as trial design specification, clinical sites management, laboratory management, and participants tracking. These disparate applications are banded together as a clinical trial management system. The information generated by these applications along with data from loosely controlled sources such as spreadsheets, documents and email messages are then assembled to determine the operational state of the clinical trial. The lack of common nomenclature among the different sources of the tracking information and the unreliable nature of the data generation can lead to significant operational and maintenance challenges. The applications support different but related aspects of a clinical trial, and require clinical trial data flow and knowledge exchange between the applications. Thus, there is a strong impetus to integrate these diverse applications at syntactic, structural and semantic levels so as to improve clarity,

consistency and correctness in specifying clinical trials, and in acquiring and analyzing clinical data. The situation becomes especially critical with the need to manage complex clinical trials at various sites, and to facilitate meta-analyses on across the different trials.

We present, Epoch, a knowledge-based approach to support a suite of clinical trial management applications. Our initiative uses semantic technologies to provide a consistent basis for the applications to interoperate. We are adapting this approach to the Immune Tolerance Network^{1,2} (ITN), an international consortium that aims to accelerate the development of immune tolerance therapies through clinical trials and integrated mechanistic (biological) studies. The ITN is involved in planning, developing and conducting clinical trials in autoimmune diseases, islet, kidney and liver transplantation, allergy and asthma, and operates more than a dozen core facilities that conduct bioassay services. Many groups, internal and external to ITN, collaborate in facilitating the specification and implementation of the trials and related biological assay studies. Therefore, the successful conduct of a clinical trial depends upon the interaction of professionals working for various entities, including the ITN, contract research organizations, clinical study sites, and core laboratories. Studies need to be tracked for the purposes of general planning, gauging progression, monitoring patient safety, and managing personnel and clinical resources. The management effort is especially compounded by the fact that an ITN trial often is carried out at multiple sites, geographically distributed, sometimes across the world.

The Epoch framework is being collaboratively developed by the Stanford Medical Informatics (SMI) and the ITN in addressing the informatics needs of collecting, managing, integrating and analyzing clinical trial and immunoassay—a special laboratory procedure— data. Figure 1 illustrates a set of clinical trial management applications that we have identified to manage ITN’s clinical trials. At the core of our framework is a suite of ontologies that conceptualizes the clinical trial domain. The ontologies along with semantic inferences and rules provide a common protocol definition for the applications to use to interoperate semantically. In this paper, we

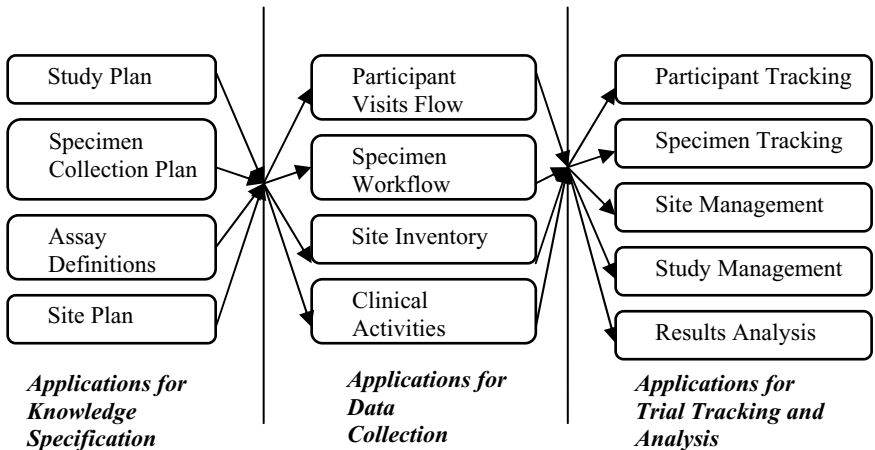


Fig. 1. A set of clinical trial management applications. The arrows indicate knowledge exchange and data flow.

present the strong ontological foundation of our approach, and describe the Epoch components. We illustrate the use of our framework in supporting the semantic interoperability of a subset of the clinical trial management applications to support specimen tracking.

