ICT-2011-288048

EURECA

Enabling information re-Use by linking clinical Research and CAre

IP
Contract Nr: 288048

Deliverable: 6.1
Formalization of eligibility criteria of CT and a patient recruitment method

Due date of deliverable: (10-31-2012)
Actual submission date: (10-23-2012)

Start date of Project: 01 February 2012 Duration: 42 months

Responsible WP: WP 6

Revision: proposed

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Service	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

0 DOCUMENT INFO

0.1 Author

Author	Company	E-mail
Anca Bucur	Philips	anca.bucur@philips.com
Annette ten Teije	VUA	annette@cs.vu.nl
Frank van Harmelen	VUA	Frank.van.Harmelen@cs.vu.nl
Gaston Tagni	Philips	Gaston.tagni@philips.com
Haridimos Kondylakis	FORTH	kondylak@ics.forth.gr
Jasper van Leeuwen	Philips	Jasper.van.Leeuwen@philips.com
Kristof De Schepper	Custodix	kristof.deschepper@custodix.com
Zhisheng Huang	VUA	z.huang@vu.nl

0.2 Documents history

Document version #	Date	Change
V0.1	1/9/2012	Starting version
V0.2	6/9/2012	z.huang@vu.nl: Add Chapter 4 on rule-based formalization
V0.3	28/9/2012	Added Chapter 5
V0.4	1/10/2012	Added Chapter 6

0.3 Document data

Keywords	Clinical trials, eligibility criteria, recruitment
Editor Address data	Name: J. van Leeuwen Partner: Philips Address: High Tech Campus 34, 5656 AE Eindhoven The Netherlands E-mail: jasper.van.leeuwen@philips.com
Delivery date	10-23-2012

0.4 Distribution list

Date	Issue	E-mailer
10-23-2012		Benoit.ABELOOS@ec.europa.eu

Table of Contents

1 Formalization Approaches: A Brief Overview

In its endeavour to build an advanced, standards-based and scalable semantic interoperability environment enabling seamless, secure and consistent bi-directional linking of clinical research and clinical care systems, EURECA is exploring the formalization of eligibility conditions of Clinical Trials.

Eligibility

Ages Eligible for Study:	*18 Years to 70 Years*
Genders Eligible for Study:	*Female*
Accepts Healthy Volunteers:	*No*

Criteria

Inclusion Criteria:

1. *Histologically-confirmed breast cancer (operable, locally advanced or inflammatory)*
2. *Age less than 70 years*
3. *Female patient*
4. *Tumor size 2 cm at ultrasound examination.*
5. *ER-negative tumors defined according to immunohistochemistry (i.e. < 10% of positive cells after immunostaining).*
6. *Multifocal and multicentric breast tumors are allowed if all foci are ER-negative.*
7. *Fixed and frozen samples from the primary tumor, obtained before treatment with epirubicin, must be available for evaluation of biological markers (topo II alpha gene and protein, HER-2 gene, p-53 gene, oligonucleotides microarrays).*
8. *Written informed consent before study registration.*
9. *Performance status 0 or 1 (ECOG scale)*
10. *Normal CBC, hepatic and renal functions*
11. *Normal left ventricular ejection fraction by echocardiography or muga scan*
12. *Negative pregnancy test for all women of childbearing potential. Patients of childbearing potential must implement adequate non-hormonal measures to avoid pregnancy during treatment.*

Exclusion Criteria:

1. *Metastatic breast cancer*
2. *Serious medical conditions like:*
    1. *congestive heart failure or unstable angina pectoris, previous history of myocardial infarction within 1 year from study entry, uncontrolled arrhythmias.*
    2. *history of significant neurologic or psychiatric disorders*

    3. *active uncontrolled infection*
    4. *active peptic ulcer, unstable diabetes mellitus*
3. *Concomitant contralateral invasive breast cancer*
4. *Concurrent treatment with hormonal replacement therapy*
5. *Concurrent treatment with any other anti-cancer therapy*
6. *Previous treatment with anthracyclines for breast cancer*

Example 1 - Eligibility criteria from the TOP trial

Eligibility criteria typically cover demographics (like age, gender) and clinical parameters (e.g. required laboratory test results) and are constructed in text form (see Example 1).It is quite common that an institute will re-use the (formulation of) eligibility criteria of their existing trials to formulate the eligibility criteria for a new trial. It is also interesting to observe that some criteria are meant to inform the reader and are not discriminating for the patient population. E.g. The degree of spread to regional lymph nodes (a parameter typically named **N**) for breast cancer is expressed with a number (N0, N1, N2, N3). For trials where there is no restriction on **N** there can be a criterion "Any **N**" which will not exclude any patient but makes it explicit to clinicians that all patients are included.

The formalization of eligibility criteria will support the goals of EURECA in various ways. It will allow a more efficient verification of a patient's eligibility for enrolling in a clinical trial. By making the verification of trial eligibility for a patient efficient, it is likely that the number of trials verified for a patient will increase. In addition, it can allow trial managers to "scout" for patients (by matching the data of patients residing in the clinical care systems with the trial eligibility criteria), resulting in an increased enrolment rate for their trials.

Besides trial enrolment, formalization of clinical trial criteria can also aid in designing the enrolment criteria for a new clinical trial. It may allow verifying the feasibility of a trial by assessing the (estimated) enrolment rate of patients (based on the patient population residing in the clinical care systems). It may also allow comparing the newly designed criteria with existing trials.

There are several interesting topics related to the domain of formalization of clinical trial criteria such as how to conveniently construct formalized expressions of criteria de novo, how to come to an expression of eligibility criteria given the (natural) text of the criteria, how to use formalized criteria to assess eligibility given patient data (expressed in a particular data model) and how to use formalisms to compare criteria.

A lot of the work described in this deliverable relies on a joint work with other workpackages. Examples are workpackage 4 –semantic interoperability – in which the data model and semantics will be defined, and workpackage 3 - Information extraction and data access – which has a large natural language processing component.

2 Pattern-based Formalization of Eligibility Criteria

This chapter reports on the preliminary results from experiments conducted in order to extend the set of patterns used for formalizing eligibility criteria in clinical trials that were identified in previous work (K. Milian, 2011).

In order to derive new patterns we used a semi-automatic method that relies on the use of NLP tools to annotate eligibility criteria with concepts from medical terminologies and the analysis of eligibility criteria to identify keywords and their relations. This analysis results in the identification of patterns that capture both the semantics and syntax of several eligibility criteria.

In the following we will describe the steps followed in identifying new patterns and will report on the patterns found with this approach.

2.1 Overview of the Approach

In order to process eligibility criteria to enable applications like patient recruitment a formalization systems should allow for the identification of domain-specific information, in this case biomedical terms and, the identification of specific keywords that denote the actions that need to be carried out with the domain-specific information. In other words, these keywords tell us what exactly needs to be done with the medical terms found in the criteria in order to check whether patients are eligibility or not to a given clinical trial. For example, consider the following eligibility criterion:

“For patients receiving adjuvant therapy”.

In order to check whether a patient satisfies this criterion we need to check whether there is a record in the database that associates the patient to any adjuvant therapy. The relevant concept in this criterion is that denoted by the term *“adjuvant therapy”.* However, that term alone does not give sufficient information about the criterion and about how to evaluate it. What it is needed is the expression *“receiving”* which implicitly denotes a data association between the patient record and an (adjuvant) therapy. Using this keyword we know that what needs to be done in order to evaluate the criterion is to search the database for patients and therapies and their associations. Those database records for which an association exists reveal the patients that satisfy the criterion.

Our approach is a semi-automatic method that uses NLP techniques for parsing eligibility criteria from clinical trials expressed in natural language and extracting relevant information that can be used in the patient selection process. We combine this automatic method with a manual analysis of different eligibility criteria in order to find patterns underlying the criteria and which characterize a set of patients.

2.2 Semantic Annotation of Eligibility Criteria

Eligibility criteria of clinical trials are expressed in natural language. In order for computer programs to be able to process them and capture their meaning, criteria need to be represented in a machine processable form or, in other words, they need to be formalized.

Eligibility criteria contain both domain-specific terminology as well as domain-independent terms. In the context of the EURECA project domain-specific information stems from the medical domain and in particular from the area of cancer research and care. Over the past few years several controlled vocabularies or terminologies have been defined in order to capture different aspects of the medical domain. Some of these terminologies are expressed as ontologies which define concepts, their properties and different types of relationships among them. Relationships are typically arranged in a hierarchy, one of the most commonly used being the is-a relationship.

One way to capture the semantics of eligibility criteria is to associate semantic data to them in the form of annotations, or metadata. In order to do that we used NLP tools to parse the criteria expressed in natural language and identify relevant medical concepts using standard medical vocabularies. In particular we used the SNOMED CT terminology of clinical terms. The result of this is a set of criteria annotated with different medical terms.

