Generating Medical Logic Modules for Clinical Trial Eligibilty

by
Craig G. Parker

A Thesis Proposal Presented to the
Department of Computer Science
Brigham Young University

In Partial Fulfillment of the Requirements
for the Degree Master of Science

1. Introduction

Clinical trials are important to the advancement of medical science. They provide the experimental and statistical basis needed to determine the benefit of diagnostic and therapeutic agents and procedures. As a basic principle of statistics, the more people that can be enrolled in a clinical trial, the greater the confidence we can have in the results of the trial. However it can be difficult to identify a significant number of patients who meet the criteria for participation. This is because trials usually have very specific criteria for age, gender, state of a given disease, number and types of co-existing diseases, family history of diseases, etc.

When the eligibility criteria for a trial are very narrow, it is easier to associate the outcome of the trial with the experimental variables in question because there are fewer confounding factors. However this comes at the expense of the number of patients that are eligible to take part in the trial. A trial with narrow eligibility criteria makes it easier to associate an experimental result with an experimental variable, but at the same time it makes it more difficult to achieve a result that is statistically significant.

1.1 Determining Eligibility

There are many ways to identify patients who are eligible for clinical trials. One commonly used method is for the clinicians who are participating the trial to evaluate each patient they see in their clinic for eligibility. The advantages of this method include: (1) The workflow of the clinician is only minimally disturbed. (2) The clinician generally has an up-to-date picture of the patient's health conditions. (3) For any eligibility criteria that the clinician is unsure about, the patient is present for questioning. The biggest disadvantage of this method is the fact that it only identifies patients who happen to have a clinic visit with a participating clinician during the enrollment phase of the trial.

Another common method for identifying candidates is through advertisements distributed via television, radio, the internet, newspapers or magazines. These advertisements usually present a number of eligibility criteria and a method for contacting someone who can further evaluate their eligibility. The main advantage of this approach is that it can screen a large number of people, including people who would not have normally visited a clinician's office during the enrollment period. One of the obvious drawbacks of this method is the cost of advertising. In addition, the criteria must be presented in a manner understandable by individuals without medical training. This often means that many people who may meet the criteria presented in the advertisement will not be eligible for the trial when evaluated against the detailed and specific trial criteria by a clinician. Additionally this method usually requires a clinician to spend significant time evaluating potential trial enrollees. This is time that must be allocated outside of their normal clinic schedule and may present a significant impact on their practice.

A third method for identifying candidates is to review medical records looking for patients that may meet the eligibility criteria. This method can find individuals who are eligible, but may not have normally visited a clinic during the enrollment phase of the trial. It may also be better at initially screening candidates because details of the patient's medical status are available and the screener usually has clinical training. However searching through medical records can be a laborious task, and the cost of hiring someone with medical training to do this can be significant. In addition, the information available may be out-of-date causing some eligible patients to be missed, and some ineligible patients to be evaluated further. Finally, recent legislation regarding the privacy of medical records may significantly limit the number of people allowed to view a patient's medical record.

1.2 Electronic Medical Records

We could greatly reduce the cost and time needed to search through medical records if they were available in an electronic format and we had the proper tools to automate the process. The feasibility of such an approach is becoming increasingly realistic as more and more patient-specific medical data is being stored in electronic medical records.

Electronic medical records take many forms. They vary greatly in the depth, breadth and format of information they hold. They may be as simple as scanned images of hand-written notes or simple databases to remind patients when they are due for their next pap smear. On the other extreme they may be constructed from very complex information structures and medical vocabularies that attempt to be able to store anything that can be said about a patient's medical state in a structured, machine-understandable fashion. In practice, most of the electronic medical records used by large healthcare organizations are somewhere in between and are slowly evolving toward the latter model. Most of these electronic medical records will represent some data such as medications and lab results in structured instances using a controlled vocabulary, but other data will remain in unstructured or semi-structured text documents or text fields, or will not be collected electronically at all.

1.3 Overview

These problems motivate us to design a system that can transform statements of clinical trial eligibility into a format that is executable against an electronic medical record. Figure 1 below gives an overview, dividing the system into two major processes. The first step takes clinical trial documents, extracts the eligibility criteria, and transforms these into natural language predicates and first order predicate logic in conjunctive normal form. The second step maps the natural language predicates to concepts in an electronic medical record and uses these mappings with the logic generated in Step 1 to create executable modules. Since not all predicates will be mappable to an electronic medical record, the system will also output a questionnaire that covers the eligibility criteria that we cannot automatically determine. In this thesis we only plan to implement Step 2.