2 The Epoch Core Ontologies

A clinical trial protocol (the plan for a trial) lays out specification, implementation and data analysis details. For example, it includes the reason for undertaking the study, the number of participants that will be in the study and the recruitment process, the sites (clinical and laboratory) where the study will be conducted, the study drug that the participants will take, the medical tests that the participants will undergo, the data that will be collected, and the statistical analyses that will be performed on the data. We highlight four pieces of protocol definitions that are required to support these activities. The *protocol schema* divides the temporal span of the study into phases such as the treatment phase and follow-up phase, and specifies the temporal sequence of the phases. It also includes information on the arms of the protocol. The *schedule of events* 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 and a list of protocol events (assessments, procedures and tests) that are planned at that visit. The *specimen table* lists the clinical specimens that will be collected from the participant, the visits at which they will be collected, the processing and storage conditions, and the assays —special tests— that will be performed on them. The *specimen flow* describes the workflow associated with the processing of the specimens. The specimens are typically shipped from the collection sites to bio-repository sites and, from there to the core laboratories where they are assayed.

We recognize that a structured and standardized knowledge representation that conceptualizes the protocol entities relevant to our management applications is crucial to the interoperability of these applications. We created a suite of ontologies that provide a common nomenclature and semantics of protocol elements and that spans the entire clinical trials process:

- The *clinical ontology* includes terms that specify clinical and biological knowledge on immune tolerance disorders and other concepts relevant to ITN clinical trials.
- The *protocol ontology* is a knowledge model of the clinical trial protocol. It simplifies the complexity inherent in the full structure of the protocol by focusing only on concepts required to support clinical trial management. Other concepts are either ignored or partially represented. The main concepts represented in the protocol ontology are the protocol schema and the schedule of events.
- The *assay ontology* models characteristics of mechanistic studies relevant to immune disorders. An assay specification includes the clinical specimen that can be analyzed using that assay, and the processing instructions at the core laboratories.
- The *specimen ontology* models the workflow of specimens – collection, shipping and processing workflow of specimens at the clinical, laboratory, and bio-repository sites.

- The *specimen container* ontology catalogs the different specimen containers such as tubes and slides, and the attributes of each container such as material, size, manufacturer, specimen types, additives, etc. Ancillary ontologies define different specimen types and additives.
- The *site ontology* provides a structure to store site-related data such as protocols implemented at the site, participants on each protocol, relevant clinical resources and personnel.
- The *virtual data ontology* encapsulates the study data that is being collected, such as participant clinical record, specimen workflow logs, and site related data. A mapping component can then map clinical trial data (found in a relational database) to these virtual data records using a mapping ontology. The data model concept is similar to the Virtual Medical Record³ (VMR) specification promoted in the clinical guideline modeling efforts.
- The *temporal ontology*⁴ provides a uniform representation of all temporal information in our models.

We have developed these ontologies in OWL⁵ —the Web Ontology Language proposed by W3C— by building hierarchies of *classes* describing concepts in the ontologies and relating the classes to each other using *properties*. OWL can also represent data as instances of OWL classes —referred to as *individuals*— and also provides mechanisms for reasoning with the data and manipulating it. OWL also provides a powerful constraint language for precisely defining how concepts in ontology should be interpreted. The Semantic Web Rule Language⁶ (SWRL) allows