In our experiments we use the NLP framework GATE [1], a *general architecture for text processing* which is capable of incorporating different NLP techniques in the form of plugins in order to solve text processing problems. Together with GATE we used MetaMap (A. R. Aronson, 2010), a (GATE-plugin) system developed by the National Library of Medicine (NLM) [2] that identifies biomedical concepts in a corpus and maps them to the UMLS metathesaurus. The MetaMap tagger connects to either a local or remote server in order to retrieve relevant concepts. In our experiments we used a local version of the server and in particular its version 2.

In our experiments we used a (sub) set of all available criteria and focused on those that contain the (temporal) keyword expression *"currently receiving"*; a set of 30 eligibility criteria was used. These criteria are meant to denote patients that are receiving, at the moment of cohort selection, a given substance or that are subjects of a given procedure or therapy such as for instance chemotherapy. This in turn suggests the type of condition that needs to be evaluated in order to check whether a patient satisfies a given criterion or not. In other words, the expression *"currently receiving"* implies that in order for a patient to satisfy the criterion the database must contain a record specifying that the given patient is in fact receiving a given substance or is subject of a certain procedure. The specific procedure or substance is specified in the eligibility criteria along with the temporal expression.

For each eligibility criteria GATE outputs a set of annotations using the results of the MetaMap tagger. Annotations are expressed as an XML document where the original term in natural language is associated with different types of metadata. For example, the term *"breast cancer"* is annotated as follows:

```
<MetaMap gate:gateId="3"
        PreferredName="Breast cancer"
        ConceptId="C0006142"
        SemanticTypes="neop", Type="Mapping"
        Sources="[ICD10CM, MTHICD9, MTH, NCI, CHV, ICPC, MEDLINEPLUS, MSH]"
        ConceptName="breast cancers" Score="-861">
```

[1] http://gate.ac.uk/
[2] http://nlm.nih.gov/

The most important information that we can use from this annotation is the concept identification code (conceptId) which uniquely identifies the concept “breast cancer” and the semantic type. The score represents the similarity between the term being searched for and the ontology concept that most closely refers to it. A value of -1000 denotes a perfect match. This value is computed by the MetaMap tagger.

Using this automatic approach we were able to annotate all the eligibility criteria that contain the temporal expression *“currently receiving”*. However, some important issues were identified:

- Undesired annotations due to term association: Sometimes the annotations produced by the NLP tools were not the desired ones. For example, consider the following case of a criterion including the following expression:

  “...breast cancer therapy”

  Ideally we would like to identify two concepts in this criterion. The first is *“breast cancer”* and the second one is *“therapy”*. The first should be annotated with the SNOMED CT term whose ID is *“254837009”*, while the second should be identified as the SNOMED CT term *“276239002”*. See Figure 1 for an example.

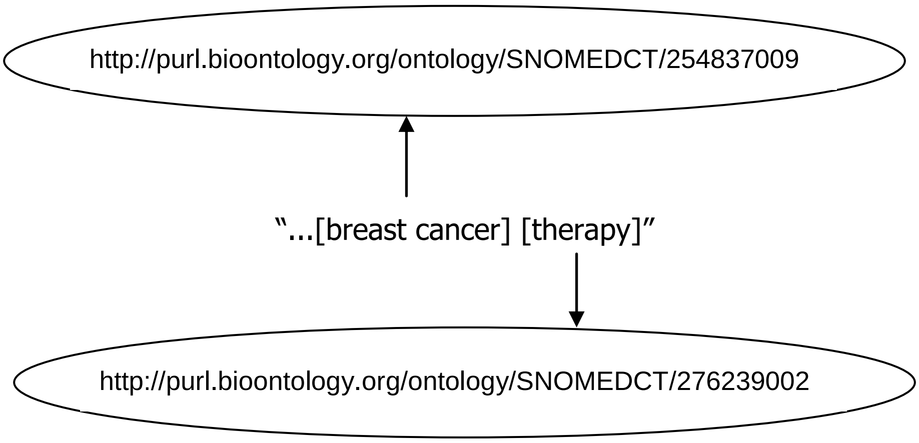

http://purl.bioontology.org/ontology/SNOMEDCT/254837009
“...[breast cancer] [therapy]”
http://purl.bioontology.org/ontology/SNOMEDCT/276239002

Figure 1 Desired annotation

However, MetaMapper produces the following annotation (see Figure 2):

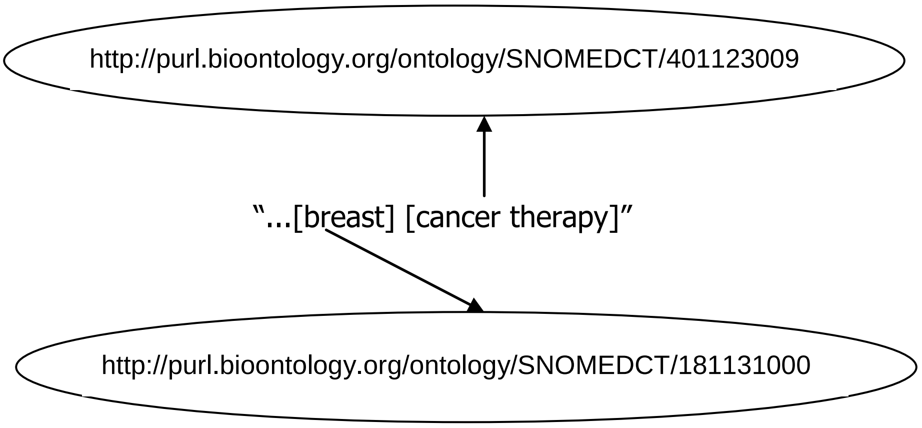

http://purl.bioontology.org/ontology/SNOMEDCT/401123009
"...[breast] [cancer therapy]"
http://purl.bioontology.org/ontology/SNOMEDCT/181131000

Figure 2 Annotation produced by MetaMap for this expression

- Missing annotations when relevant concepts exist: Sometimes MetaMap may not output the desired annotations due to non-existing terms in the ontologies used. For example, when using SNOMED CT as the main ontologythe following eligibility criteria expression is not properly annotated.

  "...currently receiving IV biophosphonates"

  In this case the term IV refers to the medical term intravenous, which denotes a method of substance administration. In this case it refers to a biophosphonates (a type of drug) administered intravenously. Ideally, we should be able to annotate this criterion with two concepts. A first concept to identify the method of administration (IV) which corresponds to the SNOMED CT concept with ID *"255560000"*. The second concept should be the actual drug being given to a patient. However, MetaMap doesn't produce an annotation for the term IV (or for intravenous) thus the criterion is not properly annotated. A solution here would be to add additional parsing to find error/omissions in the annotations.

- Missing annotations due to non-existence SNOMED CT concepts: Another cause for missing annotations is the nonexistence of SNOMED CT concepts. For example, the following criteria *"Patients who are receiving other [investigational drugs]"* cannot be properly annotated because SNOMED CT does not define the concept *"biophosphonates"* . One possible solution is to use other sources for annotation such as the MeSH ontology [3] which does define the concept identified by the ID *http://purl.bioontology.org/ontology/MSH/D015507*. Therefore, a solution could be to annotate the criterion with concepts from other ontologies. However, this would depend on the specific medical vocabularies used in the project.

[3] http://www.nlm.nih.gov/mesh/MBrowser.html

2.3 Identifying Patterns

Once the eligibility criteria are annotated with biomedical terms defined in SNOMED CT the next step consists in identifying the relevant patterns that will capture the full semantics and syntax of the eligibility criteria expressions.

Each pattern identifies the relevant keywords that will determine how to use the medical terms present in the criteria and the semantic type of the medical concepts found in each criterion. Patterns are expressed using regular expressions. For example, from the following two criteria:

1. *“patients who are receiving other investigational drugs”*
2. *“patients currently receiving oral bisphosphonates”*

We can identify the temporal keyword *“receiving”* and the type of concept that plays the role of the subject of such keyword, in other word, the substance or element a patient is receiving. We do that by analysing the semantic type of the concept associated (by annotation) with the terms *“investigational drug”* and *“biophosphonates”*. In this example the semantic type is *“Pharmacological Substance”*. The word *“oral”* in this case refers to a method of administration which is typically associated with substances such as drugs. Another instance of a method of administration is intravenous or IV. From this analysis the pattern that can be derived is as follows:

currently receiving (*modifier*)? (B)? (A)

where:

- A is an argument of (semantic) type *Pharmacological Substance*.
- B is an optional argument that refers to a *Method of Administration*.
- *Modifier* is an optional argument that refers to terms such as *any* or *other*.