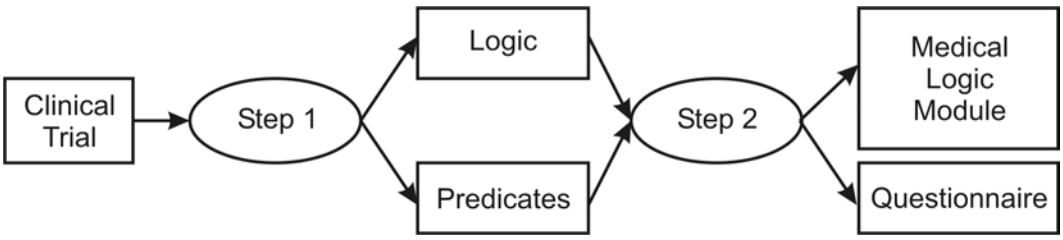

Figure 1 – Overview of transforming clinical trial eligibility into executable modules.

2. Thesis Statement

Assuming that statements of eligibility for clinical trials are already in the form of first order logic over natural language literals, we will generate mappings from the concepts in the predicates to the information model of the target electronic medical record. We will then use these mappings to create medical logic modules to evaluate eligibility. We will also propose a method for handling eligibility criteria that cannot be processed automatically.

3. Methods

Figure 2 shows some representative eligibility criteria from a clinical trial.

Eligibility

Gender: Female

Criteria

Inclusion criteria:

- Pregnant
- Gestational age > 23.0 wks and < 31.6 wks
- Received corticosteroids within the previous 7 days
- . . .

Exclusion criteria:

- Chorioamnionitis
- Non-reassuring fetal testing
- . . .

Figure 2 – Example eligibility criteria.

For this thesis we will assume that the first step in Figure 1 (generating natural language predicates and first-order logic from a trial document) has previously been performed. Figure 3 shows an example of the outputs of this step based on Figure 2.

Predicates:

- (P1) Gender: Female
- (P2) Pregnant
- (P3) Gestational age > 23.0 wks
- (P4) Gestational age < 31.6 wks
- (P5) Received corticosteroids within the previous 7 days
- (P6) Chorioamnionitis
- (P7) Non-reassuring fetal testing
- . . .

Statement of first-order logic:

directly from the trial:

P1 v P2 v (P3 v P4) v P5 v . . . v 5(P6 w P7 w . . .)

and in conjunctive normal form:

P1 v P2 v P3 v P4 v P5 v 5P6 v 5P7 . . .

Figure 3 – Natural language predicates and first order logic.

These predicates and statements of logic become the input for the second step of the process and the focus of this thesis. Based on these inputs we will attempt to create a mapping between the predicates and a target database. We will then create a medical logic module which represents the intent of the eligibility criteria, but is expressed in terms of the target database.

3.1 Mapping Preparation

Before we map the natural language terms in the predicates to concepts in the target database, we will perform a few preparatory operations. These operations include classifying the predicates based on their structure, dealing with units, and dealing with temporal quantities.

3.1.1 Predicate Classification

We will first attempt to classify the natural language predicates. The classifications, based on the structure of the predicate, will assist us in the mapping process. Examples of the classifications include predicates made up of a single noun phrase (e.g. P2 in Figure 3), predicates made up of two noun phrases (e.g. P1), and predicates

made up of a noun phrase and a numeric value (e.g. P3). When attempting to map a predicate, we will use the classification to provide some direction. For example single noun phrases are likely to be diagnoses or clinical observations where predicates made up of two noun phrases are probably name-value pairs.

3.1.2 Units

We will also need to handle the units of any measurements in an appropriate manner. For instance, in Figure 3 gestational age is specified in weeks. In the database gestational age may be represented in days. Each time we find units of measure in a predicate we need to identify the dimension that they measure (e.g. time, area or volume) and be prepared to perform conversions if the same dimension is measured with different units in the target database. We plan to create data frames which will specify unit conversions and use this as a knowledge source in our system [Emb80].

3.1.3 Temporal Quantities

Many of the predicates will contain temporal quantifiers such as, ‘currently,’ or ‘within the last 7 days.’ We will need to recognize and handle these appropriately [Lyo00]. We will use a data frame recognizer [ECJ+99] to identify temporal quantities in predicates. Extracting the time values, we will then attempt to perform any appropriate calculations to determine the real values to be used in our data query.