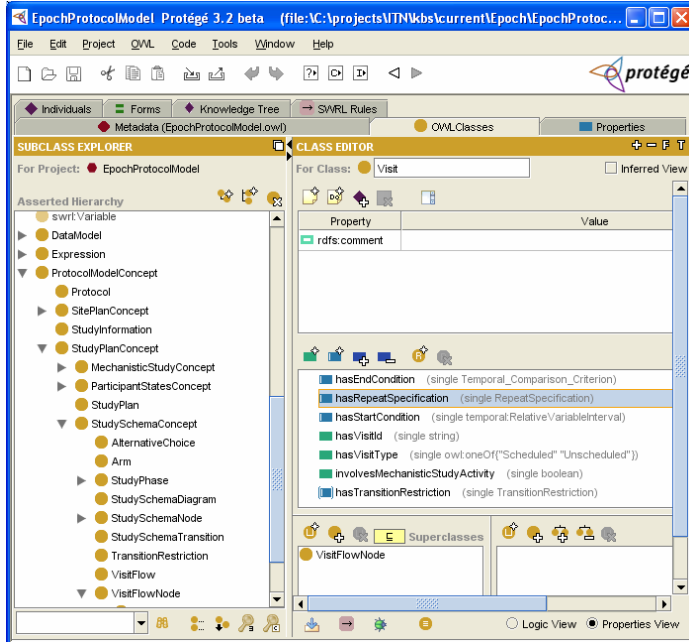


Fig. 2. The Protégé-OWL editor displaying part of the *Protocol ontology*

users to write Horn-like rules that can be expressed in terms of OWL concepts and that can reason about OWL individuals. SWRL provides deductive reasoning capabilities that can infer new knowledge from an existing OWL knowledge base. We use SWRL to specify temporal constraints and rules found in our ontologies in terms of the temporal model. Using SWRL's built-in extension facility, we have implemented a rich library of temporal operators to write rules to express complex temporal constraints. Protégé^{7,8} is a software tool that supports the specification and maintenance of terminologies, ontologies and knowledge-bases in OWL. It has a plug-in called SWRL Tab⁹, an editor for SWRL rules. We used Protégé to create the ontologies in OWL and SWRL (Figure 2). We, then, entered specific protocols and assays using Protégé's knowledge-acquisition facilities.

3 Components of the Epoch Framework

The Epoch framework broadly supports three types of methods that applications can use to support clinical trial management activities. The knowledge acquisition methods allow users to encode specific protocols and related operational elements, and thus, to create the protocol knowledge base. Ontology-database mapping methods integrate the protocol and biomedical knowledge with clinical trial data including clinical results and operational data stored in the ITN data repository. Concept-driven querying methods support integrated data management, and can be used to create high-level abstractions of clinical data during analysis of clinical results. At the center of all these methods and the applications that use these methods is the suite of Epoch ontologies that we have described in Section 2.

The *Epoch Knowledge Base* contains the ontologies enumerated in Section 2. It also stores specific instantiations of the ontologies for different clinical trials. The repository uses a file backend to store the OWL ontologies in XML format. The SWRL rules are stored as part of the knowledge base. Here is an example of a SWRL rule that is used to set a participant's time of transplant:

```

Observation(?o) ^
associatedVisitRecord(?o, ?vrecord) ^
hasParticipantId(?vrecord, ?pid) ^
hasCode(?o, ?code) ^
swrlb:equal(?code, "transplant") ^
temporal:hasValidTime(?o, ?vtO) ^
TemporalAnchorPoint(?a) ^
hasName(?a, "Transplant")
→ temporal:hasValidTime(?a, ?vtO)

```

Significant events in a protocol such as the transplant time are annotated as temporal anchor points. By definition, other events are temporally constrained by anchor points. For e.g., a visit time window (when the visit should happen) can be specified as a temporal constraint based on the anchor point 'transplant'. The example SWRL rule associates data model concepts such as Observation and VisitRecord to protocol model concepts such as TemporalAnchorPoint. The execution of the rule will set the value of the temporal anchor point 'transplant' with the transplant time for the participant found in the clinical trial database.

A *Knowledge Base Server* provides a programmatic interface (API) that other components can use to access the contents of the ontology repository. We are developing a protocol domain specific API on top of the generic Protégé-OWL API. We have developed a tool to generate XML renditions of the OWL knowledge base based on custom XML Schema. In Section 4, we show how we employed this tool to configure a data collection application with information in the knowledge base. We are building other utility tools to support querying and rule execution. These tools will eventually be integrated with the knowledge base server.

The *Clinical Trial Database* is a relational database system that stores data related to the implementation and execution of clinical trials. The types of data include participant enrollment data, specimen shipping and receiving logs, participant visits and activities, and clinical results.

The *Model-Data Mapper* facilitates runtime access to relational data in the clinical trial database as instances of the Epoch data model. It uses a mapping ontology to connect data model concepts to database entities i.e. properties of an OWL class are mapped to columns of a relational table.