Given an eligibility criterion with the keyword "*currently having”* it is possible to see whether it is an instance of the pattern above by analysing the semantic type of the concept identified by the tagging method. If the argument A and B are instances of a *Pharmacological Substance* and a *Method of Administration* respectively then the criterion contains the pattern described above.

A special case of the pattern introduced above is the one that characterize patients currently receiving therapeutic doses of a given substance, eg. a drug. Such pattern can be expressed by the following regular expression:

currently receiving (*therapeutic doses of*)? (A)

where:

- A is an optional term whose semantic type is *Pharmacological Substance*.

An example of an eligibility criterion matching such pattern is the the statement *“currently receiving therapeutic doses of warfarin (Coumadin)”*.

Another related pattern is the following:

receiving (A | B)+ ***or other similar medications***

where:

- The semantic type of A should be *Organic Chemical*
- The semantic type of B should be *Pharmacological Substance*

An example of an eligibility criteria matching this pattern is the expression:

"Patients currently receiving Neurontin (gabapentin), glutamine supplements, Elavil (amitriptyline), Dilantin, Tegretol, tricyclic antidepressants or other similar medications".

Another related pattern is the following:

receiving (A)? (B)

where:

- The semantic type of the optional argument A should be *Functional Concept*.
- The semantic type of argument B should be *Therapeutic or Preventive Procedure*.

An example of an eligibility criteria matching this pattern are the expressions *"For patients receiving adjuvant therapy"* and *"For patients receiving neo-adjuvant therapy"*.

2.4 Future work

The next step in our work is to validate the new set of patterns with respect to the large corpus of eligibility criteria in order to measure the success rate and to make the appropriate changes. Also, we plan to continue our work with the goal of deriving additional patterns for those eligibility criteria not covered by the current set.

3 Rule-based Formalization of Eligibility Criteria

3.1 Introduction

One of the main goals for the formalizations of eligibility criteria is to provide the functionalities of automatic identification of patient for clinical trials and automatic identification of clinical trials for patients. That requires the implementation of the advanced reasoning services for matching patient data with formalized eligibility criteria. This matching also requires the semantic interoperability between structured patient data (i.e.,EHRs and CMRs) and CT systems.

Rule-based formalization of eligibility criteria in clinical trials is an approach for automatic patient recruitments and trial identification. It is expected to be efficient and effective services of automatic patient recruitment, because of the following features of rule-based formalization.

- **Declaration**. Rule-based formalization is a declarative language that expresses the logic of a computation without the need of exactly describing its control flow. That is significantly different from the traditional programming languages, like Java and many others, which use a procedural approach for the specification of control flow in the computation. A declarative approach of formalization is more suitable for knowledge representation and reasoning because it needs no carefully design its computation (or reasoning) procedure. Thus, a rule-based formalization of eligibility criteria would provide a more convenient and efficient way for the automatic patient recruitments in clinical trials, compared with other procedural approaches, like script-based formalization, and pattern-based approaches, like SPARQL querying.

- **Easy Maintenance**. Rule-based formalization provides an approach in which specified knowledge is easy to be understood for human users, because they are very close to human knowledge. It would not be too hard for human users to check the correctness of the specification of eligibility criteria if they are formalized as a set of rules. Furthermore, changing or revising a single rule would not make an effect on other part of the formalization significantly, because the meaning of the specification is usually represented in the specific rule. Thus, it is much easier for maintenance of knowledge, compared with procedural/scripting approaches of the formalization of eligibility criteria.

- **Reusability**. In a rule-based formalization, a single rule (or a set of rules) is usually considered to be independent from other part of knowledge. Thus, it is much more convenient to re-use some rules of a specification of eligibility criteria of a clinical trial in the specification of another clinical trial, compared with those formalizations which use SPARQL queries with regular expressions. Furthermore, some rules which specify common knowledge, like those rules for temporal reasoning, and domain knowledge, like those involved in knowledge of targeted disease, can be designed to be a common library, which can be re-used for the specification of other trials.

- **Expressiveness**. Automatic patient recruitment usually involves comprehensive scenarios of deliberation and decision-making procedures. To facilitate those capabilities, it may require sophisticated data processing in workflows. An expressive rule-based language can support various functionalities of data processing. Thus, it provides the possibility to build workflows for various scenarios of medical applications.

There exist various rule languages which can be used for the formalization of eligibility criteria. In the researches of artificial intelligence, logic programming languages, like Prolog, are well known and popular rule-based languages. Several rule-based languages, like SWRL[4] and RIF[5] have been proposed for the semantics-enable rule-based language. In biomedical domain, the Arden syntax[6] has been developed to formalize rule-like medical knowledge. However, compared with logic programming language Prolog, both SWRL, RIF, and the Arden syntax have very limited functionalities for data processing.

In this chapter, we will propose a rule-based formalization, which is based on the logic programming language Prolog. More exactly, that rule-based formalization is developed based on the SWI-Prolog[7]. The reasons why we select the SWI-Prolog as the basic language for the rule-based formalization of eligibility criteria, because of the following features of SWI-Prolog[SWIProlog2008,SWIProlog2012]:

- **Semantic Web Support**. SWI-Prolog has been facilitated with powerful libraries for semantic data processing and services. It provides a basic tool for the communication with SPARQL endpoints and other REST based web servers. Furthermore, SWI-Prolog also supports the basic reasoning and storage of semantic data. Thus, the SWI-Prolog has the advantage of the support for semantic data processing, compared with the domain-specific Arden Syntax. Moreover, we can show that the formalization of SWI-Prolog would subsume the functionalities of the Arden Syntax. Namely, any formalization of the Arden Syntax can be automatically converted into the formalization of SWI-Prolog.

- **Powerful Processing Facilities**. SWI-Prolog is a logic programming language. It provides various libraries for data processing, which includes not only the tools for text processing and database-like storage management, but also workflow processing and distributed/parallel processing. Thus, it has significant advantage for the existing rule-based language, which have been proposed in the community of the Semantic Web. Furthermore, we can show that it is possible to convert all the formalization of rule languages like SWRL and RIF into the Prolog-based formalization.

3.2 Rule-based Language Prolog

Prolog is a general purpose logic programming language associated with artificial intelligence, in particular, for knowledge representation and reasoning. Prolog is developed based on the first-order logic. Prolog is a rule-based language, thus, a

[4] http://www.w3.org/Submission/SWRL/
[5] http://www.w3.org/TR/rif-overview/
[6] http://www.hl7.org/special/Committees/arden/index.cfm
[7] http://www.swi-prolog.org/

declarative language for programming. In Prolog, the program logic is expressed in terms of relations, represented as facts and rules. A computation is initiated by running a query over these relations.

SWI-Prolog is an open source implementation of the programming language Prolog, commonly used for teaching and semantic web applications. It has a rich set of features, libraries for constraint logic programming, multithreading, unit testing, GUI, interfacing to Java, ODBC and others, literate programming, a web server, SGML, RDF, RDFS, developer tools (including an IDE with a GUI debugger and GUI profiler), and extensive documentation.

SWI-Prolog can run on various platform, like Windows, Macintosh, Unix/Linux platforms. SWI-Prolog has been developed by a team in the University of Amsterdam, the Netherlands. The main developers of SWI-Prolog are now members in a team at Vrije University Amsterdam.

In Prolog, program logic is expressed in terms of relations. More exactly, those relations are formalized as a set of the predicates, like those in the first order logic. A computation is initiated by running a query over these relations. Relations and queries are constructed using terms and a predicate.

Data types of Prolog

Prolog's single data type is the term. Terms are either atoms, numbers, variables or compound terms.

- An atom is a general-purpose name. Examples of atoms include frank, ‘frank’, and 'Frank van Harmelen'.
- Numbers can be floats or integers, like 3.14159 and 3.
- Variables are denoted by a string which begins with an upper-case letter or underscore and consisting of letters, numbers and underscore characters, like these: X, Y, PatientID, and _anything. The variables which begins with an undercore means that they are placeholders for arbitrary terms.
- A compound term is composed of a function name and a number of arguments. which are again terms which corresponds with a comma-separated list of argument terms and is contained in parentheses, like these: father(“Mary’). A List is considered to be a special case of compound terms, i.e., an ordered collection of terms. It is represented as square brackets with the terms separated by commas or in the case of the empty list, []. For example [1,2,3,4,5] or [banana,apple,orange]. A string is a sequence of characters surrounded by quotes is equivalent to a list of character codes.

Rules and facts of Prolog

As we have discussed above, Prolog programs describe relations. In Prolog, the relations are represented as an atom (or alternatively called atomic formula) which consists of a predicate with several terms as its parameters, like this:

```
age_between(PatientData, AgeMin, AgeMax)
```

where age_between is a predicate and PatientData, AgeMin, and AgeMax are terms.