3.2 Mapping

For the task of mapping concepts in the trial to the target database we will make use of all available tools, building on previous work in schema matching [EXJ01]. Two tools that we use extensively in the mapping process are vocabularies and ontologies. The distinction between these is that vocabularies are a source of terms and their synonyms whereas ontologies give other relationships (e.g. parent-child relationships) between terms.

This project will use two distinct sources of medical vocabularies and ontologies. Primarily we will use the vocabulary and ontology of the target database. This will allow us to make direct mappings against concepts known to be in the electronic medical record. When we are not able to create mappings using only the information that we have about the target database we will employ the vocabularies and ontologies of the Unified Medical Language System (UMLS).

The UMLS [LIND90] is a metathesaurus developed and maintained by the National Library of Medicine. It consists of over 95 medical vocabularies and classifications. Each concept in the UMLS is identified by a Concept Unique Identifier or CUI. A semantic network in the UMLS relates concepts to each other both within a single vocabulary and across vocabularies.

The two major steps in the mapping process are first, to identify possible synonyms for a term that have a representation in the target database, and second, to choose which of the possible mappings is best. We will look for synonyms first in the vocabulary of the target database. If this fails to yield suitable concepts we will next look for synonyms in the UMLS and then see if these synonyms have representations in the target system. Once we have a list of possible mappings we will use the classifications described above with other information that we have available to determine the best match. Examples of the other information that we anticipate using include units of measure, actual values in the database, and ontological relationships. We can use a system of voting to combine the different pieces of information available. To illustrate the mapping process we will look at how we would attempt to map certain predicates in Figure 3 to a target database.

3.2.1 Mapping Example 1

The second predicate in Figure 3 is 'pregnant.' To create a mapping to the target database we would first search the vocabulary of the target database for the term. Figure 4 shows an example of the results that we may obtain from such a query.

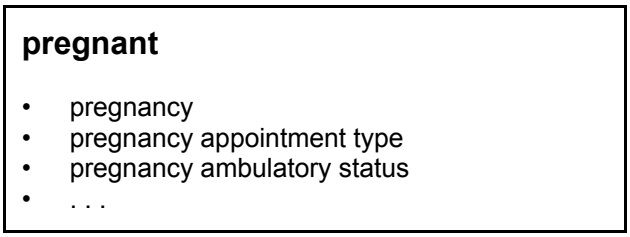

Figure 4 – Result of searching for pregnant in vocabulary of the target database.

The next step is to determine which of the results is the best match for our term. During the initial classification of predicates we would have recognized this single noun phrase as likely being a diagnosis or observation. Using the ontology of the target database we check each result to see if it is a diagnosis or an observation. Figure 5 illustrates the results of this step.

pregnancy
has-parent: Diagnosis
has-parent: Observation
has-parent: Observation Identifier
has-parent: Problem

pregnancy appointment type
has-parent: Appointment type
has-parent: Observation

pregnancy ambulatory status
has-parent: Ambulatory status
has-parent: Observation

Figure 5 – Domain relationships for terms represented by 'pregnant.'

Since pregnancy is a child of both 'Diagnosis' and 'Observation' we select this term for our mapping. Now that we know what we are looking for we need to know where to find it. Again, looking at the relationships in Figure 5 we see that 'pregnancy' is a child of 'Observation Identifier.' This means that we can look in the database for a 'Pregnancy Observation.' Figure 6 shows a simplified example of the structure of this observation from an electronic medical record.

```
PregnancyObservation :Is-Subtype-Of: DiagnosisAndFindingObservation {
    value(codedTerm({Pregnancy, 83035}));
    negation(boolean);
}
```

Figure 6 – Simplified structure of a 'Pregnancy Obseration.'

This result tells us that '83035,' which represents the term 'Pregnancy,' is the value of a 'Pregnancy Observation' and is a type of diagnosis. Therefore, to determine the predicate, 'pregnant,' we would query the database for 'PregnancyObservation' for the patient we are evaluating. If we find '83035' we would make the predicate true.