The *Inference / Rule Engine* executes temporal and non-temporal constraints – that have been expressed as SWRL rules – in Epoch ontologies. We have developed a SWRL *built-in* deployment module⁹ that provides a general mechanism to define Java implementations of SWRL built-ins, dynamically load them, and invoke them from a rule engine. We used this mechanism to define a set of temporal predicates to operate on temporal values. These predicates support the standard Allen¹⁰ temporal operators such as *before*, *during*, *starts*, *ends*, *inside*, *overlaps*, *before and after*. The interface with the Model-Data Mapper allows SWRL rules to be executed on data stored in the clinical trial database. Here is an example of a SWRL rule to check if a participant's visit time fell within that visit's time window:

```
VisitRecord(?vrecord) ^
hasVisitId(?vrecord, ?vid1) ^
hasParticipantId(?vrecord, ?pid) ^
temporal:hasValidTime(?vrecord, ?vtO) ^
Visit(?v) ^
hasVisitId(?v, ?vid2) ^
swrlb:equal(?vid1, ?vid2) ^
hasStartCondition(?v, ?vsc) ^
temporal:inside(?vtO, ?vsc)
→
```

The empty head of the rule indicates that this rule is formulated as a query. This rule uses a built-in *temporal:inside* that takes in as arguments a time and a relative variable interval, and returns *true* if the time point is within the interval, and returns *false* otherwise. The *relative variable interval* concept is expressed in terms of a temporal anchor point. It is defined as

temporal anchor point + offset (+ high variance/- low variance)

Using the relative variable interval, we can specify visit time windows like

transplant time + 28 days with a variance of plus or minus 3 days

We are currently using JESS¹¹, a production rule-engine, to selectively execute the rules based on the context. For example, the rule that specifies the constraint on a visit time window will alone need to be executed when checking if a specific participant's visit satisfied the constraint.

The *Clinical Trial Management Applications* are a suite of applications as shown in Figure 1. These applications interoperate via the Epoch components at syntactic, structural and semantic levels to support the management of clinical trials.

4 An Example Usage Scenario – Specimen Collection and Tracking

Clinical specimens are collected from participants at different visits based on clinical assessments and clinical studies (biological assays) planned in the protocol. These specimens are then processed and stored in pre-determined containers and shipped to bio-repositories. The specimens (or portions of them) are shipped to the core laboratories that can perform specific assays on the specimens. The assay results are then sent to a data warehouse for storage and subsequent analysis. The bio-repositories may also archive portions of the specimens for future interrogation. The trials managed by ITN generate enormous amount of specimen traffic across different sites. Tracking the specimen from the point of collection to the point of processing and archival becomes paramount to maintain the integrity of the operation. Appropriate type and number of specimen containers should be stocked at the clinical

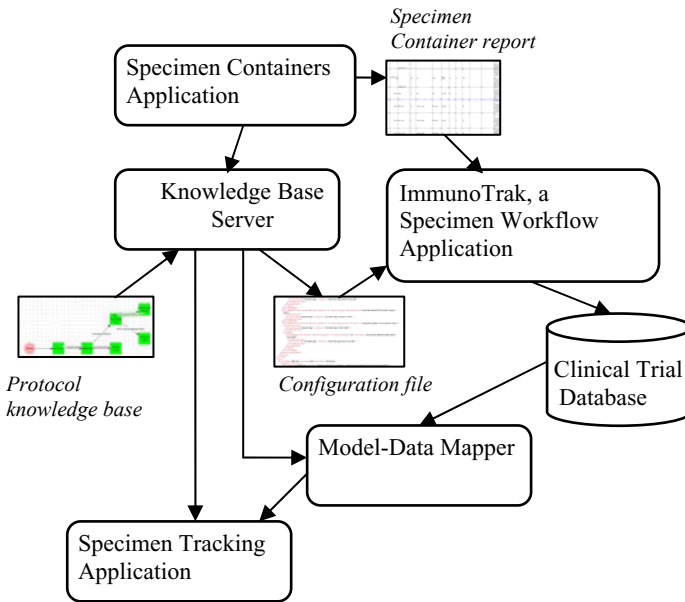


Fig. 3. A high-level view of semantic interoperation among Epoch architectural components and applications

sites in preparation for the anticipated participant visits. At the time of a participant's visit, appropriate specimens should be collected and stored in matching containers. The containers are shipped to the bio-repositories, and then to the core laboratories based on the shipping instructions in the specimen table and the specimen flow of the protocol. Specimens have to be accounted for at all times using shipping and receiving logs.

