Supporting patient screening to identify suitable clinical trials

Anca Bucur, Jasper van Leeuwen, Njin-Zu Chen, Brecht Claerhout, Kristof de Schepper, David Perez-Rey, Raul Alonso-Calvo, Lina Pugliano, Kamal Saini
September 2nd, 2014
The INTEGRATE objectives

- Close the loop between clinical research and care and help improve outcomes through innovative applications and by supporting the data and knowledge flow.
**Context**

- Based on a clinical scenario defined by the Breast International Group
A breast cancer patient is treated at a particular hospital
A BIG centre connected to the environment
The clinician would like to evaluate whether the patient is eligible for enrolment in one of the clinical trials
The clinician has access to the environment to:

- **Collect and manage all necessary medical information** for completing the eligibility check
  - Automated import from EHRs, lab systems, etc.
- Performing the actual **eligibility check**
- **Keep track** of all clinical trials (and their eligibility criteria)

Multi-stage eligibility process:
- Patient **data collected within the standard of care** (in EHR) is used to **pre-screen** the patient for available clinical trials
- Only **patients matching the pre-screening criteria** are **considered for** (expensive) **genomic tests**
Increasing semantic interoperability

Enables exchange of structured data with meaning in a heterogeneous environment

Sharable context

Standards-based linkage between trial criteria and patient data

Implementation heterogeneity

Semantic agreement

Core Dataset

Mapping Formalism

EHR Canonical model

CTS Canonical model

EHR systems

Clinical Trial Systems

Vdovjak R et al., Bridging the gap between clinical research and care - Approaches to semantic interoperability, security & privacy. Proc. of HEALTHINF. 2012
Automating patient screening - steps

Formalize clinical trial descriptions
  Capturing structure/context of eligibility criteria
  Identifying content
Select relevant subsets of ontologies (core dataset)
Validate the subsets with medical experts
Implement the reasoning functionality with the selected ontologies
Define the semantic mappings:
  Between the core dataset and patient data (in EHR)
  Between the core dataset and the trial criteria
Build the criteria matching algorithm and the patient screening tool
End-to-end eligibility evaluation process

End User Applications

Matcher result interpretation

<table>
<thead>
<tr>
<th>Matcher Result</th>
<th>Inclusion</th>
<th>Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATCH</td>
<td>Eligible</td>
<td>Ineligible</td>
</tr>
<tr>
<td>NON-MATCH</td>
<td>Ineligible</td>
<td>Eligible</td>
</tr>
<tr>
<td>UNDETERMINATE</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

Each criteria is 1 matching script

(outcome: MATCH, NONMATCH, UNDETERMINED)

SPARQL Queries based on CIM

E.g. Expansion of concepts

SPARQL Queries to CDM

‘raw’ data structure query
(1) Select relevant BRIDG concepts

(2) Extend with application specific concepts

(3) Generate
From criteria to templates

“No concurrent warfarin”
“No concurrent radiotherapy”

Context
Semantic content

No concurrent

“No prior cancer except for non-melanoma skin cancer”
“No prior hormonal treatment except for tamoxifen”
“No history of cancer except for carcinoma in situ”

No prior ()
except for ()

No concurrent ()

Criteria templates: provide the execution logic of the criteria

Core dataset: subsets of selected ontologies provide the semantic link to the data
Templates and associated formalisms

Templates and associated formalisms

// CRITERION
// Exclusion: Previous treatment with anthracycline
// Query:
SELECT DISTINCT ?act_id
WHERE {
    ?op a htrm:participation;
    htrm:participation_entityId ?id;
    htrm:participation_act ?actAct;
    htrm:act_classCode "VACDAV";
    htrm:.act_code ?code;
    htrm:act_id ?act_id.
    FILTER (?id = "Patient:1000000001").
}

// MAP
String values = semanticLayer.WSDL2Templ.ExecuteQuery(target, query);
QueryResult X = new QueryResult();
result[0] = new QueryResult();
if (values == null){
    result[0].setResult(MatchResult.MATCH);
    String act_id = values.get("act_id");
    result[0].setResult(MatchResult.MATCH);
    Evidence evidence = new Evidence();
    evidence.setEvidenceId(act_id);
    result[0].setEvidence(evidence);
}
return result;

Each template can be associated with several formalisms.
In the current implementation:
Groovy scripts with Sparql queries instantiated with core dataset concepts.
Patient screening application
Conclusions

• The application aims to automate certain tasks of the patient screening procedure to enhance modern clinical trial recruitment
• Identifies the clinical trials for which a patient is potentially eligible by matching the trial criteria to the available patient information
• Combines formal representations of criteria with a pragmatic implementation in which templates are linked to execution logic and extensively reused
• Solution leverages widely-adopted standards (data models, terminologies/ontologies)
• The tool is being evaluated & validated in concrete scenarios at clinical sites
Thank you!
Any Questions?