3.2.2 Mapping Example 2

When searching for the observation corresponding to a particular term we may be presented with more than one possibility. For example, in addition to 'Pregnancy Observation,' our query may have also returned a 'Pregnancy Trimester Observation.' To decide between these we can look at the types of values these observations take. As shown above, 'Pregnancy Observation' takes a value of 'Pregnancy' which is compatible with our original term 'pregnant.' On the other hand 'Pregnancy Trimester

Observation' takes as values 'First,' 'Second,' or 'Third.' Since we have no information in our predicate related to these values we would choose 'Pregnancy Observation' over 'Pregnancy Trimester Observation.'

In a similar manner we could use the units of measure assigned to an observation to make distinctions. Suppose that a trial made reference to an event in the third or fourth month of pregnancy. This would be input into our system as two predicates, 'third month of pregnancy' and 'fourth month of pregnancy' connected by a disjunction. Our database on the other hand may have information about the event of interest with time measured in trimesters. We would recognize that even though the dimension of the measurement is the same (i.e. a time period), the precision of the measurements is different. We could use data frames to convert from between months and trimesters. From this we may not be able to determine the absolute truth of the predicates, but we may be able to determine if it is possible for the predicates to be true. We could then present the user with the information that the system was able to determine and let them handle the situation appropriately.

3.2.3 Mapping Example 3

Consider the first predicate in Figure 3, 'Gender: Female.' Recognizing that two noun phrases are likely to be a name-value pair we can search for such a relationship. Querying the target data dictionary, we find that the term 'Female' is in fact a child of the domain 'Gender,' and we find that 'Gender' is an 'Observation Identifier.' From this information we can evaluate the predicate by searching for a 'Gender Observation' with the value of 'Female.'

3.3 Generating Medical Logic Modules

Once the mappings from the eligibility predicates to the target database are established we will attempt to generate medical logic modules to evaluate eligibility. We will represent our medical logic modules using the Arden Syntax [HCP+90]. The Arden Syntax was developed in 1992 as a language for encoding medical knowledge. It was developed in an attempt to address the need for sharing medical knowledge between hospitals and other medical institutions. It is currently maintained by the HL7 Arden Syntax Special Interest Group and is an ANSI standard. Many vendors of electronic medical records have implemented Arden compilers in their systems.

There are two logical steps in this process of generating medical logic modules. First we need to evaluate each of the predicates to get a boolean value. Then we need to place the boolean values in the input statement of first-order predicate logic to get a final answer. Figure 7 shows and example of the data and logic slots of an Arden Syntax module.

```
KNOWLEDGE:
    TYPE: . . .
    DATA:
        . . .
        Pregnant := READ {select coded_concept from QualitativeObservation
                        where coded_concept = '83035'}
        . . .
    EVOKE: . . .
    LOGIC:
        IF Gender == 'female'
        AND Pregnant IS NOT NULL
        AND Gestational_age >= 23
        AND Gestational_age <= 31.6
        AND . . .
        THEN . . .
    ACTION: . . .
```

Figure 7 – Example of data and logic slots in Arden Syntax.

When generating the Arden Syntax, special care will need to be devoted to the representation of negatives. For example P7 in Figure 3 is an exclusion criteria of non-reassuring fetal testing. If we are able to locate a finding of reassuring fetal testing in the target database we can conclude NOT P7. Since P7 is an exclusion criteria, NOT P7 contributes to the eligibility of the patient. If P7 were an inclusion criteria, then NOT P7 would make the patient ineligible for the trial.

When evaluating the first-order predicate logic we will work under an open-world assumption. The rationale for this is the fact that medical databases are not complete. Therefore if we are not able to find a specific piece of information in the database, we cannot conclude that the event associated with that data did not occur. It is quite possible that the event did occur but was not recorded, or it was recorded in a manner that we cannot retrieve (e.g. as a natural language comment).

Finally, we realize that we will not be able to map all predicates to the target database. This is because trials are often concerned with information that is not normally stored in the medical record. For example, an inclusion criterion may specify that the patient must be willing to travel weekly to the medical center where the trial is being conducted. To manage these situations we will reform the original statement of first-order logic to generate a result for the predicates that we are able to map. The system could then generate a preliminary result based on the mappable predicates, and ask the user for information on the unmapped predicates if the preliminary result was consistent with eligibility.