The ITN has contracted with Cimarron Software, Inc.¹² to build a specimen workflow system called ImmunoTrak based on Cimarron's Laboratory Workflow Systems product. Clinical trial personnel at the sites will use the system to log participant's visit, specimen collection, shipping and receiving of bar-coded specimen containers, etc. ImmunoTrak can be configured using a graphical user interface or via an XML specification. The configuration parameters include, the participant visit flow, the specimen container specification, list of participants, list of clinical and laboratory sites, and specimen workflow. The system should also be configured with the container manufacturer's report on the empty specimen containers shipped to the collection sites. The specimen tracking data that is collected by the system during the course of the trial is stored in a relational database. A Specimen Tracking application can then access the database to monitor the status of the specimen collection and processing.

Figure 3 shows the usage scenario employing the Epoch framework to specimen tracking. The first step is to specify the specimen workflow in the Protégé-OWL editor using relevant Epoch ontologies. Next, the Knowledge Base Server uses an XML Schema file to generate the configuration file for ImmunoTrak, the Specimen Workflow System. The Specimen Container application generates container specifications that form the basis of the manufacturer's report. During the course of the clinical trial, research coordinators at different clinical sites access ImmunoTrak to enter specimen collection data which is stored in the Clinical Trial Database. The Specimen Tracking Application employs the Model-Data Mapper to access the data via the Epoch data models. It can then satisfy user queries for specific specimen processing status, specimen collection inventory. It can also execute any validation rules or temporal constraints as specified in the ontologies on the tracking data using the production rule engine.

All the applications in this example work on the same set of semantic descriptions of specimen workflow concepts found in the Epoch knowledge base. The applications are built by different vendors, and are pressed into service at different stages – specification, execution and monitoring – of the clinical trial. The Epoch ontologies are the foundation that scans across these disparate applications. The semantic chaining of the applications, from protocol specification, to data collection, to data analysis can improve standardization, data integrity, data integration and data analysis.

5 Related Work

In the past few years, we have seen considerable interest in building knowledge-based systems that automate clinical trial protocols and clinical practice guidelines. The Epoch framework employs a task-based paradigm that combines an explicit representation of the clinical trial domain with rules that capture the logical conditions

found in the trial management process. There have been a number of proposals on task-based clinical guideline representation formats – EON¹³, PROforma¹⁴, GLIF¹⁵, etc. In our laboratory, at Stanford Medical Informatics, we have developed the EON guideline models that are used to build guideline-based decision support systems. These advisory systems provide patient care that is informed by the guidelines. The Epoch models have adapted some of the representational formalisms – workflow, expressions, and coded concepts – found in the EON models. In the area of clinical trials, several modeling efforts have addressed different requirements of trial management activities. The Trial Bank Project¹⁶ is a trial registry that captures information on randomized clinical trials such as intervention, outcomes, and eligibility criteria. The underlying knowledge base can support systematic reviewing and evidence-based practice. Design-A-Trial¹⁷ (DAT) enables the design and specification of protocols that can be exported to software tools for management of clinical trials. DAT presents users with forms in which to enter data describing a trial. It critiques the data entry using a trial domain knowledge base and guides the user to design a clean protocol devoid of inconsistencies. Currently, the Epoch users interact with the generic knowledge-acquisition forms that Protégé provides to specify a protocol. We plan to create rich graphical user interfaces coupled with DAT-like guidance mechanism that will lead our users on custom design pathways based on the restrictions and rules defined in the Epoch knowledge base. The PROforma language, just like the EON models, can capture the structure and content of a clinical guideline, and has been embedded in a commercially available clinical trial management system. The intent of the PROforma knowledge base is mainly to drive patient therapy during clinical trials, in contrast to the Epoch knowledge base that supports trial management activities.