There are two types of clauses: facts and rules. A rule has the following form

```
Head :- Body.
```

Where Head is an atomic formula, and Body is a list of atomic formulas which are separated with commas and ends with a dot '.', like this:

```
triple_negative(Patient):-er_negative(Patient),
                          pr_negative(Patient),
                          her2_negative(Patient).
```

Which means that a patient is triple negative if it is ER negative, PR negative, and HER2 negative.

3.3 Rule-based Formalization of Eligibility Criteria by Using Prolog

3.3.1 General Consideration

As we have discussed it above, one of the distinguished features of rule-based formalization is the reusability, for which we can build various rule libraries which are reusable in different context and application scenarios.

Thus, we can formalize the knowledge rules of the specification of eligibility criteria of clinical trials with respect to the following different libraries, which are re-usable with different levels of knowledge:

1) Trial-specific Knowledge
   Trial-specific knowledge are those rules which specify the concrete details of the eligibility criteria of a specific clinical trial. Those criteria are different from a trial to another trial. Namely, they are trial dependent. They are not expected to be re-usable in other trials.

2) Domain-specific Knowledge
   Those trial-specific rules above may involve some knowledge which are domain relevant, i.e., the domain knowledge, which are trial independent. We formalize those part of knowledge which are relevant with domain knowledge in the libraries of domain-specific knowledge. For example, for clinical trials of breast cancer, we formalize the knowledge of breast cancer in the knowledge bases of breast cancer, a domain-specific library of rules.

3) Common Knowledge
   The specification of the eligibility criteria may involve some knowledge which are domain independent, like those knowledge for temporal reasoning and the knowledge for manipulating semantic data and interacting with data servers,

e.g. how to obtain the data from SPARQL endpoints. We formalize those knowledge in several rule libraries, which can be reusable for different applications.

3.3.2 Formalization of Trial-specific knowledge

In the specification of the eligibility criteria of clinical trials, we usually formalize their inclusive criteria and exclusive criteria respectively.

Given a patient ID, we suppose that we can obtain its patient data through the common knowledge of the interface with SPARQL endpoints and its internal data storage. Thus, in order to check if a patient meets an inclusive criterion, we can check if its patient data meet the criterion. Furthermore, we would not expect to check all the criteria with respect to the patient data, because some of those required data may be missing in the patient data. We introduce a special predicate getNotYetCheckedItems to collect those criteria which have not yet been formalized for the trial.

Take the example of the trial NCT00002720, the eligibility criteria are:

DISEASE CHARACTERISTICS:

- Histologically proven stage I, invasive breast cancer
- Hormone receptor status:
  - Estrogen receptor positive
  - Progesterone receptor positive or negative

PATIENT CHARACTERISTICS:

Age:

- 65 to 80

Sex:

- Female

Menopausal status:

- Postmenopausal

Performance status:

- Not specified

Life Expectancy:

- Not specified

Hematopoietic:

- Not specified

Hepatic:

- Not specified

Renal:

- Not specified

Other:

- No serious disease that would preclude surgery

- No other prior or concurrent malignancy except basal cell carcinoma or carcinoma in situ of the cervix

PRIOR CONCURRENT THERAPY:

==

The inclusive criteria in the trial NCT00002720 above can be formalized in the following:

```
meetInclusionCriteria(_PatientID, PatientData, CT, NotYetCheckedItems):-
            CT = 'nct00002720',
            breast_cancer_stage(PatientData, '1'),
            invasive_breast_cancer(PatientData),
            er_positive(PatientData),
            known_pr_status(PatientData),
            age_between(PatientData, 65, 80),
            postmenopausal(PatientData),
            getNotYetCheckedItems(CT, NotYetCheckedItems),
            true.
```

Which formalizes that

(i) the stage of the breast cancer must be “stage 1”, i.e., breast_cancer_stage(PatientData, '1'),

(ii) the breast cancer of the patient must be “invasive”, i.e., invasive_breast_cancer(PatientData),

(iii) the hormone receptor status must be estrogen receptor positive, i.e., er_positive(PatientData),

(iv) the hormone receptor status must be progesterone receptor positive or negative, namely, known progesterone receptor status, i.e., known_pr_status(PatientData),

(v) the age of the patient should be that between 65 and 80, i.e., age_between(PatientData, 65, 80),

(vi) the menopausal status of the patient must be “postmenopausal”, i.e., postmenopausal(PatientData).

Note that the knowledge to check whether those facts are true or not for a patient are trial independent and domain-specific. Thus, we will formalize that knowledge in the domain-specific rule libraries, which will be discussed in the next section.

There are no exclusive criteria for the trial ‘NCT00002720’. Thus, we formalize it as follow:

```
meetExclusionCriteria(_PatientID, _PatientData, CT):-
                CT='nct00002720',
                false.
```

```
getNotYetCheckedExclusionItems(CT, NotYetCheckedItems):-
                    CT='nct00002720',
                    Item1 = '',
                     NotYetCheckedItems=[Item1].
```

We formalize the criteria which have not been checked in the following rule.

```
getNotYetCheckedItems(CT, NotYetCheckedItems):-
                CT='nct00002720',
                Item1 = 'No serious disease that would preclude surgery',
                Item2 = 'No other prior or concurrent malignancy except basal cell
carcinoma or carcinoma in situ of the cervix',
                NotYetCheckedItems = [Item1, Item2].
```

3.3.3 Formalization of Domain-specific Knowledge

We consider patient data as a set of property-value pairs. For example, we use the following pair to denote that the birth year of a patient is 1957:

```
birthyear:1957
```

In this rule-based formalization of eligibility criteria of breast cancer, a general format of patient data is designed to be a list of property-value pairs, like this

```
[disease:'BreastCancer',
birthyear:BirthYear,
menopause:Menopause,
currentlyPregnant:Pregnant,
currentlyNursing:Nursing,
diagnosis:Diagnosis,
diagnosisyear:DiagnosisYear,
diagnosismonth:DiagnosisMonth,
her2:HER2,
er:ER,
pr:PR,
stage:Stage,
tumorsize:TumorSize,
lymphnodes:Lymphnode,
metastases:Metastases,
…
]
```

This general format of patient data is flexible for different formats of clinical medical records (CMRs), because we can design a CMR-specific interface to obtain the corresponding data via different data servers, which can be a SPARQL endpoint, internal data storage server, or a database server.

We introduce the predicate getItem(PatientData+, Property+, Value-)[8] to get the value of the property from the patient data. Thus, the predicate getItem(PatientData+, Property+, Value-) can be formalized as a set of the Prolog rules as follows:

```
getItem([], _Item, nil).

getItem([Item:Value|_L], Item, Value).

getItem([_H|L], Item, Value):-
    getItem(L, Item, Value).
```

Several receptor status of breast cancer cells have been considered to be very important for the classification of breast cancer. Those important receptors are estrogen receptor (ER), progesterone receptor (PR), and Human Epidermal growth factor Receptor 2 (HER2). Thus, the receptor status is formalized as follows:

```
er_positive(PatientData):-
          getItem(PatientData, er, ER),
          ER = 'positive'.

er_negative(PatientData):-
          getItem(PatientData, er, ER),
          ER = 'negative'.

her2_positive(PatientData):-
          getItem(PatientData, her2, HER2),
          HER2 = 'positive'.

her2_negative(PatientData):-
          getItem(PatientData, her2, HER2),
          HER2 = 'negative'.


pr_positive(PatientData):-
          getItem(PatientData, pr, PR),
          PR = 'positive'.
```

[8] A parameter ending with the sign "+" means an input parameter. A parameter ending with the sign "-" means an output parameter. A parameter ending with the sign "?" means both an input parameter and an output parameter.

```
pr_negative(PatientData):-
                getItem(PatientData, pr, PR),
                PR = 'negative'.
```

We can formalize the triple-negative breast cancer as follows:

```
triple_negative(PatientData):-
                er_negative(PatientData),
                pr_negative(PatientData),
                her2_negative(PatientData).
```

Known receptor status means that either the receptor status is positive or negative. Thus, they can be formalized as follows:

```
known_er_status(PatientData):-
                er_positive(PatientData).

known_er_status(PatientData):-
               er_negative(PatientData).

known_pr_status(PatientData):-
                pr_positive(PatientData).


known_pr_status(PatientData):-
               pr_negative(PatientData).



known_her2_status(PatientData):-
                her2_positive(PatientData).

known_her2_status(PatientData):-
               her2_negative(PatientData).
```