3.4 Evaluation

We will evaluate our system on approximately 25 trials selected from the National Library of Medicine's 'ClinicalTrials.gov' web site. We anticipate the average number of predicates per trial to be about eight, giving us about 200 predicates to evalute. To evaluate the correctness of our mappings, we will calculate precision and recall values for both the individual terms that we map and for the predicates (functions of our system). We will also measure the percentage of predicates that are mappable (a function of the trial and the target database). This measure will give us an idea of how useful a theoretically perfect system could be. For the generation of Arden Syntax we will evaluate the correctness of the generated logic. We will report this as precision and recall.

4. Contribution to Computer Science

We will develop a method to transform first-order logic predicate expressions into a standard medical knowledge representation language. The results of this project will enable semi-automatic determination of patient eligibility for clinical trials.

5. Delimitations of the Thesis

It is beyond the scope of this thesis to automatically generate the predicates from the natural language source text. Instead, we will perform this step manually.

This project will not result in a production ready application. Instead it will demonstrate that the desired outcome is achievable.

This project will only deal with English language documents and concepts.

This project will only deal with logic that can be addressed with first order predicate calculus.

6. Thesis Outline

I. Introduction (7 pages)
II. Design of System (18 pages)
    A. System Inputs
    B. Mapping Preparations
    C. Generating Mappings
    D. Generating Logic
III. Experiments (8 pages)
    A. Results
    B. Discussion
VI. Conclusion and Future Work (2 pages)

7. Thesis Schedule

A tentative schedule of this thesis is as follows:

Design of Process	December 2002
Creation of Eligibility Predicates	January 2003
Creation of Mapping Preparation Functions	February - March 2003
Creation of Mapping Functions	March - May 2003
Creation of Arden Generator	May - June 2003
Evaluation of Results	July 2003
Thesis Revision and Defense	August 2003

8. Bibliography

[ECJ+99] D.W. Embley, E.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.K. Ng, R.D. Smith. Conceptual-Model-Based Data Extraction From Multiple-Record Web Documents. *Data and Knowledge Engineering*, 31(3):227-251, 1999.

This paper describes the data extraction techniques used in the BYU Ontos system which is based on data frames and ontologies. We will use the these methods to extract temporal quantities from our predicates.

[Emb80] D.W. Embley. Programming with data frames for everyday data items. In The American Federation of Information Processing Societies Proceedings, pages 301-305, 1980.

This paper describes data frames. We will use data frames to perform unit conversions in our system.

[EXJ] D.W. Embley, D. Jackman, and L. Xu. Multifaceted Exploitation of Metadata for Attribute Match Discovery in Information Integration. In *Proceedings of the Interantional Workshop on Information Integration on the Web (WIIW'01)*, pages110-117, Rio de Janeiro, Brazil, April 2001.

This paper describes methods used in schema matching. We will employ some of these methods in our mapping process.

[HCP+90] G. Hripcsak, P.D. Clayton, T.A.Pryor, P. Haug, O.B. Wigertz, J. van der Lei. The Arden Syntax for Medical Logic Modules. In R.A. Miller (ed.), *Proceedings of the Fourteenth Annual Symposium on Computer Applications in Medical Care*, Washington, D.C. New York: IEEE Computer Society Press, 1990; pages 200-204.

This paper describes the Arden Syntax for Medical Logic Modules. We will use the Arden Syntax to encode the logic of our system.

[LIND90] C. Lindberg. The Unified Medical Language System (UMLS) of the National Library of Medicine. In *Journal of the American Medical Record Association*, 61(5):40-42, May 1990.

This paper describes the Unified Medical Language System. We will use the vocabulary and semantic network of the UMLS in our mapping process.

[Lyo00] R.W. Lyon. Identification of Temporal Phrases in Natural Language, Master's Thesis, Brigham Young University, Department of Computer Science, 2000.

This paper describes the identification of temporal phrases. We will use this for finding temporal phrases in our predicates

9. Artifacts

In addition to the written thesis, we will produce an implementation of the software described. This implementation will build on previously developed schema matching and data extraction software as described above. The majority of the project will be written in the Java Programming Language. The target electronic medical record database for this project will be th Central Data Repository at Intermountain Health Care of Salt Lake City, Utah.

10. Signatures

This thesis proposal by Craig G. Parker is accepted in its present form by the Department of Computer Science of Brigham Young University as satisfying the thesis proposal requirement for the degree of Masters of Science.

_____	_____
Date	David W. Embley, Committee Chair
	_____
	Deryl E. Lonsdale, Committee Member
	_____
	William A. Barrett, Committee Member
	_____
	David W. Embley, Graduate Coordinator