There is an ongoing effort by CDISC¹⁸, 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, project managers, etc. A parallel effort is the BRIDG¹⁹ project, a partnership of several organizations including CDISC, the HL7²⁰ 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 representation. The model, in its current state, cannot fully support the requirements of ITN's trial management. However, we are closely following the development of the BRIDG model, and incorporating the model's semantic descriptions of clinical trials – concepts relevant to our trial management activities – within Epoch.

6 Discussion

The increasing complexity of clinical trials has generated an enormous requirement for knowledge and information management at all stages of the trials – planning,

specification, implementation, and analysis. Our focus is currently on two application areas: (1) tracking participants of the trial as they advance through the studies, and (2) tracking clinical specimens as they are processed at the trial laboratories. The core of the Epoch framework is a suite of ontologies that encodes knowledge about the clinical trial domain that is relevant to trial management activities. We used OWL to specify the ontologies, and SWRL rules written in terms of concepts in these ontologies to express any constraints. The Epoch ontologies, especially the *Protocol ontology*, have been influenced by past and ongoing modeling work. Our laboratory has demonstrated in previous projects, the viability of frame-based languages to build knowledge based systems. So, it begs the question: *How does our current approach of using OWL/SWRL to build knowledge based systems compare to our experience with frame-based languages?* A recent work²¹ compares the two knowledge-representation languages at the syntactic and semantic levels, and suggests appropriate usage situations. The EON guideline decision support architecture uses a frame-based language to specify guidelines, and an interpreter to execute the guidelines on specific patient data. We spent significant effort in developing custom interpreters to execute domain-specific logic. We do not know if the combination of OWL, SWRL, Jena, and JESS will obviate the need to build custom interpreters for our clinical trial management applications. However, we view the growing interest in the OWL standards, and the plethora of tools and software packages as a significant practical advantage of using OWL and SWRL over frame-based languages.

Native RDF Store (storing data as RDF triples) has advanced recently in performance and scalability. It would have been a natural solution for us to use RDF store for storing clinical trial data, and then seamlessly operate on the data using our OWL ontologies and SWRL rules. ITN uses a legacy relational database system to store clinical trial data, and therefore, prevents us from using native RDF Stores as our backend. We have to devise ways to map the database tables to our data model OWL classes. In a previous project, the BioSTORM disease-surveillance framework²² employs techniques to map disparate data sources to a data model. These techniques were developed using a frame-based language and we are translating these methodologies to use OWL and SWRL. We are also actively investigating the possibility of using the D2RQ²³, a language to describe mappings between relational database schema and OWL/RDFS ontologies. With these solutions, our virtual data model remains flexible and independent of the structure of the data sources.

Currently, we use the Protégé-OWL editor to build the Epoch models. Based on the class and property definitions, Protégé automatically generates graphical user interface (GUI) forms that can be used to create instances of these classes (OWL *individuals*). Thus, domain specialists can use to enter a specification of a protocol, say for a transplant clinical trial, using these Protégé-generated forms. Unfortunately, domain specialists find it cumbersome and non-intuitive to use the generic user interfaces as they are exposed to the complexities of the Epoch models, the OWL expressions and SWRL rules. We are building custom graphical user interfaces that hide the complexities of the knowledge models, and that facilitate guided knowledge-acquisition. Providing a friendly user interface to enter SWRL rules can be challenging.

A major concern in building and maintaining a knowledge repository of several clinical trial protocols over long periods of time is the versioning of ontologies. Specifications can change even after the trial has started. Protégé-OWL provides some limited versioning capability. It also provides some tool support for comparing

and merging different versions of ontology. With continued interest in building and maintaining large OWL-based biomedical ontologies²⁴, we can expect improved tool and methodology support. It is not clear if the existing and proposed tools can fully address the issues of ontology changes during the execution of a clinical trial and the resulting complexities in collating and analyzing trial data.