Furthermore, more domain knowledge of breast cancer can be formalized as follows:

```
age_between(PatientData, AgeMin, AgeMax):-
                getItem(PatientData, birthyear, BirthYear),
                atom_number(BirthYear, BirthYearN),
                thisyear(ThisYear),
                Age  is ThisYear - BirthYearN,
                Age >= AgeMin,
                Age =< AgeMax.
```

```
breast_cancer_stage(PatientData, Stage):-
            getItem(PatientData, disease, 'BreastCancer'),
             getItem(PatientData, stage, Stage).


measurableTumor(PatientData):-
             getItem(PatientData, tumorsize, TumorSize),
             atom_number(TumorSize, TumorSizeN),
             TumorSizeN >2.



invasive_breast_cancer(PatientData):-
            getItem(PatientData, diagnosis, Diagnosis),
            (Diagnosis = 'Invasive ductal carcinoma'; Diagnosis = 'Invasive lobular
carcinoma'),
             true.

postmenopausal(PatientData):-
             getItem(PatientData, menopause, Menopause),
             Menopause = 'postmenopausal'.

premenopausal(PatientData):-
             getItem(PatientData, menopause, Menopause),
             Menopause = 'premenopausal'.

perimenopausal(PatientData):-
             getItem(PatientData, menopause, Menopause),
             Menopause = 'perimenopausal'.
```

The menopausal statue of a female patient is simply considered as a value of a property in the patient data. Actually in medical science, menopausal status is defined in terms of menstrual periods.

For example, Postmenopausal is usually defined with more details as:
- At least 1 year since last menstrual period
- At least 2 months since bilateral oophorectomy prior to breast cancer diagnosis
- 4-12 months since last menstrual period and FSH elevated to postmenopausal range
- Postmenopausal estrogen therapy and 55 years of age or older

The formalization of the knowledge above requires the rules which involve with some temporal operators, like the predicate last_time, the predicate today, and the predicate at_least_earlier in the following:

```
last_time(Patient, menstrual_period, LastMenstrucalPeriod):-
       hasPatientData(Patient,PatientData),
       postmenopausal(PatientData),
       today(Today),
       at_least_earlier(LastMenstrucalPeriod, Today, 1, year).
```

3.3.4 Formalising common knowledge

3.3.4.1 Temporal Reasoning

The rules for formalizing temporal reasoning and others are not domain-specific, because that kind of knowledge can be used in different applications. Thus, they are designed to be separated libraries, which are different from the domain specific libraries.

In reasoning of breast cancer knowledge, we may need various temporal operators /predicates, like the following:

- today(Today), which returns the date of today. It can be defined in terms of Prolog build-in predicates as follows:

```
today(Today):-
   get_time(Stamp),
   stamp_date_time(Stamp, D, 0),
   date_time_value(date, D, Today).
```

- thisyear(ThisYear), which returns the year of today.

```
thisyear(ThisYear):-
    today(Today),
    Today = date(ThisYear, _Month, _Date).
```

- Thismonth(ThisMonth): which returns the month of today. It can be formalized as follows:

```
thismonth(ThisMonth):-
    today(Today),
    Today = date(_ThisYear, ThisMonth, _Date).
```

Similarly, we can formalize other temporal operators like the last/next-month/year/week, like this:

```
lastmonth(LastMonth):-
    today(Today),
    Today = date(_Year, ThisMonth, _Date),
    ThisMonth > 1,
    LastMonth  is ThisMonth - 1.

lastmonth(LastMonth):-
    today(Today),
    Today = date(_Year, ThisMonth, _Date),
    ThisMonth is 1,
    LastMonth  is 12.

lastmonth_with_year(LastMonth, Year):-
```

```
    today(Today),
    Today = date(Year, ThisMonth, _Date),
    ThisMonth > 1,
    LastMonth  is ThisMonth - 1.

lastmonth_with_year(LastMonth, LastYear):-
    today(Today),
    Today = date(ThisYear, ThisMonth, _Date),
    ThisMonth is 1,
    LastMonth  is 12,
    LastYear is ThisYear - 1.

nextmonth(NextMonth):-
    today(Today),
    Today = date(_Year, ThisMonth, _Date),
    ThisMonth < 12,
    NextMonth  is ThisMonth + 1.

nextmonth(NextMonth):-
    today(Today),
    Today = date(_Year, ThisMonth, _Date),
    ThisMonth is 12,
    NextMonth  is 1.

nextmonth_with_year(NextMonth, ThisYear):-
    today(Today),
    Today = date(ThisYear, ThisMonth, _Date),
    ThisMonth < 12,
    NextMonth  is ThisMonth + 1.

nextmonth_with_year(NextMonth, NextYear):-
    today(Today),
    Today = date(ThisYear, ThisMonth, _Date),
    ThisMonth is 12,
    NextMonth is 1,
    NextYear is ThisYear + 1.

lastyear(LastYear):-
    today(Today),
    Today = date(Year, _Month, _Date),
    LastYear  is Year - 1.

nextyear(NextYear):-
    today(Today),
    Today = date(Year, _Month, _Date),
    NextYear is Year + 1.
```

To simplifying the calculate the day difference between two dates, we consider the average day of a month is 30. Thus, it can be simply formalized as follows:

```
daydiff(Date1,Date2, Days) :-
    Date1 = date(Year1,Month1, Day1),
    Date2 = date(Year2,Month2, Day2),
    Days is Day1-Day2 + 30*(Month1-Month2 + 12*(Year1-Year2)).
```

Thus, we can formalize the temporal operator earlier with respect to day, and the temporal operator, earlier with respect to month, and earlier with respect to year as follows:

```
earlier(Date1,Date2, N, day):-
    daydiff(Date2,Date1, N).

earlier(Date1,Date2, N, month):-
    daydiff(Date2,Date1, Days),
     N > Days/30.

earlier(Date1,Date2, N, year):-
    daydiff(Date2,Date1, Days),
     N > Days/360.


at_least_earlier(Date1, Date2, N, year):-
   daydiff(Date1,Date2,Days),
   N > Days/360.

at_least_earlier(Date1, Date2, N, month):-
   daydiff(Date1,Date2,Days),
   N > Days/30.

at_least_earlier(Date1, Date2, N, day):-
   daydiff(Date1,Date2,Days),
   N >= Days.

at_least_after(Date1, Date2, N, Unit):-
    at_least_earlier(Date2,Date1, N, Unit).
```

3.3.4.2 Semantic Interoperability

As we have discussed above, semantic interoperability has been considered to be a basic requirement for the formulation of eligibility criteria of clinical trials. Semantic interoperability of the formalisms allow for shared meaning of the knowledge representation of clinical trials. Semantic interoperability is a requirement to enable machine computable logic, inferencing, knowledge discovery, and data federation between information systems. Semantic interoperability is achieved by embedding medical terminologies and domain ontologies in the formulization of clinical trials and patient data, like clinical medical records (CMRs) and personal health records (PHRs).

SWI-Prolog provides a powerful library of the semantic web, by which we can achieve the semantic interoperability in the rule-based formulation of eligibility criteria efficiently and effectively. SWI-Prolog handles the semantic web RDF model and OWL data naturally. RDF and OWL provides stable models for knowledge representation with nice support for semantic interoperability.

The rule-based formulation of eligibility criteria of clinical trials is developed with the support of the following two libraries of the semantic web in SWI-Prolog:

i) Web-server and client library

This is the core web package of the semantic web in SWI-Prolog. It provides an HTTP server and client, session handling, authorization, logging, etc. Libraries for generating HTML pages and JSON. Based on this library, we can develop the interface with SPARQL endpoints to obtain semantic data (e.g. semantics-enable patient data and medical ontologies) for the rule-based formulation of eligibility criteria.

For example, the following rule in Prolog is designed to obtain the patient data for a SPARQL endpoint, which is located at the localhost with the port '8183':

```
getPatientData(PatientID, PatientData):-
                  get_sparql_query(patientdata, Query, PatientID),
                  findall(Row, (sparql_query(Query,
            Row,[host('localhost'), port(8183), path('/sparql/')])),
            Answers),
                 sparql_answer_to_list(patientdata(PatientID),
Answers, PatientData).
```

Namely, given a patienID, the predicate 'getPatientData' would return the patient data from the corresponding SPARQL endpoint. We use the predicate get_sparql_query(patient, Query, PatientID) to get a system specific SPARQL query for the given patient ID. We use the build-in predicate sparql_query to obtain the result Answers from the SPARQL endpoint. We design a predicate sparql_answer_to_list to convert the answers from the SPARQL endpoint into the internal representation of the patient data (i.e., a Prolog list), so that the patient data can be processed further by the predicate getItem, as we have discussed in the section about the formalization of domain specific knowledge.

ii) RDF storage and query library

SWI-Prolog provides a convenient and powerful support for the storage and manipulation of semantic data, like loading and saving RDF data and querying them. This RDF library loads and saves XML/RDF and Turtle. In the latest version of SWI-prolog, Its database scales to approx. 20M triples on 32-bit hardware and 300M triples on a 64-bit machine with 64-Gb memory. It also provides simple RDFS and OWL support. That would be sufficient for the temporal and internal storage of the semantic data in the rule-based formulation of eligibility criteria.