The knowledge representation and reasoning requirements borne out of the need for semantic interoperability in our clinical trial management system align well with the touted strengths of semantic technologies – uniform domain-specific semantics, flexible information models, and inference technology. Using semantic approaches, we will be able to integrate existing software applications and databases with our knowledge based framework with greater transparency and dynamic communication.

Acknowledgements. This work was supported in part by the Immune Tolerance Network, which is funded by the National Institutes of Health under Grant NO1-AI-15416.

References

1. Rotrosen, D., Matthews, J.B., Bluestone, J.A. The Immune Tolerance Network: a New Paradigm for Developing Tolerance-Inducing Therapies. *J Allergy Clinical Immunology*, Jul;110(1):17-23 (2002)
2. ITN: <http://www.immunetolerance.org/>
3. Johnson, P.D., Tu, S. W., Musen, M. A., Purves, I. A Virtual Medical Record for Guideline-Based Decision Support. *AMIA Annual Symposium*, Washington, DC, 294-298 (2001).
4. O'Connor, M.J., Shankar, R.D. Das, A.K. An Ontology-Driven Mediator for Querying Time-Oriented Biomedical Data. *19th IEEE International Symposium on Computer-Based Medical Systems*, Salt Lake City, Utah, 264-269 (2006)
5. OWL Specification: <http://www.w3.org/2004/OWL/>
6. SWRL Specification: <http://www.w3.org/Submission/SWRL/>
7. Protégé: <http://protege.stanford.edu/>
8. Knublauch, H. Fergerson, R.W., Noy, N.F. and Musen, M.A. The Protégé OWL Plugin: An Open Development Environment for Semantic Web applications *Proc Third ISWC (ISWC 2004)*, Hiroshima, Japan, 229-243 (2004)
9. O'Connor, M.J., Knublauch, H., Tu, S.W., Groszof, B., Dean, M., Grosso, W.E., Musen, M.A. Supporting Rule System Interoperability on the Semantic Web with SWRL. *Fourth International Semantic Web Conference (ISWC2005)*, Galway, Ireland, 974-986 (2005)
10. Allen, J.F. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11): 832-843 (1993)
11. JESS: <http://www.jessrules.com/>
12. Cimarron: <http://www.cimsoft.com/>
13. Musen, M.A., Tu, S.W., Das, A.K., Shahar, Y. EON: A component-based approach to automation of protocol-directed therapy. *Journal of the American Medical Informatics Association*, 3(6), 367–388 (1996)
14. Fox, J., Johns, N., Rahmanzadeh, A., Thomson, R. PROforma: A method and language for specifying clinical guidelines and protocols. *Proceedings of Medical Informatics Europe*, Amsterdam (1996)

15. Boxwala, A.A., Peleg, M., Tu, S. W., Ogunyemi, O., Zeng, Q. T., Wang, D., Patel, V. L., Greenes, R. A., Shortliffe, E. H. GLIF3: A Representation Format for Sharable Computer-Interpretable Clinical Practice. *Journal of Biomedical Informatics*, 37(3):147-161 (2004)
16. Sim, I., Olasov, B., and Carini, S. The Trial Bank system: capturing randomized trials for evidence-based medicine. *Proceedings of the AMIA Annual Symposium*, 1076 (2003)
17. Modgil, S., Hammond, P. Decision support tools for clinical trial design. *Artificial Intelligence in Medicine*, 27(2):181-200. (2003)
18. CDISC: <http://www.cdisc.org/standards/>
19. BRIDG: <http://www.bridgproject.org/>
20. HL7: <http://www.hl7.org/>
21. Wang, H., Rector, A., Drummond, N., et al. Frames and OWL Side by Side. 9th International Protégé Conference, Stanford, CA (2006)
22. Crubezy, M., O'Connor, M.J., Buckeridge, D.L., Pincus, Z.S., Musen, M.A. Ontology-Centered Syndromic Surveillance for Bioterrorism. *IEEE Intelligent Systems*, 20(5):26-35 (2005)
23. D2RQ: <http://www.wiwiss.fu-berlin.de/suhl/bizer/d2rq/>
24. CBIO: <http://www.bioontology.org/>