3.3.5 Discussion

For clinical trials, identifying eligible patients are mostly manually. Thus, it often results in low clinical trial enrolment. The formulation of eligibility criteria provides the possibility for automatic identification of patients for clinical trials. Based on the rule-based formulation of eligibility criteria, we expect to achieve a more efficient way in the EURECA services for automatic identification of eligible patients whenever possible.

With the support of the semantic web library in the Prolog-based formalism, we can can achieve the semantic interoperability among EHR and clinical trial systems, because the relevant can be exploited to allow more efficient patient enrolment in clinical trials. The eligibility and exclusion criteria of running clinical trials can be used for matching checking for this automatic patient enrolment. Semantic interoperability between EHR and CT systems enables us to provide solutions for patient recruitment that help avoid double data entry: establishing a single source for each data item, automatic storing of clinical trial eligibility criteria into the EHR and using the EHR data for automatic Electronic Data Capture (EDC).

However, automatic patient recruitment for clinical trials would not be considered as a simple system of automatic checking with the criteria of patient data. In many application scenarios of patient recruitment, it should be considered to be one with a decision making system, which involves complex procedure and comprehensive processing over various data and workflows. Furthermore, it would be also beneficial if the formalism can accommodate and integrate with the clinical guidelines for specific diseases [de Clercq et al. 2004]. That requires that rule-based formulation of eligibility criteria can be extended with the support of workflow-enable formalisms. We will leave the integration of rule-based formulation of eligibility criteria with workflow-enable formalism as one of the future work.

4 Patient recruitment based on ontology matching

CTs are based on statistical tests and population sampling and because they rely on adequate sample sizes it is common for CTs to fail in their objectives because of the difficulty of meeting the necessary recruitment targets in an effective time and a reasonable cost.

In this chapter we will describe a research prototype created in order to match patient records and clinical trials using semantic technologies. The proposed solution hopefully will reduce the necessary recruitment time and the corresponding cost. The approach, presented in Figure 3, receives as input an Ontology describing the clinical domain (i.e. Clinical Ontology) and patient instances, and an Ontology describing Clinical Trials (i.e. Clinical Trial Ontology) with the appropriate instances as well. The output is a list of matchings between clinical trials and patients.

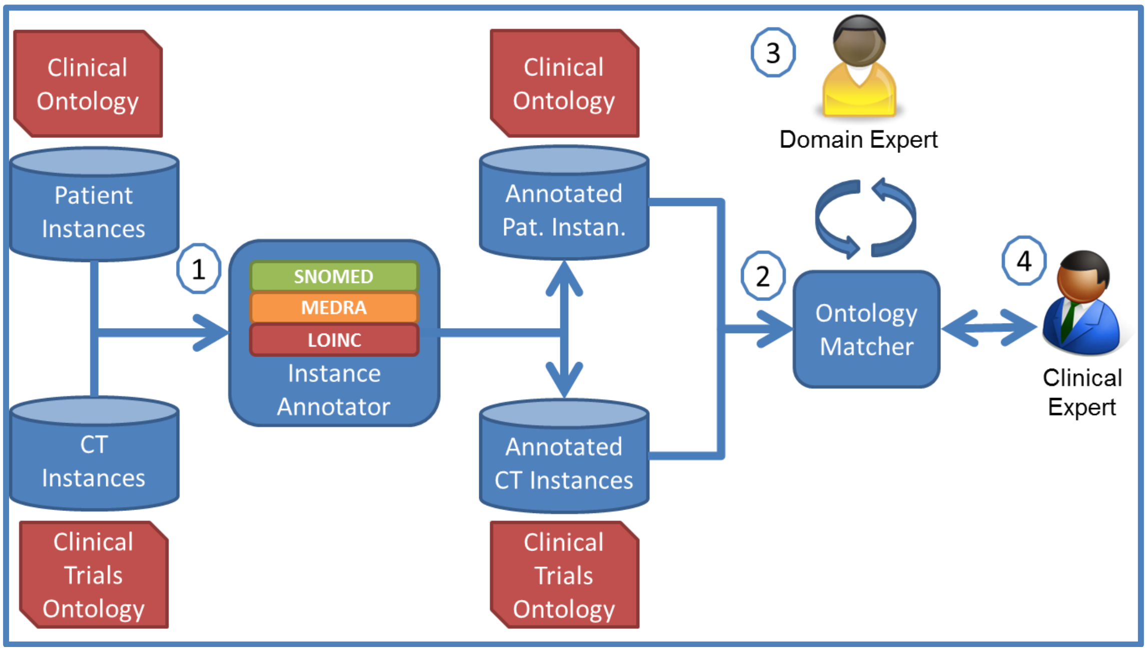


Figure 3. Matching Patients & Clinical trials using semantic technologies

The necessary steps in our approach are the following:

1. Annotate Patient & Clinical Trial instances
2. Match Annotated instances using an ontology matcher
3. Review and refine identified matchings by IT Expert
4. Verify identified matchings by Clinical Expert

The requirements of this approach as long as the aforementioned steps are described in the following sections.

4.1 System set-up

Our approach requires two different ontologies with the corresponding instances. There are several ontologies that are used for modelling clinical domain such as the Ontology of Clinical Research[9], the ACGT ontology [1] etc. However, besides, having

[9] http://rctbank.ucsf.edu/home/ocre

the ontologies, instances are required as well. Those instances can be obtained in two ways:

1. Using Extract-Transform-Load (ETL) tools: Using such tools the instances of the local databases can be extracted, transformed to the appropriate format and then loaded to the ontology file.
2. Using Virtual Instances: Instead of extracting the instances from the local databases, mappings can be provided to link the ontologies to the source data. Our matcher is capable of using those mappings to retrieve the mapped instances.

Independent of the method selected, the fundamental hypothesis we make is that linking terms from patients' records and eligibility criteria to a formal ontology both minimizes the risk of ambiguity and allows automated reasoning.

4.2 Annotate Patient Records & Clinical Trials

Besides having the appropriate semantics defining the structure of the information in patient records and clinical trials we need also a common understanding when using clinical terms, medications etc. in the instance level. So the input of this step is the two ontologies with the corresponding instances and the output is the two ontologies with the instances annotated using medical terminologies.

So, the first step towards the matching of patient records and clinical trials is to have the instances annotated using relevant terminologies. We chose to use the following terminologies for annotation:

- **MedDra**[10] (**Med**ical **D**ictionary for **R**egulatory **A**ctivities, developed by the IFPMA -International Federation of Pharmaceutical Manufacturers and Associations): It is a medical vocabulary used to classify adverse events associated to the use of biopharma and other medical products on human.
- **SNOMED-CT**[11] (**S**ystematized **No**menclature of **Med**icine **- Cl**inical **T**erms): It is one of the most commonly used medical terminologies with over a million medical concepts. It provides medical terms covering codes, terms, synonyms and definitions covering diseases, findings, procedures, microorganisms, substances, etc
- **LOINC**[12] (**L**ogical **O**bservation **I**dentifiers **N**ames and **C**odes): It provides a standard to identify clinical information in electronic reports. The purpose of LOINC is to facilitate the exchange, pooling and management of results for clinical care and research. The context of LOINC includes laboratory test and other clinical observations but has expanded to include nursing diagnosis, nursing interventions, outcomes classification and patient care data. LOINC actually have about 50.000 tests and observations with about 45.000 concepts.

Besides these terminologies, others could be used as well such as MeSH[13] , ICD-10[14] etc. but for the time being we used those in our experiments. There are several

[10] www.meddramsso.com
[11] *http://www.ihtsdo.org/*
[12] http://loinc.org/
[13] *http://www.nlm.nih.gov/mesh/*
[14] http://www.who.int/classifications/icd/en/

technologies that could be used for annotation such as NLP approaches or existing annotation tools such as the BioPortal Annotator[15] which we used in our examples.

4.3 Match Annotated Databases

In order to match annotated patient and clinical trial instances an ontology matcher is used. The input of this step is the two ontologies with annotated instances and the output is a set of schema and instance matchings.

The task of finding relationships between entities belonging to two different ontologies is called ontology matching (Flouris, 2008). In ontology matching the main research efforts are focused on the ontology schema level. In contrast to ontology schema matching, instance matching deals with the actual individuals, not with the alignment of class structures or entity types. Instance matching has to decide whether two entity descriptions refer to the same individual. Thus, it crucially depends on measuring the similarity between sets of instances. The output identifies related entities so it is ideal for the identification of relations between patients and clinical trials.

However, yet, only a few systems consider the ontology instance level. At the same time, the demand for high-quality ontology instance matching is crucial in the context of identity recognition, ontology population and semantic integration. This has been widely recognized recently and ontology matching initiatives (such as Ontology Alignment Evaluation Initiative[16]) already produce benchmarks for evaluating instance matching.

In our approach we used a multi-strategy matching system called OtO [2] that focuses on the ontology instance matching, but also implements schema alignment. The OtO matching system is domain independent and fully customizable by a domain expert at any level. The aforementioned system implements several different ontology matching processes, which can be roughly categorized into the processes that include schema matching algorithms and those that include instance matching algorithms. The schema matching process leverages the lexical, semantic and syntactic similarities of the entities of the schema. As far as the instance matching process is concerned, it does not only rely on lexical similarity measures and pair-wise instance comparison of the instance properties. The system optimizes the instance matching process by leveraging (a) the rich semantic knowledge we gain from the output mappings of the schema matching process, (b) the implicit knowledge of the domain expert by capturing the identification power of the properties and (c) the probability calculation of the result’s truth, in order to accurately and efficiently detect the ontology instances that represent the same real-world entity.

Advanced techniques for ontology instance matching are required to correctly combine data describing individuals in different sources and to improve the accuracy of the schema ontology alignment process. The annotation process imposed on our prototype improves this accuracy since common elements are easier to identify. The notion behind this is that the more significant the overlap of common instances of two ontology concepts is the more related these concepts are.

[15] http://bioportal.bioontology.org/annotator
[16] http://oaei.ontologymatching.org/

4.4 Review identified matchings (IT Expert)

Since we are using a generic purpose ontology matcher, a domain expert should be included in the process in order to enhance the quality of the produced matchings.

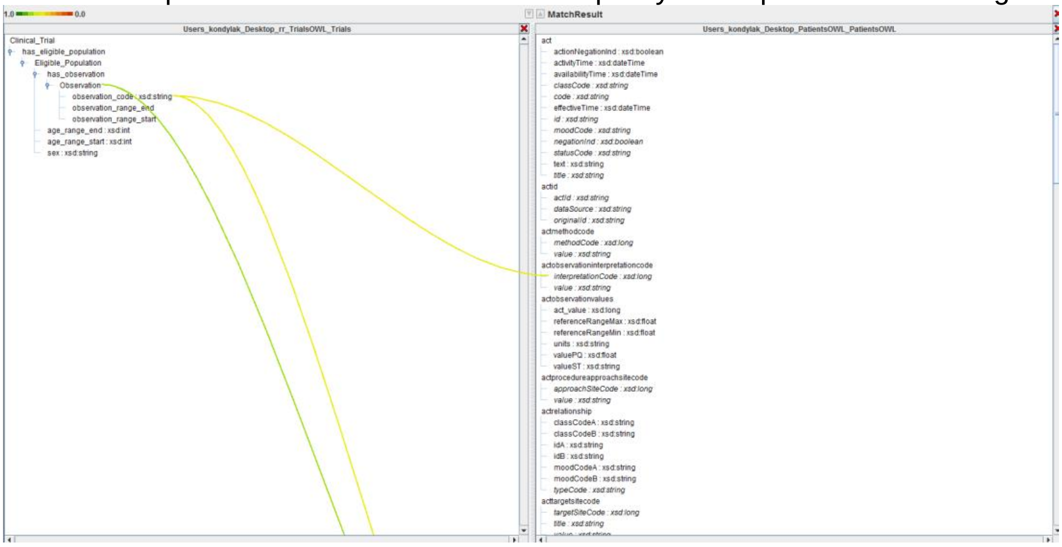
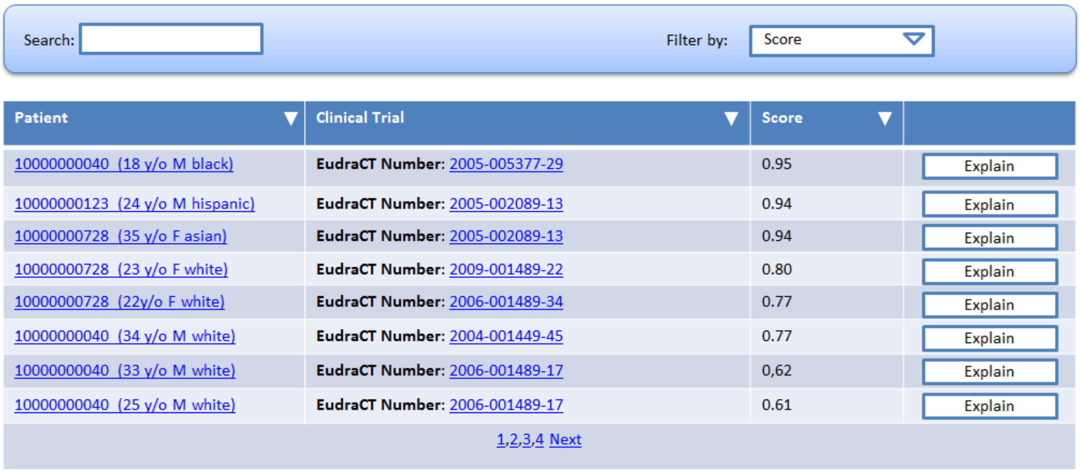


Figure 4. Reviewing Identified Schema Matchings

In our approach after executing for the first time the ontology matching, the results are presented graphically to him, similarly to the Figure 4. Then, he is able to correct and refine them. Although he has to make the necessary corrections only once and he is no longer required, he can later make additional corrections if required. Those corrections are saved in our database and used from then on to continue with instance matching.

4.5 Verify identified matchings (Domain Expert)

In this step the identified matchings between patients and clinical trials are presented to the domain expert along with a score between 0 and 1 describing the quality of the identified matching.



Search:

Filter by: Score

Patient ▼	Clinical Trial ▼	Score ▼	
10000000040 (18 y/o M black)	**EudraCT Number:** 2005-005377-29	0.95	Explain
10000000123 (24 y/o M hispanic)	**EudraCT Number:** 2005-002089-13	0.94	Explain
10000000728 (35 y/o F asian)	**EudraCT Number:** 2005-002089-13	0.94	Explain
10000000728 (23 y/o F white)	**EudraCT Number:** 2009-001489-22	0.80	Explain
10000000728 (22y/o F white)	**EudraCT Number:** 2006-001489-34	0.77	Explain
10000000040 (34 y/o M white)	**EudraCT Number:** 2004-001449-45	0.77	Explain
10000000040 (33 y/o M white)	**EudraCT Number:** 2006-001489-17	0,62	Explain
10000000040 (25 y/o M white)	**EudraCT Number:** 2006-001489-17	0.61	Explain

1,2,3,4 Next

Figure 5. Verifying identifying matchings

The clinician can then filter the results per patient, in order to find potential trials to enrol a specific patient, or per trial, in order to identify eligible patients for a specific trial. Moreover, he is able to set the baseline of the score describing the quality of the identified matchings. Of course, the system does not enrol automatically a patient to a clinical trial but only proposes possible matchings. It is up to the clinician to identify whether the proposed matching is actually valid. That's why the necessary links are provided to the corresponding clinical trial and patient.

4.6 Discussion & Conclusions

Although the first results of our approach seem to be promising, there are several issues that need further elaboration.

Lack of structured databases: In our approach a fundamental hypothesis is that we have structured databases for describing patients and trials. Although in most of the cases indeed we have structured databases for storing patient records this is not always true for clinical trial databases. Examples are EUdraCT[17] and ClinicalTrials.org[18] where although there is some structure on the text, the information there requires pre-processing before being able to use it.

Customizing generic ontology matchers for matching Clinical Trials with patient records: Although ontology matchers are ideal for retrieving instance similarities when trying to link Clinical Trials with patient records two problems occur that require matching customization: a) ranges and b) exclusion criteria.

In the first case customization is needed because the information in inclusion/exclusion criteria is usually described using ranges (e.g. age between 40 and 60), whereas in patient records ranges are not used. So customization is required in the matcher to correctly identify those cases.

Considering the exclusion criteria we chose not to include to the output presented to the clinician, patients which match the exclusion criteria. However, since ontology matchers, at the same time consider both inclusion and exclusion criteria and try to match patient records, special customization is required as well here. A solution when using generic matchers could be to match patient records to exclusion criteria, then remove from the matching list the patients that are identifies to match to the exclusion criteria and then proceed in matching the remaining patients using the inclusion criteria.

Extensive Evaluation: The implemented system is only a research prototype and cannot be used as is in a production environment. So, more work is needed in order to be usable from the clinicians. Moreover, although the initial experiments seem to be promising our future plans are to acquire a larger list of clinical trials and patients in order to identify the potential benefits of our system.

[17] https://eudract.ema.europa.eu/
[18] http://clinicaltrials.gov

5 Patient recruitment method

5.1 Introduction

One application where the formalization of eligibility criteria is useful, is in the domain of patient screening. In this screening the eligibility of a patient of interest is checked against one or more running clinical trials on a particular site. In current implemented trial workflows this checking is very cumbersome and time consuming for the investigator of the trial (especially when there are a large proportion of running trials available on the site) because they require a lot of manual intervention. A consequence of this is that a patient can be enrolled in a trial that does not provide the best treatment solution for the disease of this patient. An (semi-)automatically solution, using formalized criteria, can speed up the identification of eligible patients for enrolment in a clinical trial and thus can result in a more efficient recruitment. It is clear that both patient and trial investigator can profit from such a solution.

In the FP7 project INTEGRATE[19] a first approach was taken towards building an (semi-) automated patient screening method. The findings of the first iteration of the screening tool (focussing on the link with eligibility criteria) will be used as guidance in the next sections.

5.2 Approach

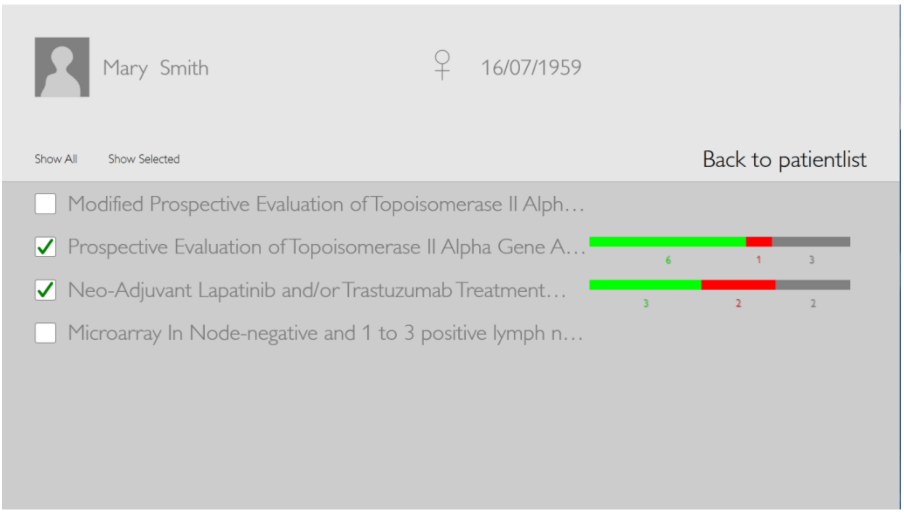
5.2.1 Semantic Interoperability

Before the method behind the matching of patient information with formalized eligibility criteria can be explained, it is important that the terminology between both is aligned. As seen in the previous chapters this task will rely on the semantic interoperability layer that enables the linkage among the concepts in the clinical trial system to concepts in the EHR system through the relevant core data set.

5.2.2 Screening Method

In an (semi-)automated patient screening tool, the trial investigator first selects the patient of interest. This way the system will know which records need to be queried in the EHR datawarehouse in order to retreive patient data. Next, an overview of available running trials (coming from a meta-data trial repository) on the site is presented to the investigator (see Figure 6). The trial investigator can select each of the trials to match the formalized trial criteria with the available patient data. This matching is the responsibility of a criteria matcher solution (see next section). The criteria matcher will return for each formalized criteria a match result, which is presented to the investigator of the trial (Figure 7). The investigator can approve and/or overrule a match result in a manual step. Further he can consult more detailed information about how the matcher came to the generated result.

[19] INTEGRATE Driving Excellence in Integrative Cancer Research. Available at: http://www.fp7-integrate.eu/ [20 September 2012]



☐ Modified Prospective Evaluation of Topoisomerase II Alph…
☑ Prospective Evaluation of Topoisomerase II Alpha Gene A…
☑ Neo-Adjuvant Lapatinib and/or Trastuzumab Treatment…
☐ Microarray In Node-negative and 1 to 3 positive lymph n…

Figure 6: Trial selection

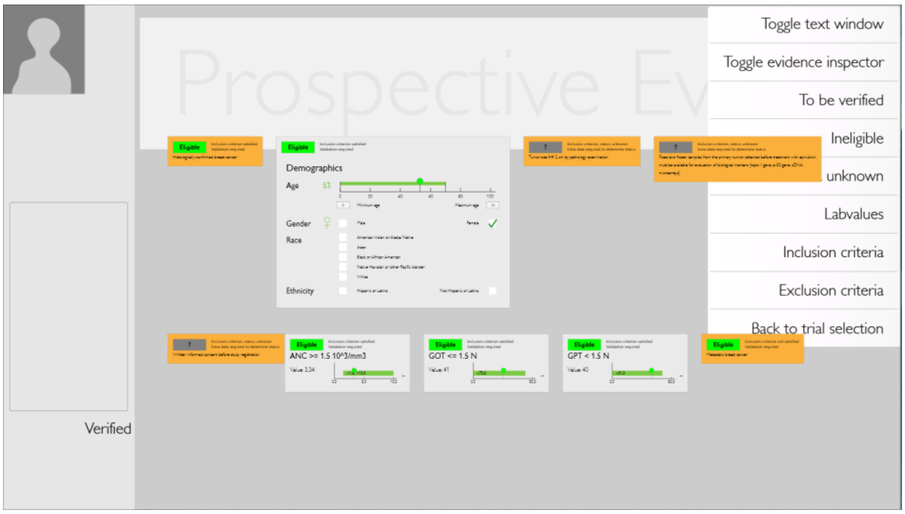


Figure 7: Trial criteria results

5.2.3 Criteria Matcher

The criteria matcher is the core component of the screening component, responsible for automatically verifying that a patient satisfies a given eligibility criteria of a clinical trial. When a request enters the matcher, it will first request the list of formalised eligibility criteria coming with the selected trial. Each of these criteria is connected to a matching script in the meta-data trial repository. Each script describes how the criteria should be matched with the patient data coming from the EHR DW, in such a way that it is understandable by the matcher executing engine. The matcher executing engine will execute these scripts and generates a matching result together with evidence for the matching. There are three possible outcomes for a matching result:

- MATCH: The patient matches the criteria
- NON-MATCH: The patient does not match the criteria
- UNDETERMINED: The matcher cannot make a match

5.3 Conclusions

A (semi-)automatically solution for patient screening making use of formalized eligibility criteria can be a powerful tool for improving the cumbersome and time-consuming flow that is currently implemented on most sites. In the INTEGRATE project this was this solution was already tested for a basic set of formalized eligibility criteria, the initial results showed a substantial time saving.

6 REFERENCES

A. R. Aronson, F. L. (2010). An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association (JAMIA), 17*, 229-236.

K. Milian, A. t. (2011). Patterns of Clinical Trial Eligibility Criteria. *Proceedings of the AIME'11 workshop on Knowledge Representation for Healthcare (KR4HC11).*

[SWIProlog2012] Jan Wielemaker, Tom Schrijvers, Markus Triska, Markus, and Torbjorn Lager, SWI-Prolog, Journal of Theory and Practice of Logic Programming, 12:1-2, 67-96, 2012.

[SWIProlog2008] Jan Wielemaker, Zhisheng Huang, Lourens van der Meij, SWI-Prolog and the Web, Journal of Theory and Practice of Logic Programming, 8:3, 363-392, 2008.

[de Clercq et al. 2004] de Clercq PA, Blom JA, Korsten HH, Hasman A., Approaches for creating computer-interpretable guidelines that facilitate decision support, Artif Intell Med. 2004 May;31(1):1-27.

[1] Brochhausen M, Spear AD, Cocos C, Weiler G, Martìn L, Anguita A, Stenzhorn H, Daskalaki E, Schera F, Schwarz U, Sfakianakis S; Kiefer S, Dörr M, Graf N, Tsiknakis M (2010) The ACGT Master Ontology and Its Applications - Towards an Ontology-Driven Cancer Research and Management System. Journal of Biomedical Informatics. E-published ahead of print. DOI 10.1016/j.jbi.2010.04.008

[2] Daskalaki, E., Plexousakis, D.: OtO Matching System: A Multi-strategy Approach to Instance Matching. CAiSE 2012: 286-300

[3] Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: Ontology change: classification and survey. Knowledge Eng. Review (KER) 23(2):117-152 (2008)